"Have you heard of Google?"

It was a blazing hot July day in 2007, in the rural Indian village of Ragihalli, located thirty miles outside Bangalore. Twenty-two people from a company based in Mountain View, California, had driven in SUVs and vans up an unpaved road to this enclave of seventy threadbare huts with cement floors, surrounded by fields occasionally trampled by unwelcome elephants. Though electricity had come to Ragihalli some years earlier, there was not a single personal computer in the community. The visit had begun awkwardly, as the outsiders piled out of the cars and faced the entire population of the village, about two hundred people, who had turned out to welcome them. It was as if these well-dressed Westerners had dropped in from another planet, which in a sense they had. Young schoolchildren were pushed forward, and they performed a song. The visitors, in turn, gave the children notebooks and candy. There was an uncomfortable silence, broken when Marissa Mayer, the delegation's leader, a woman of thirty-two, said, "Let's interact with them." The group fanned out and began to engage the villagers in awkward conversation.

That is how Alex Vogenthaler came to ask a spindly young man with a wide smile whether he had heard of Google, Vogenthaler's employer. It was a question that he would never have had to ask in his home country: virtually everyone in the United States and everywhere in the wired-up world
knew Google. Its uncannily effective Internet search product had changed the way people accessed information, changed the way they thought about information. Its 2004 IPO had established it as an economic giant. And its founders themselves were the perfect examples of the superbrained engineering mentality that represented the future of business in the Internet age.

The villager admitted that, no, he had never heard of this Google. "What is it?" he asked. Vogenthaler tried to explain in the simplest terms that Google was a company that operated on the Internet. People used it to search for information. You would ask it a question, and it would immediately give you the answer from huge repositories of information it had gathered on the World Wide Web.

The man listened patiently but clearly was more familiar with rice fields than search fields.

Then the villager held up a cell phone. "Is this you what mean?" he seemed to ask.

The little connectivity meter on the phone display had four bars. There are significant swaths of the United States of America where one can barely pull in a signal—or gets no bars at all. But here in rural India, the signal was strong.

Google, it turns out, was on the verge of a multimillion-dollar mobile effort to make smart phones into information prostheses, adjuncts to the human brain that would allow people to get information to a vast swath of the world's knowledge instantly. This man might not know Google yet, but the company would soon be in Ragihalli. And then he would know Google.

I witnessed this exchange in 2007 as an observer on the annual trip of Google associate product managers, a select group pegged as the company's future leaders. We began our journey in San Francisco and touched down in Tokyo, Beijing, Bangalore, and Tel Aviv before returning home sixteen days later.

My participation on the trip had been a consequence of a long relationship with Google. In late 1998, I'd heard buzz about a smarter search engine and tried it out. Google was miles better than anything I'd used before. When I heard a bit about the site's method of extracting such good results—it relied on sort of a web-based democracy—I became even more intrigued. This is how I put it in the February 22, 1999, issue of Newsweek: "Google, the Net's hottest search engine, draws on feedback from the web itself to deliver more relevant results to customer queries."

Later that year, I arranged with Google's newly hired director of corporate communications, Cindy McCaffrey, to visit its Mountain View headquarters. One day in October I drove to 2400 Bayshore Parkway, where Google had just moved from its previous location above a Palo Alto bicycle shop. I'd visited a lot of start-ups and wasn't really surprised by the genial chaos—a vast room, with cubicles yet unfilled and a cluster of exercise balls. However, I hadn't expected that instead of being attired in traditional T-shirts and jeans, the employees were decked out in costumes. I had come on Halloween.

"Steven, meet Larry Page and Sergey Brin," said Cindy, introducing me to the two young men who had founded the company as Stanford graduate students. Larry was dressed as a Viking, with a long-haired fur vest and a hat with long antlers protruding. Sergey was in a cow suit. On his chest was a rubber slab from which protruded huge, wart-specked teats. They greeted me cheerfully and we all retreated to a conference room where the Viking and the cow explained the miraculous powers of Google's PageRank technology.

That was the first of many interviews I would conduct at Google. Over the next few years, the company became a focus of my technology reporting at Newsweek. Google grew from the small start-up I had visited to a behemoth of more than 20,000 employees. Every day, billions of people used its search engine, and Google's remarkable ability to deliver relevant results in milliseconds changed the way the world got its information. The people who clicked on its ads made Google wildly profitable and turned its founders into billionaires—and triggered an outcry among traditional beneficiaries of ad dollars.

Google also became known for its irreverent culture and its data-driven approach to business decision-making; management experts rhhapsodized about its unconventional methods. As the years went by, Google began to interpret its mission—to gather and make accessible and useful the world's information—in the broadest possible sense. The company created a series of web-based applications. It announced its intention to scan all the world's books. It became involved in satellite imagery, mobile phones, energy generation, photo storage. Clearly, Google was one of the most important contributors to the revolution of computers and technology that marked a turning point in civilization. I knew I wanted to write a book about the company but wasn't sure how.

Then in early July 2007, I was asked to join the associate product managers on their trip. It was an unprecedented invitation from a com-
pany that usually limits contact between journalists and its employees. The APM program, I learned, was a highly valued initiative. To quote the pitch one of the participants made in 2006 to recent and upcoming college graduates: “We invest more into our APMs than any other company has ever invested into young employees... We envision a world where everyone is swayed by the fact that Google’s executives, the best CEOs in the Silicon Valley, and the most respected leaders of global non-profits all came through the Google APM program.” Eric Schmidt, Google’s CEO, told me, “One of these people will probably be our CEO one day—we just don’t know which one.”

The eighteen APMs on the trip worked all over Google: in search, advertising, applications, and even stealth projects such as Google’s attempt to capture the rights to include magazines in its index. Mayer’s team, along with the APMs themselves, had designed the agenda of the trip. Every activity had an underlying purpose to increase the participants’ understanding of a technology or business issue, or make them more (in the parlance of the company) “Googley.” In Tokyo, for instance, they engaged in a scavenger hunt in the city’s legendary Akihabara electronics district. Teams of APMs were each given $50 to buy the weirdest gadgets they could find. Ducking into backstreets with stalls full of electronic parts and gizmos, they wound up with a cornucopia: USB-powered ashtrays shaped like football helmets that suck up smoke; a plate-sized disk that simulated the phases of the moon; a breathalyzer you could install in your car; and a stubby wand that, when waved back and forth, spelled out words in LED lights. In Bangalore, there was a different shopping hunt—an excursion to the market area where the winner of the competition would be the one who haggled best. (Good training for making bulk purchases of computers or even buying an Internet start-up.) Another Tokyo high point was the 5 a.m. trip to the Tsukiji fish market. It wasn’t the fresh sushi that fascinated the APMs but the mechanics of the fish auction, in some ways similar to the way Google works its AdWords program.

In China, Google’s top executive there, Kai-Fu Lee, talked of balancing Google’s freewheeling style with government rules—and censorship. But during interviews with Chinese consumers, the APMs were discouraged to hear the perception of the company among locals: “Baidu [Google’s local competitor] knows more [about China] than Google,” said one young man to his APM interlocutors.

At every office the APMs visited, they attended meetings with local Googlers, first learning about projects under way and then explaining to the residents what was going on at Mountain View headquarters. I began to get an insider’s sense of Google’s product processes—and how serving its users was akin to a crusade. An interesting moment occurred in Bangalore when Mayer was taking questions from local engineers after presenting an overview of upcoming products. One of them asked, “We’ve heard the roadmap for products, what’s the roadmap for revenue?” She almost bit his head off. “That’s not the way to think,” she said. “We’re focused on our users. If we make them happy, we will have revenues.”

The most fascinating part of the trip was the time spent with the young Googlers. They were generally from elite colleges, with SAT scores approaching or achieving perfection. Carefully culled from thousands of people who would have killed for the job, their personalities and abilities were a reflection of Google’s own character. During a bus ride to the Great Wall of China, one of the APMs charted the group demographics and found that almost all had parents who were professionals and more than half had parents who taught at a university—which put them in the company of Google’s founders. They all grew up with the Internet and considered its principles to be as natural as the laws of gravity. They were among the brightest and most ambitious of a generation that was better equipped to handle the disruptive technology wave than their elders were. Their minds hummed like tuning forks in resonance with the company’s values of speed, flexibility, and a deep respect for data.

Yet even while immersed in an optimism bubble with these young people, I could see the strains that came with Google’s abrupt growth from a feisty start-up to a market-dominating giant with more than 20,000 employees. The APMs had spent a year navigating the folkways of a complicated corporation, albeit a determinedly different one—and now they were almost senior employees. What’s more, I was stunned when a poll of my fellow travelers revealed that not a single one of them saw him- or herself working for Google in five years. Marissa Mayer took this news calmly, claiming that such ambition was why they had been hired in the first place. “This is the gene that Larry and Sergey look for,” she told me. “Even if they leave, it’s still good for us. They’re going to take the Google DNA with them.”

After covering the company for almost a decade, I thought I knew it pretty well, but the rare view of the company I got in those two weeks made me see it in a different, wider light. Still, there were considerable mysteries. Google was a company built on the values of its founders, who harbored ambitions to build a powerful corporation that would impact the
entire world, at the same time loathing the bureaucracy and commitments that running such a company would entail. Google professed a sense of moral purity—as exemplified by its informal motto, “Don’t be evil”—but it seemed to have a blind spot regarding the consequences of its own technology on privacy and property rights. A bedrock principle of Google was serving its users—but a goal was building a giant artificial intelligence learning machine that would bring uncertain consequences to the way all of us live. From the very beginning, its founders said that they wanted to change the world. But who were they, and what did they envision this new world order to be?

After the trip I realized that the best way to answer these questions was to report as much as possible from inside Google. Just as I’d had a rare glimpse into its inner workings during that summer of 2007, I would try to immerse myself more deeply into its engineering, its corporate life, and its culture, to report how it really operated, how it developed its products, and how it was managing its growth and public exposure. I would be an outsider with an insider’s view.

To do this, of course, I’d need cooperation. Fortunately, based on our long relationship, Google’s executives, including “LSE”—Larry Page, Sergey Brin, and Eric Schmidt—agreed to let me in. During the next two years—a critical time when Google’s halo lost some of its glow even as the company grew more powerful—I interviewed hundreds of current and former Googlers and attended a variety of meetings in the company. These included product development meetings, “interface reviews,” search launch meetings, privacy council sessions, weekly TGIF all-hands gatherings, and the gatherings of the high command known as Google Product Strategy (GPS) meetings, where projects and initiatives are approved or rejected. I also ate a lot of meals at Andale, the burrito joint in Google’s Building 43.

What I discovered was a company exulting in creative disorganization, even if the creativity was not always as substantial as hoped for. Google had massive goals, and the entire company channeled its values from the founders. Its mission was collecting and organizing all the world’s information—and that’s only the beginning. From the very start, its founders saw Google as a vehicle to realize the dream of artificial intelligence in augmenting humanity. To realize their dreams, Page and Brin had to build a huge company. At the same time, they attempted to maintain as much as possible the nimble, irreverent, answer-to-no-one freedom of a small start-up. In the two years I researched this book, the clash between those goals reached a peak, as David had become a Goliath.

My inside perspective also provided me the keys to unlock more of the secrets of Google’s two “black boxes”—its search engine and its advertising model—than had previously been disclosed. Google search is part of our lives, and its ad system is the most important commercial product of the Internet age. In this book, for the first time, readers can learn the full story of their development, evolution, and inner workings. Understanding those groundbreaking products helps us understand Google and its employees because their operation embodies both the company’s values and its technological philosophy. More important, understanding them helps us understand our own world—and tomorrow’s.

The science fiction writer William Gibson once said that the future is already here—just not evenly distributed. At Google, the future is already under way. To understand this pioneering company and its people is to grasp our technological destiny. And so here is Google: how it works, what it thinks, why it’s changing, how it will continue to change us. And how it hopes to maintain its soul.
"It was science fiction more than computer science."

On February 18, 2010, Judge Denny Chin of the New York Southern District federal court took stock of the packed gallery in Courtroom 23B. It was going to be a long day. He was presiding over a hearing that would provide only a gloss to hundreds of submissions he had already received on this case. "There is just too much to digest," he said. He shook his head, preparing himself to hear the arguments of twenty-seven representatives of various interest groups or corporations, as well as presentations by some of the lawyers for various parties, lawyers who filled every place in two long tables before him.

The case was *The Authors Guild, Inc., Association of American Publishers, et al. v. Google Inc.* It was a lawsuit tentatively resolved by a class settlement agreement in which an authors' group and a publishers' association set conditions for a technology company to scan and sell books. Judge Chin's decision would involve important issues affecting the future of digital works, and some of the speakers before the court engaged on those issues. But many of the objectors—and most who addressed the court were
objectors to the settlement—focused on a young company headquartered on a sprawling campus in Mountain View, California. That company was Google. The speakers seemed to distrust it, fear it, even despise it.

“A major threat to ... freedom of expression and participation in cultural diversity”
“An unjustified monopoly”
“Eviscerates privacy protections”
“Concealment and misdirection”
“Price fixing ... a massive market distortion ... preying on the desperate”
“May well be a per se violation of the antitrust laws”

(That last statement held special weight, as it came from the U.S. deputy assistant attorney general.)

But the federal government was only one of Google’s surprising opponents. Some of the others were supporters of the public interest, monitoring the privacy rights and pocketbooks of citizens. Others were advocates of free speech. There was even an objector representing the folksinger Arlo Guthrie.

The irony was that Google itself explicitly embraced the lofty values and high moral standards that it was being attacked for flouting. Its founders had consistently stated that their goal was to make the world better, specifically by enabling humanity’s access to information. Google had created an astonishing tool that took advantage of the interconnected nature of the burgeoning World Wide Web, a tool that empowered people to locate even obscure information within seconds. This search engine transformed the way people worked, entertained themselves, and learned. Google made historic profits from that product by creating a new form of advertising—nonintrusive and even useful. It hired the sharpest minds in the world and encouraged them to take on challenges that pushed the boundaries of innovation. Its focus on engineering talent to accomplish difficult goals was a national inspiration. It even warned its shareholders that the company would sometimes pursue business practices that serve humanity even at the expense of lower profits. It accomplished all those achievements with a puckish irreverence that captivated the public and made heroes of its employees.

But that didn’t matter to the objectors in Judge Chin’s courtroom. Those people were Google’s natural allies, and they thought that Google was no longer ... good. The mistrust and fear in the courtroom were reflected globally by governments upset by Google’s privacy policies and businesses worried that Google’s disruptive practices would target them next. Everywhere Google’s executives turned, they were faced with protests and lawsuits.

The course of events was baffling to Google’s two founders, Larry Page and Sergey Brin. Of all Google’s projects, the one at issue in the hearing—Google’s Book Search project—was perhaps the most idealistic. It was an audacious attempt to digitize every book every printed, so that anyone in the world could locate the information within. Google would not give away the full contents of the books, so when users discovered them, they would have reason to buy them. Authors would have new markets; readers would have instant access to knowledge. After being sued by publishers and authors, Google made a deal with them that would make it even easier to access the books and to buy them on the spot. Every library would get a free terminal to connect to the entire corpus of the world’s books. To Google, it was a boon to civilization.

Did people understand?

By all metrics, the company was still thriving. Google still retained its hundreds of millions of users, hosted billions of searches every day, and had growing businesses in video and wireless devices. Its employees were still idealistic and ambitious in the best sense. But a shadow now darkened Google’s image. To many outsiders, the corporate motto that Google had taken seriously—“Don’t be evil”—had become a joke, a bludgeon to be used against it.

What had happened?

Doing good was Larry Page’s plan from the very beginning. Even as a child, he wanted to be an inventor, not simply because his mind aligned perfectly with the nexus of logic and technology (which it did) but because, he says, “I really wanted to change the world.”

Page grew up in Lansing, Michigan, where his father taught computer science at Michigan State. His parents divorced when he was eight, but he was close with both his father and mother—who had her-own computer science degree. Naturally, he spoke computers as a primary language. As he later told an interviewer, “I think I was the first kid in my elementary school to turn in a word-processed document.”

Page was not a social animal—people who talked to him often wondered if there were a jigger of Asperger’s in the mix—and could unnerv people by simply not talking. But when he did speak, more often than not
he would come out with ideas that bordered on the fantastic. Attending a summer program in leadership (motto: “A healthy disregard for the impossible”) helped move him to action. At the University of Michigan, he became obsessed with transportation and drew up plans for an elaborate monorail system in Ann Arbor, replacing the mundane bus system with a “futuristic” commute between the dorms and the classrooms. It seemed to come as a surprise to him that a fanciful multimillion-dollar transit fantasy from an undergraduate would not be quickly embraced and implemented. (Fifteen years after he graduated, Page would bring up the issue again in a meeting with the university’s president.)

His intelligence and imagination were clear. But when you got to know him, what stood out was his ambition. It expressed itself not as a personal drive (though there was that, too) but as a general principle that everyone should think big and then make big things happen. He believed that the only true failure was not attempting the audacious. “Even if you fail at your ambitious thing, it’s very hard to fail completely,” he says. “That’s the thing that people don’t get.” Page always thought about that. When people proposed a short-term solution, Page’s instinct was to think long term. There would eventually be a joke among Googlers that Page “went to the future and came back to tell us about it.”

Page earned a degree in computer science like his father did. But his destiny was in California, specifically in the Silicon Valley. In a way, Page’s arrival at Stanford was a homecoming. He’d lived there briefly in 1979 when his dad had spent a sabbatical at Stanford; some faculty members still remembered him as an insatiably curious seven-year-old. In 1995, Stanford was not only the best place to pursue cutting-edge computer science but, because of the Internet boom, was also the world capital of ambition. Fortunately, Page’s visions extended to the commercial: “Probably from when I was twelve, I knew I was going to start a company eventually,” he’d later say. Page’s brother, nine years older, was already in Silicon Valley, working for an Internet start-up.

Page chose to work in the department’s Human-Computer Interaction Group. The subject would stand Page in good stead in the future with respect to product development, even though it was not in the HCI domain to figure out a new model of information retrieval. On his desk and permeating his conversations was Apple interface guru Donald Norman’s classic tome The Psychology of Everyday Things, the bible of a religion whose first, and arguably only, commandment is “The user is always right.” (Other Norman disciples, such as Jeff Bezos at Amazon.com, were adopting this creed on the web.) Another influential book was a biography of Nikola Tesla, the brilliant Serb scientist; though Tesla’s contributions arguably matched Thomas Edison’s—and his ambitions were grand enough to impress even Page—he died in obscurity. “I felt like he was a great inventor and it was a sad story,” says Page. “I feel like he could’ve accomplished much more had he had more resources. And he had trouble commercializing the stuff he did. Probably more trouble than he should’ve had. I think that was a good lesson. I didn’t want to just invent things, I also wanted to make the world better; and in order to do that, you need to do more than just invent things.”

The summer before entering Stanford, Page attended a program for accepted candidates that included a tour of San Francisco. The guide was a grad student Page’s age who’d been at Stanford for two years. “I thought he was, pretty obnoxious,” Page later said of the guide, Sergey Brin. The content of the encounter is now relegated to legend, but their argumentative banter was almost certainly good-natured. Despite the contrast in personalities, in some ways they were twins. Both felt most comfortable in the meritocracy of academia, where brains trumped everything else. Both had an innate understanding of how the ultraconnected world that they enjoyed as computer science (CS) students was about to spread throughout society. Both shared a core belief in the primacy of data. And both were rock-stubborn when it came to pursuing their beliefs. When Page settled in that September, he became close friends with Brin, to the point where people thought of them as a set: Larry and Sergey.

Born in Russia, Brin was four when his family immigrated to the United States. His English still maintained a Cyrillic flavor, and his speech was dotted with anachronistic Old World touches such as the use of “whatnot” when peers would say “stuff like that.” He had arrived at Stanford at nineteen after whizzing through the University of Maryland, where his father taught, in three years; he was one of the youngest students ever to start the Stanford PhD program. “He skipped a million years,” says Craig Silverstein, who arrived at Stanford a year later, and would eventually become Google’s first employee. Sergey was a quirky kid who would zip through Stanford’s hallways on omnipresent Rollerblades. He also had an interest in trapeze. But the professors understood that behind the goofiness was a formidable mathematical mind. Soon after arriving at Stanford, he knocked off all the required tests for a doctorate and was free to sample the courses until he found a suitable entree for a thesis. He supplemented his academics with swimming, gymnastics, and sailing. (When his father asked
him in frustration whether he planned to take advanced courses, he said that he might take advanced swimming.) Donald Knuth, a Stanford professor whose magisterial series of books on the art of computer programming made him the Proust of computer code, recalls driving down the Pacific coast to a conference with Sergey one afternoon and being impressed at his grasp of complicated issues. His adviser, Hector Garcia-Molina, had seen a lot of bright kids go through Stanford, but Brin stood out. "He was brilliant," Garcia-Molina says.

One task that Brin took on was a numbering scheme for the new Gates Computer Science Building, which was to be the home of the department. (His system used mathematical flourishes.) The structure was named after William Henry Gates III, better known as Bill, the cofounder of Microsoft. Though Gates had spent a couple of years at Harvard and endowed a building named after his mother there, he went on a small splurge of funding palatial new homes for computer science departments at top technical institutions that he didn't attend, including MIT and Carnegie Mellon—along with Stanford, the trifecta of top CS programs. Even as they sneered at Windows, the next generation of wizards would study in buildings named after Bill Gates.

Did Gates ever imagine that one of those buildings would incubate a rival that might destroy Microsoft?

The graduate computer science program at Stanford was built around close relationships between students and faculty members. They would team up to work on big, real-world problems; the freshness perspective of the young people maintains the vitality of the professor's interests. "You always follow the students," says Terry Winograd, who was Page's adviser. (Page would often remind him that they had met during his dad's Stanford sabbatical.) Over the years Winograd had become an expert at figuring out where students stood on the spectrum of brainiacs who found their way into the department. Some were kids whose undergrad record was straight A phases, GRE scores scraping perfection, who would come in and say, "What thesis should I work on?" On the other end of the spectrum were kids like Larry Page, who would come in and say, "Here's what I think I can do." And his proposals were crazy. He'd come into the office and talk about doing something with space tethers or solar kites. "It was science fiction more than computer science," recalls Winograd. But an outlandish mind was a valuable asset, and there was definitely a place in the current science to channel wild creativity.

In 1995, that place was the World Wide Web. It had sprung from the restless brain of a (then)-obscure British engineer named Tim Berners-Lee, who was working as a technician at the CERN physics research lab in Switzerland. Berners-Lee could sum up his vision in a sentence: "Suppose all the information stored on computers everywhere were linked... there would be a single global information space."

The web's pedigree could be traced back to a 1945 paper by the American scientist Vannevar Bush. Entitled "As We May Think," it outlined a vast storage system called a "memex," where documents would be connected, and could be recalled, by information breadcrumbs called "trails of association." The timeline continued to the work of Douglas Engelbart, whose team at the Stanford Research Institute devised a linked document system that lived behind a dazzling interface that introduced the metaphors of windows and files to the digital desktop. Then came a detour to the brilliant but erratic work of an autodidact named Ted Nelson, whose ambitious Xanadu Project (though never completed) was a vision of disparate information linked by "hypertext" connections. Nelson's work inspired Bill Atkinson, a software engineer who had been part of the original Macintosh team; in 1987 he came up with a link-based system called HyperCard, which he sold to Apple for $100,000 on the condition that the company give it away to all its users. But to really fulfill Vannevar Bush's vision, you needed a huge system where people could freely post and link their documents.

By the time Berners-Lee had his epiphany, that system was in place: the Internet. While the earliest websites were just ways to distribute academic papers more efficiently, soon people began writing sites with information of all sorts, and others created sites just for fun. By the mid-1990s, people were starting to use the web for profit, and a new word, "e-commerce," found its way into the lexicon. Amazon.com and eBay became Internet giants. Other sites positioned themselves as gateways of portals, to the wonders of the Internet.

As the web grew, its linking structure accumulated a mind-boggling value. It treated the aggregate of all its contents as a huge compost of ideas, any one of which could be reached by the act of connecting one document to another. When you looked at a page you could see, usually highlighted in blue, the pointers to other sites that the webmaster had coded on the page—that was the hypertext idea that galvanized Bush, Nelson, and Atkinson. But for the first time, as Berners-Lee had intended, the web was coagulating a critical mass of these linked sites and documents into a single network. In effect, the web was an infinite database, a sort of crazily expanding universe of human knowledge that, in theory, could hold every insight, thought,
image, and product for sale. And all of it had an intricate lattice of crossconnections created by the independent linking activity of anyone who had built a page and coded a link to something elsewhere on the web.

In retrospect, the web was to the digital world what the Louisiana Purchase was to the young United States: the opportunity of a century.

Berners-Lee’s creation was so new that when Stanford got funding from the National Science Foundation in the early 1990s to start a program called the Digital Library Project, the web wasn’t mentioned in the proposal. “The theme of that project was interoperability—how can we make all these resources work together?” recalls Hector García-Molina, who cofounded the project. By 1995 though, García-Molina knew that the World Wide Web would inevitably be part of the projects concocted by the students who worked with the program, including Page and Brin.

Brin already had a National Science Foundation fellowship and didn’t need funding, but he was trying to figure out a dissertation topic. His loose focus was data mining, and with Rajeev Motwani, a young professor he became close with, he helped start a research group called MIDAS, which stood for Mining Data at Stanford. In a résumé he posted on the Stanford site in 1995, he talked about “a new project” to generate personalized movie ratings. “The way it works is as follows,” he wrote. “You rate the movies you have seen. Then the system finds other users with similar tastes to extrapolate how much you like other movies.” Another project he worked on with García-Molina and another student was a system that detected copyright violations by automating searches for duplicates of documents. “He came up with some good algorithms for detecting copies,” says García-Molina. “Now you use Google.”

Page was also seeking a dissertation topic. One idea he presented to Winograd, a collaboration with Brin, seemed more promising than the others: creating a system where people could make annotations and comments on websites. But the more Page thought about annotation, the messier it got. For big sites, there would probably be a lot of people who wanted to mark up a page. How would you figure out who gets to comment or whose comment would be the one you’d see first? For that, he says, “We needed a rating system.”

Having a human being determine the ratings was out of the question. First, it was inherently impractical. Further, humans were unreliable. Only algorithms—well drawn, efficiently executed, and based on sound data—could deliver unbiased results. So the problem became finding the right data to determine whose comments were more trustworthy, or interesting, than others. Page realized that such data already existed and no one else was really using it. He asked Brin, “Why don’t we use the links on the web to do that?”

Page, a child of academia, understood that web links were like citations in a scholarly article. It was widely recognized that you could identify which papers were really important without reading them—simply tally up how many other papers cited them in notes and bibliographies. Page believed that this principle could also work with web pages. But getting the right data would be difficult. Web pages made their outgoing links transparent: built into the code were easily identifiable markers for the destinations you could travel to with a mouse click from that page. But it wasn’t obvious at all what linked to a page. To find that out, you’d have to somehow collect a database of links that connected to some other page. Then you’d go backward.

That’s why Page called his system BackRub. “The early versions of hypertext had a tragic flaw: you couldn’t follow links in the other direction,” Page once told a reporter. “BackRub was about reversing that.”

Winograd thought this was a great idea for a project, but not an easy one. To do it right, he told Page, you’d really have to capture a significant chunk of the World Wide Web’s link structure. Page said, sure, he’d go and download the web and get the structure. He figured it would take a week or something. “And of course,” he later recalled, “it took, like, years.” But Page and Brin attacked it. Every other week Page would come to García-Molina’s office asking for disks and equipment. “That’s fine,” García-Molina would say. “This is a great project, but you need to give me a budget.” He asked Page to pick a number, to say how much of the web he needed to crawl, and to estimate how many disks that would take. “I want to crawl the whole web,” Page said.

Page indulged in a little vanity in naming the part of the system that rated websites by the incoming links: he called it PageRank. But it was a sly vanity; many people assumed the name referred to web pages, not a surname.

Since Page wasn’t a world-class programmer, he asked a friend to help out. Scott Hassan was a full-time research assistant at Stanford, working for the Digital Library Project while doing part-time grad work. Hassan was also good friends with Brin, whom he’d met at an Ultimate Frisbee game during his first week at Stanford. Page’s program “had so many bugs in it, it wasn’t funny,” says Hassan. Part of the problem was that Page was using the relatively new computer language Java for his ambitious project,
and Java kept crashing. "I went and tried to fix some of the bugs in Java itself, and after doing this ten times, I decided it was a waste of time," says Hassan. "I decided to take his stuff and just rewrite it into the language I knew much better that didn't have any bugs."

He wrote a program in Python—a more flexible language that was becoming popular for web-based programs—that would act as a "spider," so called because it would crawl the web for data. The program would visit a web page, find all the links, and put them into a queue. Then it would check to see if it had visited those link pages previously. If it hadn't, it would put the link on a queue of future destinations to visit and repeat the process. Since Page wasn't familiar with Python, Hassan became a member of the team. He and another student, Alan Stereberg, became paid assistants to the project.

Brin, the math prodigy, took on the huge task of crunching the mathematics that would make sense of the mess of links uncovered by their monster survey of the growing web.

Even though the small team was going somewhere, they weren't quite sure of their destination. "Larry didn't have a plan," says Hassan. "In research you explore something and see what sticks."

By March 1996, they began a test, starting at a single-page, the Stanford computer science department home page. The spider located the links on the page and fanned out to all the sites that linked to Stanford, then to the sites that linked to those websites. "That first one just used the titles of documents because collecting the documents themselves required a lot of data and work," says Page. After they snared about 15 million of those titles, they tested the program to see which websites it deemed more authoritative.

"Even the first set of results was very convincing," Hector Garcia-Molina says. "It was pretty clear to everyone who saw this demo that this was a very good, very powerful way to order things."

"We realized it worked really, really well," says Page. "And I said, "Wow, the big problem here is not annotation. We should now use it not just for ranking annotations, but for ranking searches."" It seemed the obvious application for an invention that gave a ranking to every page on the web. "It was pretty clear to me and the rest of the group," he says, "that if you have a way of ranking things based not just on the page itself but based on what the world thought of that page, that would be a really valuable thing for search."

---

10 » IN THE PLEX

THE WORLD ACCORDING TO GOOGLE » 19
AltaVista: its breadth. When DEC opened it to outsiders on December 15, 1995, nearly 300,000 people tried it out. They were dazzled.

AltaVista's actual search quality techniques—what determined the ranking of results—were based on traditional information retrieval (IR) algorithms. Many of those algorithms arose from the work of one man, a refugee from Nazi Germany named Gerard Salton, who had come to America, got a PhD at Harvard, and moved to Cornell University, where he cofounded its computer science department. Searching through databases using the same commands you'd use with a human—"natural language" became the term of art—was Salton's specialty.

During the 1960s, Salton developed a system that was to become a model for information retrieval. It was called SMART, supposedly an acronym for "Salton's Magical Retriever of Text." The system established many conventions that still persist in search, including indexing and relevance algorithms. When Salton died in 1995, his techniques still ruled the field.

"For thirty years," wrote one academic in tribute a year later, "Gerry Salton was information retrieval."

The World Wide Web was about to change that, but the academics didn't know it—and neither did AltaVista. While its creators had the insight to gather all of the web, they missed the opportunity to take advantage of the link structure. "The innovation was that I was not afraid to fetch as much of the web as I could, store it in one place, and have a really fast response time. That was the novelty," says Monier. Meanwhile, AltaVista analyzed what was on each individual page—using metrics like how many times each word appeared—to see if a page was a relevant match to a given keyword in a query.

Even though there was no clear way to make money from search, AltaVista had a number of competitors. By 1996, when I wrote about search for Newsweek, executives from several companies were all boasting the most useful service. When pressed, all of them would admit that in the race between the omnivorous web and their burgeoning technology, the web was winning. "Academic IR had thirty years to get to where it is—we're breaking new ground, but it's difficult," complained Graham Spencer, the engineer behind the search engine created by a start-up called Excite. AltaVista's director of engineering, Barry Rubinson, said that the best approach was to throw massive amounts of silicon toward the problem and then hope for the best. "The first problem is that relevance is in the eye of the beholder," he said. The second problem, he continued, is making sense of the infuriatingly brief and cryptic queries typed into the AltaVista search field. He implied that the task was akin to voodoo. "It's all wizardry and witchcraft," he told me. "Anyone who tells you it’s scientific is just pulling your leg."

No one at the web search companies mentioned using links.

The links were the reason that a research project running on a computer in a Stanford dorm room had become the top performer. Larry Page's PageRank was powerful because it cleverly analyzed those links and assigned a number to them, a metric on a scale of 1 to 10, that allowed you to see the page's prominence in comparison to every other page on the web. One of the early versions of BackRub had simply counted the incoming links, but Page and Brin quickly realized that it wasn't merely the number of links that made things relevant. Just as important was who was doing the linking. PageRank reflected that information. The more prominent the status of the page that made the link, the more valuable the link was and the higher it would rise when calculating the ultimate PageRank number of the web page itself. "The idea behind PageRank was that you can estimate the importance of a web page by the web pages that link to it," Brin would say. "We actually developed a lot of math to solve that problem. Important pages tended to link to important pages. We convert the entire web into a big equation with several hundred million variables, which are the PageRanks of all the web pages, and billions of terms, which are all the links." It was Brin's mathematical calculations on those possible 500 million variables that identified the important pages. It was like looking at a map of airline routes: the hub cities would stand out because of all the lines representing flights that originated and terminated there. Cities that got the most traffic from other important hubs were clearly the major centers of population. The same applied to websites. "It's all recursive," Page later said. "In a way, how good you are is determined by who links to you and who you link to determines how good you are. It's all a big circle. But mathematics is great. You can solve this."

The PageRank score would be combined with a number of more traditional information retrieval techniques, such as comparing the keyword to text on the page and determining relevance by examining factors such as frequency, font size, capitalization, and position of the keyword. (Those factors help determine the importance of a keyword on a given page—if a term is prominently featured, the page is more likely to satisfy a query.) Such factors are known as signals, and they are critical to search quality. There are a few crucial milliseconds in the process of a web search during which the engine interprets the keyword and then accesses the vast index,
where all the text on billions of pages is stored and ordered just like an index of a book. At that point the engine needs some help to figure out how to rank those pages. So it looks for signals—traits that can help the engine figure out which pages will satisfy the query. A signal says to the search engine, “Hey, consider me for your results!” PageRank itself is a signal. A web page with a high PageRank number sends a message to the search engine that it’s a more reputable source than those with lower numbers.

Though PageRank was BackRub’s magic wand, it was the combination of that algorithm with other signals that created the mind-blowing results. If the keyword matched the title of the web page or the domain name, that page would go higher in the rankings. For queries consisting of multiple words, documents containing all of the search query terms in close proximity would typically get the nod over those in which the phrase matched was “not even close.” Another powerful signal was the “anchor text” of links that led to the page. For instance, if a web page used the words “Bill Clinton” to link to the White House, “Bill Clinton” would be the anchor text. Because of the high values assigned to anchor text, a BackRub query for “Bill Clinton” would lead to www.whitehouse.gov as the top result because numerous web pages with high PageRanks used the president’s name to link the White House site. “When you did a search, the right page would come up, even if the page didn’t include the actual words you were searching for,” says Scott Hassan, “That was pretty cool.” It was also something other search engines failed to do. Even though www.whitehouse.gov was the ideal response to the Clinton “navigation query,” other commercial engines didn’t include it in their results. (In April 1997, Page and Brin found that a competitor’s top hit was “Bill Clinton Joke of the Day.”)

PageRank had one other powerful advantage. To search engines that relied on the traditional IR approach of analyzing content, the web presented a terrible challenge. There were millions and millions of pages, and as more and more were added, the performance of those systems inevitably degraded. For those sites, the rapid expansion of the web was a problem, a drain on their resources. But because of PageRank, BackRub got better as the web grew. New sites meant more links. This additional information allowed BackRub to identify even more accurately the pages that might be relevant to a query. And the more recent links would improve the freshness of the site. “PageRank has the benefit of learning from the whole of the World Wide Web,” Brin would explain.

Of course, Brin and Page had the logistical problem of capturing the whole web. The Stanford team did not have the resources of DEC. For a while, BackRub could access only the bandwidth available to the Gates Building—10 megabits of traffic per second. But the entire university ran on a giant T3 line that could operate at 45 megabits per second. The BackRub team discovered that by retoggling an incorrectly set switch in the basement, it could get full access to the T3 line. “As soon as they toggled that, we were all the way up to the maximum of the entire Stanford network,” says Hassan. “We were using all the bandwidth of the network. And this was from a single machine doing this, on a desktop in my dorm room.”

In those days, people who ran websites—many of them with minimal technical savvy—were not used to their sites being crawled. Some of them would look at their logs, and see frequent visits from www.stanford.edu, and suspect that the university was somehow stealing their information. One woman from Wyoming contacted Page directly to demand that he stop, but Google’s “bot” kept visiting. She discovered that Hector Garcia-Molina was the project’s adviser and called him, charging that the Stanford computer was doing terrible things to her computer. He tried to explain to her that being crawled is a harmless, nondestructive procedure, but she’d have none of it. She called the department chair and the Stanford security office. In theory, complainants could block crawlers by putting a little piece of code on their sites called /robots.txt, but the angry webmasters weren’t receptive to the concept. “Larry and Sergey got annoyed that people couldn’t figure out /robots.txt,” says Winograd, “but in the end, they actually built an exclusion list, which they didn’t want to.” Even then, Page and Brin believed in a self-service system that worked in scale, serving vast populations. Handcrafting exclusions was anathema.

Brin and Page fell into a pattern of rapid iterating and launching. If the pages for a given query were not quite in the proper order, they’d go back to the algorithm and see what had gone wrong. It was a tricky balancing act to assign the proper weights to the various signals. “You do the ranking initially, and then you look at the list and say, ‘Are they in the right order?’ If they’re not, we adjust the ranking, and then you’re like, ‘Oh this looks really good,’” says Page. Page used the ranking for the keyword of “university” as a litmus test. He paid particular attention to the relative ranking of his alma mater, Michigan, and his current school, Stanford. Brin and Page assumed that Stanford would be ranked higher, but Michigan topped it. Was that a flaw in the algorithm? No. “We decided that Michigan had more stuff on the web, and that was reasonable,” says Page.

This listing showed the power of PageRank. It made BackRub much more useful than the results you’d get from the commercial search engines.
Their list of institutions for the "university" query seemed totally random. The number one result for that generic term in AltaVista would give you the Oregon Center for Optics. Page recalls a conversation back then with an AltaVista engineer who told him that with the way pages were scored, a query for "university" was likely to get a page where that word appeared twice in the headline. "That doesn't make any sense," Page said, noting that such a search was more likely to get a minor university with redundancy in its title.

"If you want major universities, you should type 'major universities,'" said the engineer. Page was appalled. "I'm like, well, they teach you in human computer interaction, which is my branch, that the user is never wrong. The person in the system is never wrong.'"

Until that moment, the task of compiling a list of universities and ranking them in significance had been complicated, intellectually challenging, and labor-intensive. Some magazines employed large teams working for months to do just that. If you were to try to teach a computer to do that, your instinct would be to feed it data about SAT scores, graduation rates, prizewinners among faculty, and a thousand other factors. Then you'd have to figure out how to weigh them. The odds were low that a machine would crank out a rating that squared with the gut feeling of a well-educated citizen. But BackRub knew nothing about those statistics. It just knew how to take advantage of the fact that links created by the web community had implicitly produced a ranking that was better than any group of magazine editors or knowledge curators could come up with. Larry Page and Sergey Brin had figured out how to mine that knowledge before the information retrieval establishment and commercial search engines even realized that it existed.

"The whole field had suffered blinders," says the computer scientist Amit Singhal, then a Bell Labs researcher who had been a protégé of Jerry Salton. "In some sense, search really did need two people who were never tainted by people like me to come up with that shake-up."

Larry Page was not the only person in 1996 who realized that exploiting the link structure of the web would lead to a dramatically more powerful way to find information. In the summer of that year, a young computer scientist named Jon Kleinberg arrived in California to spend a yearlong postdoctoral fellowship at IBM's research center in Almaden, on the southern edge of San Jose. With a new PhD from MIT, he had already accepted a tenure-track job in the CS department at Cornell University.

Kleinberg decided to look at web search. The commercial operations didn't seem effective enough and were further hobbled by spam. AltaVista's results in particular were becoming less useful because websites had gamed it by "word stuffing"—inserting multiple repetitions of desirable keywords, often in invisible text at the bottom of the web page. "The recurring refrain," says Kleinberg, "was that search doesn't work." But he had an intuition of a more effective approach. "One thing that was not being used at all was the fact that the web was a network," he says. "You could find people saying in the academic papers that links ought to be taken advantage of, but by 1996 it still hadn't been.'"

Kleinberg began to play around with ways to analyze links. Since he didn't have the assistance, the resources, the time, or the inclination, he didn't attempt to index the entire web for his link analysis. Instead he did a kind of prewash. He typed a query into AltaVista, took the first two hundred results, and then used that subset for his own search.

Interestingly, the best results for the query were often not included in those AltaVista solutions. For instance, if you typed in "newspaper," AltaVista would not give you links for The New York Times or The Washington Post. "That's not surprising, because AltaVista is about matching strings, and unless The New York Times happened to say, 'I'm a newspaper!' AltaVista is not going to find it," Kleinberg explains. But, he suspected, he'd have more luck if he checked out what those 200 sites pointed to. "Among those 200 people who were saying 'newspapers,' someone was going to point to The New York Times," he says. "In fact, a bunch of people were going to point to The New York Times, because among those 200 pages were some people who really liked to collect links for newspapers on the web. If you pulled in those links, and got a set of 5,000 to 10,000 of them, in a sense, you'd have a vote. The winner would be the one with the most in-links from the group." It was the same lightbulb that had brightened over Larry Page's head.

Sometime in December 1996, Kleinberg got the balance right. One of his favorite queries was "Olympics." The summer games had been held in Atlanta that year, and there were thousands of sites that in some way dealt with the athletic contests, the politics, the bomb that a domestic terrorist had planted. The AltaVista results for that keyword were riddled with spam and were generally useless. But Kleinberg's top result was the official Olympics site.

Kleinberg began showing his breakthrough around IBM. His managers quickly put him in touch with the patent lawyers. Most people took a
look at what Kleinberg had set up and wanted him to find stuff for them. Even the patent attorney wanted Kleinberg to help him find sources for his hobby, medieval siege devices. By February 1997, he says, “all sorts of IBM vice presidents were trooping through Almaden to look at demos of this thing and trying to think about what they could do with it.” Ultimately, the answer was... not much. IBM was a $70 billion business, and it was hard to see how a research project about links on this World Wide Web could make a difference. Kleinberg shrugged it off. He was going to teach computer science at Cornell.

Through mutual friends at Stanford, Kleinberg heard about Larry Page’s project, and in July 1997 they met at Page’s office in the Gates Building. Kleinberg was impressed with BackRub. “In academia, when there’s a hard problem everyone wants to solve, you’re always implicitly competing with the other people who are working on it,” says Kleinberg. But neither mentioned that issue. Kleinberg encouraged Page to publish his findings, but Page wasn’t receptive. “Larry was worried about writing a paper,” says Kleinberg. “He was wary because he wanted to see how far he could get with it while he refined it.”

Kleinberg could see that his goals were different from Page’s. “They wanted to crawl the whole web and get it on racks of servers that they would accumulate,” Kleinberg says. “My view was ‘How can I solve this problem without having to sink three months into indexing the web?’ We had the same core idea, but how we went about it was almost diametrically opposite.” Kleinberg was trying to understand network behavior. Page and Brin were building something. “Kleinberg had this notion of authority, where your page can become good just by linking to the right pages,” says Page. “Whereas what I was doing was more of a traffic simulation, which is actually how people might search the web.”

Kleinberg kept up with Google. He turned down job feelers in 1999 and again in 2000. He was happy at Cornell. He’d win teaching awards and a MacArthur fellowship. He led the life in academia he’d set out to lead, and not becoming a billionaire didn’t seem to bother him.

There was yet a third person with the idea, a Chinese engineer named Yanhong (Robin) Li. In 1987, he began his studies at Beijing University, an institution that claimed prominence in the country by way of a metric: The Science Citation Index, which ranked scientific papers by the number of other papers that cited them. The index was used in China to rank universities. “Beijing University, measured by the number of citations its professors got from their papers, was ranked number one,” said Li.

Li came to the United States in 1991 to get a master’s degree at SUNY Buffalo, and in 1994 took a job at IDD Information Services in Scotch Plains, New Jersey, a division of Dow Jones. Part of his job was improving information retrieval processes. He tried the search engines at the time—AltaVista, Excite, Lycos—and found them ineffectual and spam-ridden. One day in April 1996 he was at an academic conference. Bored by the presentation, he began to ponder how search engines could be improved. He realized that the Science Citation Index phenomenon could be applied to the Internet. The hypertext link could be regarded as a citation! “When I returned home, I started to write this down and realized it was revolutionary,” he says. He devised a search approach that calculated relevance from both the frequency of links and the content of anchor text. He called his system RankDex.

When he described his scheme to his boss at Dow Jones, urging the company to apply for a patent, he was at first encouraged, then disappointed when nothing happened. “So a couple of months later, I decided to write the application by myself.” He bought a self-help book on patent applications and filed his in June 1996. But when he told his boss, Dow Jones reasserted itself and hired a lawyer to review the patent, which it refiled in February 1997. (Stanford University would not file its patent for Larry Page’s PageRank system until January 1998.) Nonetheless, Dow Jones did nothing with Li’s system. “I tried to convince them it was important, but their business had nothing to do with Internet search, so they didn’t care,” he says.

Robin Li quit and joined the West Coast search company called Infoseek. In 1999, Disney bought the company and soon thereafter Li returned to China. It was there in Beijing that he would later meet—and compete with—Larry Page and Sergey Brin.

Page and Brin had launched their project as a stepping-stone to possible dissertations. But it was inevitable that they began to eye their creation as something that could make them money. The Stanford CS program was as much a corporate incubator as an academic institution. David Cheriton, one of the professors, once put it this way: “The unfair advantage that Stanford has over any other place in the known universe is that we’re surrounded by Silicon Valley.” It was not uncommon for its professors to...
straddle both worlds, maintaining posts in the department while playing in the high-tech scrum of start-ups striving for the big score. There was even a joke that faculty members couldn't get tenure until they started a company.

Cheriton himself was a prime example of how the Stanford network launched companies and enriched the founders. One of the earlier-gold strikes from Stanford was the founding of Sun Microsystems by a group that included Andy Bechtolsheim, Vinod Khosla, and Bill Joy. Cheriton was close to Bechtolsheim, so in 1995, when the latter decided to start Granite Systems, a networking start-up, the two collaborated. Eighteen months later, Cisco bought the company for $220 million.

Sergey Brin, Rollerblading his way around the corridors of Gates Hall, took notice. Though Brin and Page didn't have classes with Cheriton, they headed to his office for some advice. They specifically wanted to know how they might interest a company into using PageRank in its own search technology. Cheriton told them that it would be difficult—Sun Microsystems, he reminded them, had been started out of frustration when companies had spurned Bechtolsheim's attempts to sell his workstation technology.

Yet Brin and Page were reluctant at that point to strike out on their own. They had both headed to Stanford intending to become PhDs like their dads.

But licensing their search engine wasn't easy. Though Brin and Page had a good meeting with Yahoo founders Jerry Yang and David Filo, former Stanford students, Yahoo didn't see the need to buy search engine technology. They also met with an AltaVista designer, who seemed interested in BackRub. But the wise men back in DEC headquarters in Maynard, Massachusetts, nixed the idea. Not Invented Here.

Maybe the closest Page and Brin came to a deal was with Excite, a search-based company that had begun—just like Yahoo—with a bunch of sharp Stanford kids whose company was called Architektext before the venture capitalists (VCs) got their hands on it and degeekified the name. Terry Winograd, Sergey's adviser, accompanied them to a meeting with Vinod Khosla, the venture capitalist who had funded Excite.

That led to a meeting with Excite's founders, Joe Kraus and Graham Spencer, at Fuki Sushi, a Palo Alto restaurant. Larry insisted that the whole BackRub team come along. "He always likes to have more people on his side than the opposite side, to get the upper hand," says Scott Hassan, who attended along with Page, Brin, and Alan Steremberg. "They sent two people, so we had four." The Excite people began comparison tests with BackRub, plugging in search queries such as "Bob Marley." The results were a lot better than Excite's.

Larry Page laid out an elaborate plan, which he described in detail in emails to Khosla in January 1997. Excite would buy BackRub, and then Larry alone would go to work there. Excite's adoption of BackRub technology, he claimed, would boost its traffic by 10 percent. Extrapolating that in terms of increased ad revenue, Excite would take in $130,000 more every day, for a total of $47 million in a year. Page envisioned his tenure at Excite lasting for seven months, long enough to help the company implement the search engine. Then he would leave, in time for the fall 1997 Stanford semester, resuming his progress toward a doctorate. Excite's total outlay would be $1.6 million, including $300,000 to Stanford for the license, a $200,000 salary, a $400,000 bonus for implementing it within three months, and $700,000 in Excite stock. (Since Page and Brin were working for Stanford while developing their work, the school owned the PageRank patent. Stanford would commonly make financial arrangements so that such inventors could hold exclusive licenses to the intellectual property they created. Eventually Stanford did so with Google, in exchange for 1.8 million shares.) "With my help," wrote the not-quite-twenty-four-year-old student, "this technology will give Excite a substantial advantage and will propel it to a market leadership position."

Khosla made a tentative counteroffer of $750,000 total. But the deal never happened. Hassan recalls a key meeting that might have sunk it. Though Excite had been started by a group of Stanford geeks very much like Larry and Sergey, its venture capital funders had demanded they hire "adult supervision," the condescending term used when bratty geeks are pushed aside as top executives and replaced by someone more experienced and mature, someone who could wear a suit without looking as though he were attending his Bar Mitzvah. The new CEO was George Bell, a former Times Mirror magazine executive. Years later, Hassan would still laugh when he described the meeting between the BackRub team and Bell. When the team got to Bell's office, it fired up BackRub in one window and Excite in the other for a bake-off.

The first query they tested was "Internet." According to Hassan, Excite's first results were Chinese web pages where the English word "Internet" stood out among a jumble of Chinese characters. Then the team typed
"Internet" into BackRub. The first two results delivered pages that told you how to use browsers. It was exactly the kind of helpful result that would most likely satisfy someone who made the query.

Bell was visibly upset. The Stanford product was too good. If Excite were to host a search engine that instantly gave people information they sought, he explained, the users would leave the site instantly. Since his ad revenue came from people staying on the site—"stickiness" was the most desired metric in websites at the time—using BackRub's technology would be counterproductive. "He told us he wanted Excite's search engine to be 80 percent as good as the other search engines," says Hassan. And we were like, "Wow, these guys don't know what they're talking about."

Hassan says that he urged Larry and Sergey right then, in early 1997, to leave Stanford and start a company. "Everybody else was doing it," he says. "I saw Hotmail and Netscape doing really well. Money was flowing into the Valley. So I said to them, 'The search engine is the idea. We should do this.' They didn't think so. Larry and Sergey were both very adamant that they could build this search engine at Stanford."

"We weren't... in an entrepreneurial frame of mind back then," Sergey later said.

Hassan quit the project. He got a job with a new company called Alexa and worked part-time on a start-up called eGroups. In fact, Larry and Sergey—this was before they had gotten a dollar in funding for Google—pitched in $5,000 each to help him buy computers for eGroups. (The investment paid off less than three years later when Yahoo bought eGroups for an estimated $413 million.)

But for the next year and a half, all the companies they approached turned them down. "We couldn't get anyone interested," says Page. "We did get offers, but they weren't for much money. So we said, 'Whatever,' and went back to Stanford to work on it some more. It wasn't like we wanted a lot of money, but we wanted the stuff to get really used. And they would want us to work there and we'd ask, 'Do we really want to work for this company?' These companies weren't going to focus on search—they were becoming portals. They didn't understand search, and they weren't technology people."

In September 1997, Page and Brin renamed BackRub to something they hoped would be suitable for a business. They gave serious consideration to "The Whatbox," until they realized that it sounded too much like "webbox," which wasn't family-friendly. Then Page's dorm roommate suggested they call it "googol." The word was a mathematical term referring to the number 1 followed by 100 zeros. Sometimes the word "googolplex" was used generically to refer to an insanely large number. "The name reflected the scale of what we were doing," Brin explained a few years later. "It actually became a better choice of name later on, because now we have billions of pages and images and groups and documents, and hundreds of millions of searches a day." Page misspelled the word, which was just as well since the Internet address for the correct spelling was already taken. "Google" was available. "It was easy to type and memorable," says Page.

One night, using a new open-source graphics program called GIMP, Sergey designed the home page, spelling the new company name in different colors, making a logo that resembled something made from children's blocks. It conveyed a sense of amiable whimsy. He put an exclamation point after the name, just like Yahoo, another Internet company founded by two Stanford PhD dropouts. "He wanted it to be playful and young," says Page. Unlike a lot of other web pages, the Google home page was so spare it looked unfinished. The page had a box to type in requests and two buttons underneath, one for search and another labeled I'm Feeling Lucky, a startling bid of confidence that implied that, unlike the competition, Google was capable of nailing your request on the first try. (There was another reason for the button. "The point of I'm Feeling Lucky was to replace the domain name system for navigation," Page said in 2002. Both Page and Brin hoped that instead of guessing what was the address of their web-destination, they'd just "go to Google." The next day Brin ran around the CS department at Stanford, showing off his GIMP creation. "He was asking everybody whether it made any sense to put other stuff on the page," says Dennis Allison, a Stanford CS lecturer. "And everybody said no." That was fine with Page and Brin. The more stuff on the page, the slower it would run, and both of them, especially Page, believed that speed was of the essence when it came to pleasing users. Page later found it humorous that people praised the design for its Zen-like use of white space. "The minimalism is that we didn't have a webmaster and had to do it ourselves," he says.

Meanwhile, BackRub-turned-Google was growing to the point where it was difficult to run using Stanford's facilities. It was becoming less a research project than an Internet start-up run from a private university. Page and Brin's reluctance to write a paper about their work had become notorious in the department. "People were saying, 'Why is this so secret? This is an academic project, we should be able to know how it worked,'" says Terry Winograd.

Page, it seemed, had a conflict about information. On one hand, he
subscribed heartily to the hacker philosophy of shared knowledge. That was part of what his project was all about: making human knowledge accessible, making the world a better place. But he also had a strong sense of protecting his hard-won proprietary information. He remembered Nikola Tesla, who had died in poverty even as his inventions enriched others. Later, there would be speculation whether Page, a private person to begin with, had pulled back a little more after his father’s death in June 1996.

Scott Hassan recalls that the team conveyed its condolences to Page that month, but Hassan didn’t speak much about the loss with Page. “Mostly we talked about technical stuff,” he would recall. Mike Moritz, one of the venture capitalists who would fund Google, later surmised that “a large part” of Page’s later wariness could be associated with that loss. “He felt that the world was pulled out from underneath him,” Moritz said. “It makes it hard to trust anything again.”

But it wasn’t just the secrecy that stalled Brin and Page. Writing a paper wasn’t as interesting to them as building something. “Inherently, Larry and Sergey aren’t paper-oriented—they’re product-oriented,” says Winograd. “If they have another ten minutes, they want to make something better. They don’t want to take ten minutes to tell you something they did.” But finally Winograd convinced them to explain PageRank in a public forum. They presented a paper called “The Anatomy of a Large-Scale Hypertextual Web Search Engine” at a conference in Australia in May 1998.

Arthur Clarke once remarked that the best technology was indistinguishable from magic. The geeks of Silicon Valley, assuming he was talking about them, have never forgotten that and have invoked the quote in countless press releases about their creations. But Google search really did feel like magic. At Stanford, Larry’s and Sergey’s professors and friends were using the search engine to answer questions and telling their friends about it. Google was handling as many as 10,000 queries a day. At times it was consuming half of Stanford’s Internet capacity. Its appetite for equipment and bandwidth was voracious. “We just begged and borrowed,” says Page. “There were tons of computers around, and we managed to get some.” Page’s dorm room was essentially Google’s operations center, with a motley assortment of computers from various manufacturers stuffed into a homemade version of a server rack—a storage cabinet made of Legos. Larry and Sergey would hang around the loading dock to see who on campus was getting computers—companies like Intel and Sun gave lots of free machines to Stanford to curry favor with employees of the future—and then the pair would ask the recipients if they could share some of the bounty.

That still wasn’t enough. To store the millions of pages they had crawled, the pair had to buy their own high-capacity disk drives. Page, who had a talent for squeezing the most out of a buck, found a place that sold refurbished disks at prices so low—a tenth of the original cost—that something was clearly wrong with them. “I did the research and figured out that they were okay as long as you replaced the [disk] operating system,” he says. “We got 120 drives, about nine gigs each. So it was about a terabyte of space.” It was an approach that Google would later adopt in building infrastructure at low cost.

Larry and Sergey would be sitting by the monitor, watching the queries—at peak times, there would be a new one every second—and it would be clear that they’d need even more equipment. What next? they’d ask themselves. Maybe this is real.

Stanford wasn’t kicking them out—the complications of running the nascent Google were outweighed by pride that something interesting was brewing in the department. “It wasn’t like our lights were dimming when they would run the crawler,” says Garcia-Molina, who was still hoping that Larry and Sergey would develop their work academically. “I think it would have made a great thesis,” he says. “I think their families were behind them to get PhDs, too. But doing a company became too much of an attraction.”

There was no alternative; no one would pay enough for Google. And the happy visitors they were attracting gave them confidence that their efforts could make a difference. After years of dreaming how his ideas could change the world, Larry Page realized that he’d done something that might do just that. “If the company failed, too bad,” says Page. “We were really going to be able to do something that mattered.”

They went back to Dave Cheriton, who encouraged them to just get going. “Money shouldn’t be a problem,” he said. Cheriton suggested that they meet with Andy Bechtolsheim. Brin dashed off an email to Bechtolsheim that evening around midnight and got an immediate reply asking if the two students could show up at eight the next morning at Cheriton’s house, which was on the route Bechtolsheim used to go to work each day. At that ungodly hour Page and Brin demoed their search engine for Bechtolsheim on Cheriton’s porch, which had an ethernet connection. Bechtolsheim, impressed but eager to get to the office, cut the meeting short by offering to write the duo a $100,000 check.

“We don’t have a bank account yet,” said Brin.
“Deposit it when you get one,” said Bechtolsheim, who raced off in his Porsche. With as little fanfare as if he were grabbing a latte on the way to work, he had just invested in an enterprise that would change the way the world accessed information. Brin and Page celebrated with a Burger King breakfast. The check remained in Page's dorm room for a month.

Soon afterward, Bechtolsheim was joined by other angel investors, including Dave Cheriton. One was a Silicon Valley entrepreneur named Ram Shriram, whose own company had recently been purchased by Amazon.com. Shriram had met Brin and Page in February 1998; although he had been skeptical about a business model for search engines, he was so impressed with Google that he had been advising them. After the Bechtolsheim meeting, Shriram invited them to his house to meet his boss Jeff Bezos, who was enthralled with their passion and “healthy stubbornness,” as they explained why they would never put display ads on their home page. Bezos joined Bechtolsheim, Cheriton, and Shriram as investors, making for a total of a million dollars of angel-money.

On September 4, 1998, Page and Brin filed for incorporation and finally moved off campus. Sergey's girlfriend at the time was friendly with a manager at Intel named Susan Wojcicki, who had just purchased a house on Santa Margarita Street in Menlo Park with her husband for $615,000. To help meet the mortgage, the couple charged Google $1,700 a month to rent the garage and several rooms in the house. At that point they'd taken on their first employee, fellow Stanford student Craig Silverstein. He'd originally connected with them by offering to show them a way to compress all the crawled links so they could be stored in memory and run faster. (“It was basically to get my foot in the door,” he says.) They also hired an office manager. But almost as if they were still hedging on their PhDs, they maintained a presence at Stanford that fall, co-teaching a course, CS 349, “Data Mining, Search, and the World Wide Web,” which met twice a week that semester. Brin and Page announced it as a “project class” in which the students would work with the repository of 25 million web pages that they had captured as part of what was now a private company: They even had a research assistant. The first assigned reading was their own paper, but later in the semester a class was devoted to a comparison of PageRank and Kleinberg's work.

In December, after the final projects were due, Page emailed the students a party invitation that also marked a milestone: “The Stanford Research Project is now Google.com: The Next Generation Internet Search Company.”

“Dress is Tiki Lounge wear,” the invitation read, “and bring something for the hot tub.”

Larry Page did not want to be Tesla'd. Google had quickly become a darling of everyone who used it to search the net. But at first so had AltaVista, and that search engine had failed to improve. How was Google, led by two talented but inexperienced youngsters, going to tackle the devilishly difficult problems of improving its service?

“If we aren’t a lot better next year, we will already be forgotten,” Page said to one of the first reporters visiting the company.

The web was growing like digital kudzu. People were coming to Google in droves. Google's plan was to get even more traffic. “When we started the company, we had two computers,” says Craig Silverstein. “One was the web service, and one was doing everything else—the page rank, the searches. And there was a giant chain of disks that went off the back of the computer that stored twenty-five million web pages. Obviously that was not going to scale very well.” Getting more computers was no problem. Google needed brainpower, especially since Brin and Page had reached the limits of what they could do in writing the software that would enable the search engine to grow and improve. “Coding is not where their interests are,” says Silverstein.

The founders also knew that Google had to be a lot smarter to keep satisfying users—and to fulfill the world-changing ambitions of its founders. “We don't always produce what people want,” Page explained in Google's early days. “It's really difficult. To do that you have to be smart—you have to understand everything in the world. In computer science, we call that artificial intelligence.”

Brin chimed in. “We want Google to be as smart as you—you should be getting an answer the minute you think of it.”

“The ultimate search engine,” said Page. “We're a long way from that.”

Page and Brin both held a core belief that the success of their company would hinge on having world-class engineers and scientists committed to their ambitious vision. Page believed that technology companies
can thrive only by "an understanding of engineering at the highest level." Somehow Page and Brin had to identify such a group and impress them enough to have them sign on to a small start-up. Oh, and they had a policy that limited the field: no creeps. They were already thinking of the culture of their company and making sure that their hires would show traits of hard-core wizardry, user focus, and starry-eyed idealism.

"We just hired people like us," says Page.

Some of Google's early hires were simply brainy recent grads, people like Marissa Mayer, a hard-driving math whiz and ballet dancer in her high school in Wausau, Wisconsin, who had become an artificial intelligence star at Stanford. (During her interview with Silverstein, she was asked for three things Google could do better; ten years later, she was still kicking herself that she listed only two.) But Page and Brin also went after people with résumés more often seen in the recruitment offices of Microsoft Research or Carnegie Mellon's CS department. One of their first coups was a professor at the University of California at Santa Barbara named Urs Hölzle. He'd played with the earlier crop of search engines such as AltaVista and Inktomi and concluded that, as a computer scientist familiar with Boolean syntax and other techniques, he could use those techniques to find what he wanted on the Internet. But he assumed that search would never be something his mother would use. Google instantly changed his mind about that: you just typed in what you wanted, and, bang, the first thing was right. Mom would like that! "They definitely seemed to know what they were doing," he says of Larry and Sergey.

More important to him, when he visited the new company in early 1999, he understood that though he had no background in information retrieval, the problems Brin and Page were working on had a lot in common with his own work in big computer systems. This little search engine was butting up against issues in performance and scalability that only huge projects had previously grappled with. That was Google's secret weapon to lure world-class computer scientists: in a world where corporate research labs were shutting down, this small start-up offered an opportunity to break ground in computer science.

Hölzle, still wary, accepted the offer but kept his position at UCSB by taking a yearlong leave. He would never return. In April he arrived at Google with Yoshiba, a big floppy Leonberger dog, in tow, and dived right in to help shore up Google's overwhelmed infrastructure. (By then Google had moved from Wojcicki's Menlo Park house to a second-floor office over a bicycle shop in downtown Palo Alto.) Though Google had a hundred computers at that point—it was buying them as quickly as it could—it could not handle the load of queries. Hundreds of thousands of queries a day were coming in.

The average search at that time, Hölzle recalls, took three and a half seconds. Considering that speed was one of the core values of Page and Brin—it was like motherhood, and scale was apple pie—this was a source of distress for the founders. "Basically during the middle of the day we were maxed out," says Hölzle. "Nothing was happening for some users, because it would just never get a page basically back. It was all about scalability performance improvements." Part of the problem was that Page and Brin had written the system in what Hölzle calls "university code," a nice way of saying amateurish. "The web server couldn't handle more than ten requests or so a second because it was written in Python, which is a great idea for a research system, but it's not a high-performance solution," he says. He immediately set about rewriting the code.

Hölzle was joined by other computer scientists who were more daring in taking the leap to permanent Google employment. This included a minimmigration of engineers from DEC's research division. Established legend in Silicon Valley cited Xerox's Palo Alto Research Center (PARC) as the canonical lab brimming with breakthrough innovation that had been misunderstood, buried, or otherwise fumbled by the clueless parent company. (Its inventions included the modern computer interface with windows and file folders.) But when it came to missed opportunities, PARC had nothing on DEC's Western Research Laboratory, which was hauled over to Compaq when that personal computer company bought Digital Equipment Corporation in 1998. (In 2002, Hewlett-Packard would acquire Compaq.) In 1998, two years before Apple even began work on the iPod, DEC engineers were developing a digital music player that could store a whole music collection and fit in your pocket. In addition, DEC had some of the founding fathers of the Internet, as well as scientists writing pioneering papers on network theory. But DEC never used its engineers' ideas to help AltaVista become Google. ("From the moment I left DEC, I never used AltaVista," says Louis Monier, who split in 1998. "It was just pathetic. It was completely obvious that Google was better.") So it was little wonder that some of them went to Google. "The number [of former DEC scientists at Google] is really kind of staggering," says Bill Weihl, a DEC refugee who came to the company in 2004.

One of the DEC engineers had already independently discovered the power of web links in search. Jeffrey Dean suspected that it would be help-
ful to web users if a software program could point them to pages that were related to the ones they liked. In his vision, you would be reading an article in The New York Times and his program would pop up, asking if you’d like to see ten other interesting pages related to the one you were reading.

Dean had never been much interested in information retrieval. Now that he suspected a revolution was afoot, he was. But his attempts to join up with the AltaVista crew ended ignominiously. “The AltaVista team had grown really fast,” he says; “and hired a bunch of people who I think were not as technically good as they could have been.” In other words— get me away from here. In February 1999, Dean bailed from DEC to join a start-up called mySimon.

Within a few months, though, he was bored. Then he heard that Urs Hölzle, whom he’d known through his grad school adviser, had joined up with the guys who did PageRank. “I figured Google would be better because I knew more of the people there, and they seemed like they were more technically savvy,” he says. He was so excited about working there that even though his official starting date wasn’t until August 1999, in July he began coming to Google after his workday at mySimon ended.

Dean’s hiring got the attention of another DEC researcher, Krishna Bharat. He had also been thinking of ways to get web search results from links. Bharat was working on something called the Hilltop algorithm, which algorithmically identified “expert sites” and used those to point to the most relevant results. It was something like Jon Kleinberg’s hub approach, but instead of using AltaVista as a prewash to get top search results and then figure out who the expert sites were, Bharat went straight to a representation of the web—links and some bits from the pages—stored in computer memory. Bharat’s algorithms would roam around the “neighborhood of the query” to find the key sites.

The India-born computer scientist had already been on Google’s radar: when he ate lunch at a joint called World Wraps in Palo Alto, he’d run into Sergey Brin, who would invariably hand him a business card and urge him to apply to Google. Bharat was impressed with Google—he’d actually presented his Hilltop algorithm in the same session at the conference in Australia when Brin and Page showed off Google to a bowled-over audience of IR people. He also liked Sergey. Their mutual friend Rajeev Mowani once hosted a seminar where Brin had arrived on Rollerblades and began rhapsodizing about PageRank without missing a beat. Bharat thought that was incredibly cool. But Google was so small. It was hard for

Bharat to imagine leaving the creature comforts of a big company for an operation with a single-digit workforce located over a bicycle shop and decorated in a style that mixed high-tech Dumpster with nursery school. Pets he cherished the ability to pursue research, something he doubted was possible at a tiny start-up.

Then Google hired Jeff Dean, and Bharat was stunned. It was like some basketball team playing in an obscure minor league—grabbing a player who was first-round NBA material. Those guys were serious. Soon after, Bharat heard that this just-born start-up, which could barely respond to its query traffic, was starting a research group! It sounded improbable, but he climbed the flight of stairs in the Palo Alto office for an interview. Bharat said straight out that he was skeptical of Google’s research ambitions. From what he could see, there were a lot of people running around with pagers and flicking at their keyboards to keep the system going. “Larry, why do you say you want to do research?” he said to Page. “You are such a tiny group!” Page’s answer was surprising and impressive. Looking at things from a different perspective could lead to unexpected solutions, he said. Sometimes in engineering you look at things with tunnel vision and need a broader perspective. He told Bharat a story about Kodak that involved some seemingly intractable practical problem that was solved by an unexpected intervention from someone in the research division. Page wanted that kind of thing to happen at Google.

That interaction sold Bharat. Here was a guy who was young, inexperienced, and probably half nuts—but technically adept and infectiously confident. “I could respect Larry in a way that I couldn’t respect people running other start-ups,” says Bharat. “I knew the technical content of his work.” What’s more, Bharat could feel the pull of Page’s crusade to make the world better by cracking hard problems at the intersection of computer science and metaphysics. Bharat had thought a lot about search and was enthralled with its mysteries. On the face of things, it seemed so tantalizingly easy. But people had grasped only the slightest fraction of what was possible. To make progress, even appreciate this space, you would have to live in the data, breathe them in like a fish passing water through its gills. Here was his invitation. Bharat would wind up working an evolution of his Hilltop algorithm, called web connectivity analysis, into Google’s search engine. It would be the company’s first patent.

The same almost mystical attraction of Google’s ambitions led to another impressive hire in early 2000: Anurag Acharya, a Santa Barbara
professor who was a colleague of Höflze. Acharya, who'd gotten his PhD at Carnegie Mellon, had spent his entire life in academia but at age thirty-six had been questioning his existence there. He had tired of a routine where people took on a problem of limited scope, solved it, published the results, and then went on to the next. He remembered when he'd been a student and had sat with his adviser, a deep thinker who spent his entire life grappling with a single giant mystery: what is the nature of mind? More and more, Acharya thought that there was beauty in grappling with a classically hard problem that would survive after you leave the earth. Talking to Höflze during an interview for this little company, he realized that search was that kind of problem. "I had no background in search but was looking for a problem of that kind," he says. "It appeared that, yes, that could be it." Adding to Google's appeal was his own background—like several of his new colleagues, he was from provincial India. (And like many at Google, including the founders, his parents were academics.) He often thought of the people in his home country, who were not just poor but information-impoverished as well. "If you were successful at Google, people from everywhere would have the ability to find information," he says. "I come from a place where those boundaries are very, very apparent. They are in your face. To be able to make a dent in that is a very attractive proposition."

Bharat recommended another friend named Ben Gomes, who worked at Sun. The two had studied for exams together as high school friends in Bangalore, India. Gomes joined Google the same week Bharat did. And Bharat had another friend who was among the best catches of all: Amit Singhal.

Born in the Indian state of Uttar Pradesh, in the foothills of the Himalayas, Singhal had arrived in the United States in 1992 to pursue a master's degree in computer science at the University of Minnesota. He'd become fascinated with the field then known as information retrieval and was desperate to study with its pioneering innovator, Gerard Salton. "I only applied to one grad school, and it was Cornell," he says. "And I wrote in my statement of purpose that if I was ever going to get a PhD, it's with Gerry Salton. Otherwise, I didn't think a PhD was worth it." He became Salton's assistant, got his PhD at Cornell, and eventually wound up at AT&T Labs.

In 1999, Singhal ran into Bharat at a conference in Berkeley. Bharat told him he was leaving DEC for an exciting start-up that wanted to take on the biggest problems in search. It had a funny name, Google. Singhal should work there, too. Singhal thought the idea was ridiculous. Maybe it was all right for Bharat, who was a couple of years younger and unmarried.

But Singhal had a wife and daughter and a second child on the way. "These little companies are all going to die," he said. "I work for AT&T—the big ship that always sails. I can't go to Google-smoogule because I have a family to support."

Not too long afterward, the big ship AT&T began to take on water. "In 2000, I was here," says Singhal.

In barely a year since Brin and Page had formed their company, they had gathered a group of top scientists totally committed to the vision of their young founders. These early employees would be part of team efforts that led to innovation after innovation that would broaden Google's lead over its competitors and establish it as synonymous with search. But those breakthroughs were in the future. In 2000, those big brains were crammed into a single conference room working on an emergency infrastructure fix. Google had taken ill.

The problem was the index storing the contents of the web in Google's servers. For a couple of months in early 2000, it wasn't updating at all. Millions of documents created during that period weren't being collected. As far as the Google search engine was concerned, they didn't exist.

The problem was a built-in flaw in the crawling and indexing process. If one of the machines devoted to crawling broke down before the process was completed, indexing had to begin from scratch. It was like a role-playing computer game in which you would spend hundreds of hours building a character and then lose all that effort if your character got killed by a stray beast or a well-armed foe. The game world had learned to deal with the problem—dead avatars could be resurrected after a brief pause or an annoying dislocation. But Google hadn't.

The flaw hadn't been so bad in the earlier days of Google, when only five or so machines were required to crawl and index the web. It was at least a ten-day process with one of Google's first crawl engineers, Harry Cheung (everyone called him Spider-Man), at his machines, monitoring progress of spiders as they spread out through the net and then, after the crawl, breaking down the web pages for the index and calculating the page rank, using Sergey's complicated system of variables with a mathematical process using something called eigenvectors, while everybody waited for the two processes to converge. ("Math professors love us because Google has made eigenvectors relevant to every matrix algebra student in America," says Marissa Mayer.) Sometimes, because of quirks in the way the web addresses were numbered, the system crawled the same pages and showed no move-
ment, and then you'd have to figure out whether you were actually done or had hit a black hole. This problem, though, had been generally manageable.

But as the web kept growing, Google added more machines—by the end of 1999, there were eighty machines involved in the crawl (out of a total of almost three thousand Google computers at that time)—and the likelihood that something would break increased dramatically. Especially since Google made a point of buying what its engineers referred to as “el cheapo” equipment. Instead of commercial units that carefully processed and checked information, Google would buy discounted consumer models without built-in processes to protect the integrity of data.

As a stopgap measure, the engineers had implemented a scheme where the indexing data was stored on different hard drives. If a machine went bad, everyone's pager would start buzzing, even if it was the middle of the night, and they'd barrel into the office immediately to stop the crawl, copy the data, and change the configuration files. “This happened every few days, and it basically stopped everything and was very painful,” says Sanjay Ghemawat, one of the DEC research wizards who had joined Google.

“The whole thing needed rethinking,” says Jeff Dean.

Actually, it needed redoing, since by 2000 the factors impeding the crawl were so onerous that after several attempts it looked as though Google would never build its next index. The web was growing at an amazing pace, with billions of more documents each year. The presence of a search engine like Google actually accelerated the pace, offering an incentive to people as they discovered that even the quirkiest piece of information could be accessed by the small number of people who would appreciate it. Google was trying to contain this tsunami with more machines—cheap ones, thus increasing the chance of a breakdown. The updates would work for a while, then fail. And now, week after week, the updates were passing before the indexes were updated.

It's hard to overestimate the seriousness of this problem. One of the key elements of good search was freshness—making sure that the indexes have recent results. Imagine if this problem had happened a year later, after the September 11, 2001, terrorist attacks. Doing a Google search for “World Trade Center” that November or December, you would have found no links to the event. Instead, you'd have results that suggested a fine-dining experience at Windows on the World, on the 107th floor of the now-nonexistent North Tower.

A half-dozen engineers moved their computers into a conference room. Thus Google created its first war room. (By then—less than a year after moving from the house in Menlo Park to the downtown Palo Alto office—Google had moved once again, to a roomier office-park facility on Bayshore Road in nearby Mountain View. Employees dubbed it the Googoplex, a pun on the mathematical term googolplex, meaning an unthinkably large number.) When people came to work, they'd go to the war room instead of the office. And they'd stay late. Dean was in there with Craig Silverstein, Sanjay Ghemawat, and some others.

They built a system that implemented “checkpointing,” a way for the index to hold its place if a calamity befell a server or hard disk. But the new system went further—it used a different way to handle a cluster of disks, more akin to the parallel-processing style of computing (where a computational task would be split among multiple computers or processors) than the “sharding” technique Google had used, which was to split up the web and assign regions of it to individual computers. (Those familiar with computer terms may know this technique as “partitioning,” but, as Dean says, “everyone at Google calls it sharding because it sounds cooler.” Among Google's infrastructure wizards, it's key jargon.)

The experience led to an ambitious revamp of the way the entire Google infrastructure dealt with files. “I always had wanted to build a file system, and it was pretty clear that this was something we were going to have to do,” says Ghemawat, who led the team. Though there had previously been systems that handled information distributed over multiple files, Google's could handle bigger data loads and was more nimble at running full speed in the face of disk crashes—which it had to be, because, with Google's philosophy of buying supercheap components, failure was the norm. “The main idea was that we wanted the file system to automate dealing with failures, and to do that, the file system would keep multiple copies and it would make new copies when some copy failed,” says Ghemawat.

Another innovation that came a bit later was called the in-RAM system. This involved putting as much of the index as possible in actual computer memory as opposed to the pokier, less reliable hard disk drives. It sped things up considerably, allowed more flexibility, and saved money. “The in-memory index was like, a factor of two or three cheaper, because it could just handle many, many more queries per machine per second,” says Dean.

The system embodied Google's approach to computer science. At one point, the cost of fixed memory (in chips as opposed to spinning hard disks) would have been so expensive that using it to store the Internet would have been a daft concept. But Google's engineers knew that the pace of tech-
nology would drive prices down, and they designed accordingly. Likewise, Google—as its very name implies—is geared to handling the historic expansion of data that the digital revolution has triggered. Competitors, especially those who were successful in a previous age, were slow to wrap their minds around this phenomenon, while Google considered it as common as air. “The unit of thinking around here is a terabyte,” said Google engineering head Wayne Rosing in 2003. (A terabyte is equal to around 10 trillion bits of data.) A thirty-year Silicon Valley veteran whose résumé boasted important posts at DEC, Apple, and Sun, Rosing had joined Google in 2001 in part because he saw that it had the potential to realize the vision of Vannevar Bush’s famous memex paper, which he had read in high school. “It doesn’t even get interesting until there’s more than many terabytes involved in problems. So that drives you into thinking of hundreds of thousands of computers as the generic way to solve problems.” When you have that much power to solve problems, you have the ability to do much more than solve them faster. You can tackle problems that haven’t even been considered. You can build your own paradigms.

Implementing the Google File System was a step toward that new paradigm. It was also a timely development, because the demands on Google’s system were about to increase dramatically. Google had struck a deal to handle all the search traffic of Yahoo, one of the biggest portals on the web. The deal—announced on June 26, 2000—was a frustrating development to the head of Yahoo’s search team, Udi Manber. He had been arguing that Yahoo should develop its own search product (at the time, it was licensing technology from Inktomi), but his bosses weren’t interested. Yahoo’s executives, led by a VC-approved CEO named Timothy Koogle (described in a BusinessWeek cover story as “The Grown-up Voice of Reason at Yahoo”), instead were devoting their attention to branding—marketing gimmicks such as putting the purple corporate logo on the Zamboni machine that swept the ice between periods of San Jose Sharks hockey games. “I had six people working on my search team,” Manber said. “I couldn’t get the seventh. This was a company that had thousands of people. I could not get the seventh.” Since Yahoo wasn’t going to develop its own search, Manber had the task of finding the best one to license.

After testing Google and visiting Larry Page several times, Manber recommended that Yahoo use its technology. One concession that Yahoo gave Google turned out to be fateful: on the results page for a Yahoo search, the user would see a message noting that Google was powering the search. The page even had the Google logo. Thus Yahoo’s millions of users discovered a search destination that would become part of their lives.

As part of the deal, Google agreed to update its index on a monthly basis, something possible after the experience in the war room. Google now had the most current data in the industry. It also boasted the biggest index; on the day it announced the Yahoo deal, Google reported that its servers now held more than a billion web pages. This system remained state of the art until the summer of 2003, when Google launched a revamp of its entire indexing system to enable it to refresh the index from day to day, crawling popular sites more often. The code name for the 2003 update was BART. The title implied that Google’s system would match the aspirations (if not the accomplishments) of the local mass transit system: “always on time, always fast, always on schedule.” But the code name’s actual origin was an engineer named Bart.

Even though Google never announced when it refreshed its index, there would invariably be a slight rise in queries around the world soon after the change was implemented. It was as if the global subconscious realized that there were fresher results available.

The response of Yahoo’s users to the Google technology, though, was probably more conscious. They noticed that search was better and used it more. “It increased traffic by, like, 50 percent in two months,” Manber recalls of the switch to Google. But the only comment he got from Yahoo executives was complaints that people were searching too much and they would have to pay higher fees to Google.

But the money Google received for providing search was not the biggest benefit. Even more valuable was that it now had access to many more users and much more data. It would be data that took Google search to the next level. The search behavior of users, captured and encapsulated in the logs that could be analyzed and mined, would make Google the ultimate learning machine.

Amit Patel first realized the value of Google’s logs. Patel was one of Google’s very first hires, arriving in early 1999 as a part-timer still working on his Stanford CS PhD. Patel was studying programming language theory but realized he didn’t like the subject too much. (Unlike his bosses, though, he would complete his degree.) Google seemed more fun, and fun was important for Patel, a cherub-faced lover of games and distractions whose business card reads “Troublemaker.” One of his first projects at Google
turned out to be more significant than anyone expected. "Go find out how many people are using Google, who's using it, and what they're doing with it," he was told.

The task appealed to Patel, who was only beginning to learn about search engines and data analysis. He realized that Google could be a broad sensor of human behavior. For instance, he noticed that homework questions spiked on weekends. "People would wait until Sunday night to do their homework, and then they'd look up things on Google," he says. Also, by tracking what queries Google saw the most, you could get a glimpse in real time of what the world was interested in. (A few years later, Patel would be instrumental in constructing the Google Zeitgeist, an annual summation of the most popular search subjects that Google would release to the public at the end of the year.)

But the information that users provided to Google went far beyond the subject matter of their queries. Google had the capacity to capture everything people did on the site on its logs, a digital trail of activities whose retention could provide a key to future innovations. Every aspect of user behavior had a value. How many queries were there, how long were they, what were the top words used in queries, how did users punctuate, how often did they click on the first result, who had referred them to Google, where they were geographically. "Just basic knowledge," he recalls.

Those logs told stories. Not only when or how people used Google but what kind of people the users were and how they thought. Patel came to realize that the logs could make Google smarter, and he shared log information with search engineers such as Jeff Dean and Krishna Bharat, who were keenly interested in improving search quality.

To that point, Google had not been methodical about storing the information that told it who its users were and what they were doing. "In those days the data was stored on disks which were failing very often, and those machines were often repurposed for something else," says Patel. One day, to Patel's horror, one of the engineers pointed to three machines and announced that he needed them for his project and was going to reformat the disks, which at that point contained thousands of query logs. Patel began working on systems that would transfer these data to a safe place. As Google began to evolve a distribution of labor, eventually it mandated that at least one person be working on the web server, one on the index, and one on the logs.

Some years earlier, an artificial intelligence researcher named Doug-

Las Lenat had begun Cyc, an incredibly ambitious effort to teach computers all the common sense knowledge understood by every human. Lenat hired students to painstakingly type in an endless stream of even the most mundane truisms: a house is a building... people live in houses... houses have front doors... houses have back doors... houses have bedrooms and a kitchen... if you light a fire in a house, it could burn down—millions of pieces of information that a computer could draw upon so that when it came time to analyze a statement that mentioned a house, the computer could make proper inferences. The project never did produce a computer that could process information as well as a four-year-old child.

But the information Google began gathering was far more voluminous, and the company received it for free. Google came to see that instant feedback as the basis of an artificial intelligence learning mechanism. "Doug Lenat did his thing by hiring these people and training them to write things down in a certain way," says Peter Norvig, who joined Google as director of machine learning in 2001. "We did it by saying 'Let's take things that people are doing naturally.'"

On the most basic level, Google could see how satisfied users were. To paraphrase Tolstoy, happy users were all the same. The best sign of their happiness was the "long click"—this occurred when someone went to a search result, ideally the top one, and did not return. That meant Google had successfully fulfilled the query. But unhappy users were unhappy in their own ways. Most telling were the "short clicks" where a user followed a link and immediately returned to try again. "If people type something and then go and change their query, you could tell they aren't happy," says Patel. "If they go to the next page of results, it's a sign they're not happy. You can use those signs that someone's not happy with what we gave them to go back and study those cases and find places to improve search."

Those logs were tutorials on human knowledge. Google's search engine slowly built up enough knowledge that the engineers could confidently allow it to choose when to swap out one word for another. What helped make this possible was Google's earlier improvement in infrastructure, including the techniques that Jeff Dean and Sanjay Ghemawat had developed to compress data so that Google could put its index into computer memory instead of on hard disks. That was a case where a technical engineering project meant to speed up search queries enabled a totally different kind of innovation. "One of the big deals about the in-memory index is that it made it much more feasible to take a three-word query and say, 'I
want to look at the data for fifteen synonymous words, because they’re all kind of related,” says Dean. “You could never afford to do that on a disk-based system, because you’d have to do fifteen disk seeks instead of three, and it would blow up your serving costs tremendously. An in-memory index made for much more aggressive exploration of synonyms and those kinds of things.”

“We discovered a very early nifty thing,” says search engineer Amit Singhal, who worked hard on synonyms. “People change words in their queries. So someone would say, ‘Pictures of dogs,’ and then they’ll say ‘Pictures of puppies.’ That said that maybe dogs and puppies were interchangeable. We also learned that when you boil water, it’s hot water. We were learning semantics from humans, and that was a great advance.”

Similarly, by analyzing how people retracted their steps after a misspelling, Google devised its own spell checker. It built that knowledge into the system; if you typed a word inaccurately, Google would give you the right results anyway.

But there were obstacles. Google’s synonym system came to understand that a dog was similar to a puppy and that boiling water was hot. But its engineers also discovered that the search engine considered that a hot dog was the same as a boiling puppy. The problem was fixed, Singhal says, by a breakthrough late in 2002 that utilized Ludwig Wittgenstein’s theories on how words are defined by context. As Google crawled and archived billions of documents and web pages, it analyzed which words were close to each other. “Hot dog” would be found in searches that also contained “bread” and “mustard” and “baseball games” — not “puppies with roasting fur.” Eventually the knowledge base of Google understood what to do with a query involving hot dogs — and millions of other words. “Today, if you type ‘Gandhi bio,’ we know that ‘bio’ means ‘biography,’” says Singhal. “And if you type ‘bio warfare,’ it means ‘biological.’”

Over the years, Google would make the data in its logs the key to evolving its search engine. It would also use that data on virtually every other product the company would develop. It would not only take note of user behavior in its released products but measure such behavior in countless experiments to test out new ideas and various improvements. The more Google’s system learned, the more new signals could be built into the search engine to better determine relevance.

Sergey Brin had written the original part of the Google search engine that dealt with relevance. At that point it was largely based on PageRank, but as early as 2000 Amit Singhal realized that as time went on, more and more interpretive signals would be added, making PageRank a diminishing factor in determining results. (Indeed, by 2009, Google would say it made use of more than two hundred signals — though the real number was almost certainly much more — including synonyms, geographic signals, freshness signals, and even a signal for websites selling pizzas.) The code badly needed a rewrite; Singhal couldn’t even stand to read the code that Brin had produced. “I just wrote new,” he says.

Singhal completed a version of the new code in two months and by January 2001 was testing it. Over the next few months, Google exposed it to a percentage of its users and liked the results. They were happier. Sometime that summer, Google flipped the switch and became a different, more accurate service. In accordance with the company’s fanatical secrecy on such matters, it made no announcement. Five years later, Singhal was acknowledged by being named a Google Fellow, awarded an undisclosed prize that was almost certainly in the millions of dollars. There was a press release announcing that Singhal had received the award, but it did not specify the reason.

Google’s search engines would thereafter undergo major transformations every two or three years, with similar stealth. “It’s like changing the engines on a plane flying a thousand kilometers an hour, thirty thousand feet above the earth,” says Singhal. “You have to do it so the passengers don’t feel that something just happened. And in my time, we have replaced our propellers with turboprops and our turboprops with jet engines. The passengers don’t notice, but the ride is more comfortable and the people get there faster.”

In between the major rewrites, Google’s search quality teams constantly produced incremental improvements. “We’re looking at queries all the time and we find failures and say, ‘Why, why, why?’” says Singhal, who himself became involved in a perpetual quest to locate poor results that might have indicated bigger problems in the algorithm. He got into the habit of sampling the logs kept by Google on its users’ behavior and extracting random queries. When testing a new version of the search engine, his experimentation intensified. He would compile a list of tens of thousands of queries, simultaneously running them on the current version of Google search and the proposed revision. The secondary benefit of such a test was that it often detected a pattern of failure in certain queries.

As best as he could remember, that was how the vexing query of Audrey Fino came into Amit Singhal’s life.

It seemed so simple: someone had typed “Audrey Fino” into Google
and was unhappy with the result. It was easy for Singhal to see why. The results for that query were dominated by pages in Italian gushing about the charms of the Belgian-born actress Audrey Hepburn. This did not seem to be what the user was looking for. “We realized that this was a person’s name,” says Singhal. “There’s a person somewhere named Audrey Fino, and we didn’t have the smarts in the system to know this.” What’s more, he realized that it was a symptom of a larger failure that required algorithmic therapy. As good as Google was, the search engine stumbled with names.

This spurred a multiyear effort by Singhal and his team to produce a name detection system within the search engine. Names were important. Only 8 percent of Google’s queries were names—and half of those celebrities—but the more obscure names were cases where users had specific, important needs (including “vanity searches” where people Googled themselves, a ridiculously common practice). So how would you devise new signals to more skillfully identify names from queries and dig them out of the web corpus? Singhal and his colleagues began where they almost always did: with data. To improve search, Google often integrated external databases, and in this case Google licensed the White Pages, allowing it to use all the information contained in hundreds of thick newprint-based tomes where the content consisted of nothing but names (and addresses and phone numbers). Google’s search engine sucked up the names and analyzed them until it had an understanding of what a name was and how to recognize it in the system.

But the solution was trickier than that. One had to take context into effect. Consider the query “houston baker.” Was the user looking for a person who baked bread in Texas? Probably. But if you were making that query very far from the Lone Star State, it’s more likely that you were seeking someone named after the famous Texan. Google had to teach its search engine to tell the difference. And a lot of the instruction was done by the users, clicking millions of times to direct their responses to the happy zone of short clicks.

“This is all just learning,” says Singhal. “We had a computer learning algorithm on which we built our name classifier.”

Within a few months Singhal’s team built the system to make use of that information and properly parse name queries. One day not long after that, Singhal typed in the troublesome query once more. This time, rising above the pages gushing about the gamine who starred in Roman Holiday, there was a link providing information about an attorney who was, at least for a time, based in Malta: Ms. Audrey Fino.

“So now we can recognize names and do the right thing when one comes up,” says Singhal five years after the quest. “And our name-recognition system is now far better than when I invented it, and is better than anything else out there, no matter what anyone says.”

One day in 2009, he showed a visitor how well it worked, also illuminating other secrets of the search engine. He opened his laptop and typed in a query: “mike siwek lawyer mi.”

He jabbed at the key. In a time span best measured in beats of a hummingbird’s wing, ten results appeared. There were the familiar “ten blue links” of Google search. (The text consisting of the actual links to the pages cited as results was highlighted in blue.) Early in Google’s history Page and Brin had decided that ten links was the proper number to show on a page, and numerous tests over the years had reinforced the conviction that ten was the number that users preferred to see. In this case, the top result was a link to the home page of an attorney named Michael Siwek in Grand Rapids, Michigan. This success came as a result of the efforts put into motion by the Audrey Fino problem. The key to understanding a query like this, Singhal said, was the black art of “bigram breakage”: that is, how should a search engine parse a series of words entered into the query field, making the kind of distinctions that a smart human being would make?

For instance, “New York” represents two words that go together (in other words, a bigram). But do the three words in “New York Times,” which clearly indicate a different kind of search. And everything changes when the query is “New York Times Square,” in which case the breakage would come... well, you know where.

“Deconstruct this [Siwek] query from an engineer’s point of view,” says Singhal. “Most search engines I have known in my academic life will go ‘one word, two words, three words, four words, done.’ We at Google say, ‘Aha! We can break this here!’ We figure that ‘lawyer’ is not a last name and ‘Siwek’ is not a middle name,” he says. “And by the way, lawyer is not a town in Michigan. A lawyer is an attorney.”

This was the hard-won view from inside the Google search engine: a rock is a rock. It’s also a stone, and it could be a boulder. Spell it Luke, and it’s still a rock. But put “little” in front of “rock,” and it’s the capital of Arkansas. Which, is not an “ark.” Unless “Noah” is around.
All this helped to explain how Google could find someone whose name may have never appeared in a search before. (One-third of all search queries are virgin requests.) “Mike Siwek is some person with almost no Internet presence,” says Singhal. “Finding that needle in that haystack, it just happened.”

Amit Singhal turned forty in 2008. The search team celebrated with a party in his honor. As one might expect, it was a joyous celebration. Certainly there was much to celebrate besides a birthday. Consider that these were geeky mathematicians who in an earlier era would have written obscure papers and been scraping by financially on an academic’s salary. Now their work directly benefited hundreds of millions of people, and they had in some way changed the world. Plus, many of them owned stock options that had made them very wealthy.

Just before the dinner was to commence, Singhal’s boss handed a phone to him. “Someone wants to talk to you,” he said.

A female voice that Singhal did not recognize congratulated him on his milestone. “I’m sorry,” he said. “Do I know you? Did we overlap academically?”

“Oh, I’m an academic,” she said. “But we didn’t overlap.”

“Did I influence your work, or did you influence my work?”

“Well,” the woman said, “I think I influenced your work.”

Singhal was at a loss.

“I’m Audrey Fino,” she said.

Actually, she was not Audrey Fino. Singhal’s boss had hired an actress to portray the woman. The Google search engine had been able to locate the digital trail of Audrey Fino, but could not produce the actual person. That sort of magic would have to wait until later.

The secret history of Google was punctuated by similar advances, a legacy of breaking ground in computer science and keeping its corporate mouth shut. The heroes of Google search were heroes at Google but nowhere else. In every one of the four aspects of search—crawling, indexing, relevance, and speedy delivery of results—Google made advances. Search quality specialists such as Amit Singhal were like the quarterbacks and wide receivers on a football team: the eye-popping results of their ranking efforts got the lion’s share of attention. But those results relied on collecting as much information as possible. Google called this “comprehensiveness” and had a team of around three hundred engineers making sure that the indexes captured everything. “Ideally what we want to have is sort of a true mirror of the web,” says a Google engineering VP. “We want to have a copy of every document that’s out there or as many as we can possibly get, we want our copy to be as close to that original as possible both in time and in terms of representation, and then we want to organize that in such a way that it’s easy and efficient to serve, and ultimately to rank.”

Google did all it could to access those pages. If a web page required users to fill out a form to see certain content, Google had probably taught its spiders how to fill out the form. Sometimes content was locked inside programs that ran when users visited a page—applications running in the JavaScript language or a media program like Adobe’s Flash. Google knew how to look inside those programs and suck out the content for its indexes. Google even used optical character recognition to figure out if an image on the website had text on it.

The accumulation of all those improvements lengthened Google’s lead over its competitors, and the circle of early adopters who first discovered Google was eventually joined by the masses, building a dominant market share. Even Google’s toughest competitors had to admit that Brin and Page had built something special. “In the search engine business, Google blew away the early innovators, just blew them away,” says Bill Gates. “And the remains of those people will be long forgotten.”

One of PageRank’s glories (and its original advantage over AltaVista) was its resistance to spam. (The term in this sense meant not unwanted email but links in its results page that secured undeservedly high rankings by somehow tricking the system.) But as Google became the first place that millions of people looked for information on shopping, medical concerns, their friends, and themselves, the stakes were raised.

The engineer who found himself at the center of the company’s spam efforts was an inveterately social twenty-eight-year-old Kentuckian named Matt Cutts. In the summer of 1999, he was pursuing a doctorate at the University of North Carolina when he got stuck with his thesis and on a whim called Google asking what it paid engineers. He got a response saying that it didn’t reveal such information until it was actually negotiating with job candidates. Cutts went back to his thesis, but a couple of days later, he got another message: “Would you like to be in active negotiation?” Clearly, he’d been Googled. After some phone screenings, he flew out to California, getting a taste for the company’s frugality when Google put him up in one of the funky clapboard motels on El Camino Real. Visiting the Google headquarters, he was taken aback by the scene: people working at haphazardly
placed sawhorse desks and the director of engineering, Urs Hölzle, playing a high-tech game of fetch with his huge dog, making the floppy beast chase the beam of a laser pointer. In the whirl of interviews, Cutts would remember one question: “How’s your UNIX kung fu?” (UNIX being a popular operating system used in many of Google’s operations.) “My UNIX kung fu is strong,” Cutts replied, deadpan.

He got the job, though his fiancée wouldn’t move to California unless they married immediately. After a courthouse wedding and a Caribbean honeymoon, bride and groom drove across the country to Cutts’s new job in January 2000, where he sat in a cubicle outside Larry and Sergey’s office. Eventually he found himself in an office with Amit Singhal, Ben Gomes, and Krishna Bharat. It was like entering the high temple of search.

Cutts’s first job was helping to create a product called SafeSearch, which would allow people to block pornography from search results. Getting rid of unwanted porn was always a priority for Google. Its first attempt was to construct a list of five hundred or so nasty words. But in 2000, Google got a contract to provide search to a provider that wanted to offer a family-safe version of search to its customers. It needed to step up its game. Brin and Page asked Cutts how he felt about porn. He’d have to see a lot of it to produce a system to filter it out of Google.

Cutts asked his colleagues to help him locate adult websites so he could extract signals to better identify and block them, but everyone was too busy. “No one will help me look for porn!” he complained to his wife one night. She volunteered to bake chocolate chip cookies for Cutts to award to Googlers who found porn sites that slipped through Cutts’s blockade. At the time, Google was updating the index once a month, and before the new version was released, Cutts would host a Look for Porn Day, bringing in his spouse’s confections. “She’s still known as the porn cookie lady at Google,” he says.

The major porn sites were fine with the process; they knew it was bad for them when searchers unintentionally stumbled upon their warehouses of sin, making them a target for muckrakers and publicity-seeking legislators. But not all such sites were good citizens. Cutts noticed that one nasty site used some clever methods to game Google’s blocking system and score high in search results. “It was an eye-opening moment,” says Cutts. “PageRank and link analysis may be spam-resistant, but nothing is spam-proof.”

The problem went far beyond porn. Google had won its audience in part because it had been effective in eliminating search spam. But now that Google was the dominant means of finding things on the Internet, a high ranking for a given keyword could drive millions of dollars of business to a site. Sites were now spending time, energy, and technical wizardry to deconstruct Google’s processes and artificially boost page rank. The practice was called search engine optimization, or SEO. You could see their handiwork when you typed in the name of a hotel. The website of the actual hotel would not appear on the first page. Instead, the top results would be dominated by companies specializing in hotel bookings. This made Google less useful. Cutts went to Wayne Rosing and told him that the company really needed to work on stopping spam. Rosing told him to go ahead and try.

A delicate balance was required. Legitimate businesses as well as shady ones partook in the sport. Highly paid consultants tried to reverse-engineer PageRank and other Google techniques. Even amateurs could partake in the hunt for “Google juice,” buying books like Search Engine Optimization for Dummies. The conjurers of this field would gather several times a year at conferences, with hotel ballrooms packed to the gills with webmasters and consultants.

Google maintained that certain SEO methods—such as making sure that the subject matter of the page was reflected in the title and convincing webmasters of popular websites to put links to your site when relevant—were good for the web in general. This begged the question: if a website had to hire outside help to improve its rankings, wasn’t that a failure of Google, whose job it is to find the best results for its users, no matter how the information is formatted or who links to it?

“Ideally, no one would need to learn SEO at all,” Cutts says. “But the fact is that it exists and people will be trying to promote themselves, so you want to be a part of the conversation and say, ‘Here are some good ethical things to do. Here are some things that are very high risk. Stay away from them.’” Cutts would admit that because not everyone has SEO expertise, sometimes Google underranks worthy sites. One example was famous: the query “Eika Kerzen.” That was not a name but a German candle manufacturer (kerzen is the German word for “candles”), whose presence was shamelessly low in rankings for keywords that should have unearthed its excellent products. This matter was dumped on Amit Singhal, who launched an algorithmic revamp of the threshold by which Google translated part of a query into another language, a solution that resolved a whole category of such troublesome results.

A perpetual arms race was waged between Google’s search quality algorithms and companies attacking the system for gain. For several years,
Google implemented spam-fighting changes in its monthly index update. It generally aligned those updates to the lunar cycle. “Whenever the full moon was about to appear, people would start jonesing for a Google update,” says Cutts. The SEO community would nervously await changes that could potentially knock its links down the relevance chain. As soon as the new values were reflected in the scores, the SEO crowd would try to divine the logic behind the new algorithms and devise responses so the downgraded links could reclaim their previous rankings. This interaction was dubbed “the Google dance.” (Things got more complicated after the BART project switched index updates from batch-processed to incremental.)

Often the changes in ranking were slight and there were measures available to restore a link to former glory. But other times Google would identify behavior that it judged an attempt to exploit vulnerabilities in its ranking system and would adjust the system to shore up those weaknesses—relegating those using that method to the bottom of the results pile. Generally, the places that got such treatment had no business showing up in the upper reaches of results for popular keywords: they sneaked their way up by creating Potemkin villages full of “link farms” designed to pump up a PageRank. Nonetheless, companies whose sites were downgraded in that matter were often outraged. “It’s not like we’ve put all our eggs in one basket,” said the president of an SEO company called WebGuerrilla to CNET in October 2002, “it’s just that there’s no other basket.” That was the month that a company called SearchKing sued Google after a bad night at the Google dance lowered its PageRank score from 8 to 4 and its business tanked. (In May 2003, a judge dismissed the suit, on the grounds that PageRank is essentially an opinion about a website—albeit an opinion expressed by algorithms—and thus was constitutionally protected.)

Cutts understood that the obscurity of the process could sour people on the company and took it upon himself to be the company’s conduit to the SEO world. Using the pseudonym “Google Guy,” Cutts would answer questions and try as best he could to dispel various conspiracy theories, many of them centered around the suspicion that a sure way to rise in search rankings was to buy ads from Google. But there was only so much he could tell. In large part because of the threat from spammers—as well as fear that the knowledge could benefit competitors—Google treated its search algorithms with utmost confidentiality. Over the years Cutts’s spam team grew considerably (as was typical for Google, Cutts wouldn’t specify the number). “I’m proud to say that web spam is much lower than it was a few years ago,” he says.

But Google’s approach had its cost. As the company gained a dominant market share in search—more than 70 percent in the United States, higher in some other countries—critics would increasingly uncomfortable with the idea that they had to take Google’s word that it wasn’t manipulating its algorithm for business or competitive purposes. To defend itself, Google would characteristically invoke logic: any variance from the best possible results for its searches would make the product less useful and drive people away, it argued. But it withheld the data that would prove that it was playing fair. Google was ultimately betting on maintaining the public trust. If you didn’t trust Google, how could you trust the world it presented in its results?

3

“If you’ve Googled it, you’ve researched it, and otherwise you haven’t.”

To get a sense of how far Google search advanced in the first six or seven years of the company, one could look through the eyes of Udi Manber.

Manber had watched it all happen, from the outside. He was born in the town of Kiryat Haim, north of Haifa in Israel. He spent so much time in the small library there that he knew nearly every volume in the collection. Manber loved telling visitors to the library which books they might enjoy—and which ones might answer their questions. He studied information retrieval and eventually wound up at Yahoo where he brokered the Google deal, until he quit in disgust in 2002. His next job was as the leader of A9, a search start-up funded by Jeff Bezos. In February 2006, he accepted an offer from Google to become the czar of search engineering. It was like someone who worked on space science all his life finally arriving at NASA. “Suddenly I’m in charge of everybody asking questions in the whole world,” he says. “I thought I had a reasonable idea of the main problems facing search—what was minor and major. When I got here, I saw they solved many of the minor problems and made more headway on the major problems than I thought possible. Google hadn’t just said, ‘Here’s the state of the art, here’s what the textbooks say, let’s do it;’ they developed things from scratch and did it better.”
He was also amazed at how pampered employees were. Every search engineer had exclusive use of a set of servers that stored an index of the entire web—it was the digital equivalent of giving a physicist her own particle accelerator.

One of the first things that happened on Manber's watch was something called Universal Search. In its first few years, Google had developed a number of specialized forms of search, known as verticals, for various corpuses—such as video, images, shopping catalogs, and locations (maps). Krishna Bharat had created one of those verticals called Google News, a virtual wire service with a front page determined not by editors but algorithms. Another vertical product, called Google Scholar, accessed academic journals. But to access those verticals, users had to choose the vertical. Page and Brin were pushing for a system where one search would find everything.

The key engineer in this project was David Bailey, who had worked with Manber at A9. Bailey was a Berkeley computer science PhD who had once worried that by following his interests—artificial intelligence and the way computers dealt with natural language—he was locking himself in a field with few practical applications. "I figured that no one is ever going to employ someone who's got a PhD in those things because everybody knows that no computer application worth its salt would deal with plain English text." That was before Google, which he joined in 2004.

At Google, he had the luxury to figure out what he wanted to do. He found himself in an office with Amit Singhal, Matt Cutts, and Ben Gomes (who'd been his buddy in grad school)—"definitely the cool kids' office," he says—and was bowled over by the rich conversations. He needed all the expertise he could find when he was assigned the task of augmenting Google search so that the results page included not only web results but hits from pictures, books, videos, and other sources. If Google really cared about "organizing and making accessible the world's information," as it continually boasted (to the point of arrogance, it seemed), it really had to expand its ten blue links beyond web pages. But the challenges were considerable, and several attempts at executing that vision had flopped. "It had become the project of death," says Bailey.

Nonetheless, Bailey took on the task. He gathered together a team that included a bright product manager named Johanna Wright. Even though Universal Search was something that Larry Page had been urging for years, there was a lot of resistance. "There was definitely a momentum-gathering phase," says Wright, "and finally there was a point where everyone wanted to work on the project, and it all came together."

A big challenge in Universal Search was how to determine the relative value of information when it came from different places. Google had gotten pretty good at figuring out how to rank websites for a given query, and it had also learned a lot about ordering the corpus of pictures or video results to satisfy search requests. Every corpus had a different mix of signals. (Everything on the web, of course, had the benefit of linking information, but things such as videos did not have an equivalent.)

For Universal Search, though, Google had to figure out the relative weight to assign to different sets of signals. It became known as the apples- and- oranges problem. The answer, as with many things in Google, lay in determining context from the data in its logs—specifically in analyzing the long clicks in the past. "We have a lot of signals that tell us the intent of the queries," says Wright. "There could be information in the query that tells us a news result is really relevant and extremely important, and then we'd put it on top of the page." But clearly the solution involved decoding the intent of a query. In some cases, it turned out that Google's signals in a given area weren't effective enough. "It became an opportunity for us to revisit the rankings on those," says Bailey. Eventually, they got to the point where Google, he says, "transformed the ranking problem to be apples to apples."

A knottier problem turned out to be how to show these results on the page. Although Google could figure out that certain results—a video clip, a book, a picture, or a scholarly article—might be relevant to a request, the fact was that users mainly expected web links to dominate the results page.

When the Universal Search team showed a prototype to Google's top executives, everyone realized that taking on the project of death had been worth it. The results in that early attempt were all in the wrong order, but the reaction was visceral—you typed in a word, and all this stuff came out. It had just never happened before. "It definitely was one of the riskier things," says Bailey. "It was hard, because it's not just science—there are some judgment calls involved here. We are to some degree using our gut. I still get up in the morning and am astonished that this whole thing even works."

Google's search now wasn't just searching the web. It was searching everything.

In his 1991 book, Mirror Worlds, Yale computer scientist David Gelernter sketched out a future where humans would interact, and transact, with modeled digital representations of the real world. Gelernter described these doppelgänger realities as "a true-to-life mirror image trapped inside a computer." He made it a point to distinguish his vision from the trendy
sci-fi sensation of the moment, virtual reality—fantasy simulations inside the computer as opposed to a digital companion of the physical world. “The whole point of a mirror world is that it’s wired in real time and place—it’s supposed to mirror reality rather than being a parallel reality or cyberworld,” he once said. But though Gelernter looked on the overall prospect of mirror worlds with enthusiasm, he worried as well. “I definitely feel ambivalent about mirror worlds. There are obvious risks of surveillance, but I think it poses deeper risks,” he said. His main concern was that mirror worlds would be steered by the geeky corporations who built them, as opposed to the public. “These risks should be confronted by society at large, not by techno-nerds,” he said. “I don’t trust them. They are not broad-minded and don’t know enough. They don’t know enough history, they don’t have enough of a feel for the nature of society. I think that’s a recipe for disaster.”

But like it or not, Google, the ultimate techno-nerd corporation, was building a mirror world. For many practical purposes, information not stored in the vast Google indexes, which contained, among other things, all the pages of the publicly available web, may as well not have existed. “I’d like to get to a state that people think of it as ‘If you’ve Googled it, you’ve researched it, and otherwise you haven’t, and that’s it,’” says Sergey Brin.

While working on its big revisions like Universal Search, Google kept trying to improve its search in general. Dozens of engineers plugged away at failed queries, trying to determine if, as with the case of Audrey Fino, they pointed to deeper algorithmic shortcomings.

The wrong way to fix things was to patch the algorithm to address a specific failed query. That was an approach that didn’t scale; it clashed with the idea that Google’s giant search algorithm could find the most relevant material by its own logic alone. A legendary story at Google illustrated this principle. Around 2002, a team was testing a subset of search limited to products, called Froogle. But one problem was so glaring that the team wasn’t comfortable releasing Froogle: when the query “running shoes” was typed in, the top result was a garden gnome sculpture that happened to be wearing sneakers. Every day engineers would try to tweak the algorithm so that it would be able to distinguish between lawn art and footwear, but the gnome kept its top position. One day, seemingly miraculously, the gnome disappeared from the results. At a meeting, no one on the team claimed credit. Then an engineer arrived late, holding an elf with running shoes. He had bought the one-of-a kind product from the vendor, and since it was no longer for sale, it was no longer in the index. “The algorithm was now returning the right results,” says a Google engineer. “We didn’t cheat, we didn’t change anything, and we launched.”

Over the years, Google evolved a set process for search engine tweaks. After an engineer identified a flaw, he or she would be assigned a “search analyst” to manage the next several weeks, during which the improvement would be implemented. The engineer would determine the problem and recode the relevant part of the search algorithm. Maybe it would require adjusting the importance of a signal. Or perhaps altering the interpretation of multiword “bigrams.” Or even integrating a new signal. Then the counselor would submit it to testing.

Part of that testing involves hundreds of people around the world who sit at their home computers and judge results for various queries, marking whether the new tweaks return better or worse results than the previous versions. “We cover over a hundred locales,” says engineering director Scott Huffman, who is in charge of the testing process. “We have Swiss-French evaluators and Swiss-German evaluators and so on.” But Google also employs a much bigger army of testers—it’s millions of users, virtually all of whom are unwitting lab rats for Google’s constant quality experiments.

The mainstay of this system was the “A/B test,” where a fraction of users—typically 1 percent—would be exposed to the suggested change. The results and the subsequent behavior of those users would be compared with those of the general population. Google gauged every alteration to its products that way, from the hue of its interface colors to the number of search results delivered on a page. There were so many changes to measure that Google discarded the traditional scientific nostrum that only one experiment should be conducted at a time, with all variables except the one tested being exactly the same in the control group and the experimental group. “We want to run so many experiments, we can’t afford to put you in any one group, or we’d run out of people,” says a search quality manager. “On most Google queries, you’re actually in multiple control or experimental groups simultaneously. Essentially all the queries are involved in some test.”

In search tweaks, the culmination of the process would come in the weekly Search Quality Launch Meeting. In a typical session in 2009, fifty engineers, mostly in their twenties and early thirties, participated. One test query was “Terry Smith KS,” a search that appeared on a screen and had been launched from Springfield, Missouri. The baseline, or unaltered
result, assumed that the user wants a link to a town called Smith, in Kansas. A tweaked version of the search included a link to a Tarry Smith who lives in Kansas. That was considered a win by the engineers. On the other hand, when a tester in Sykesville, Maryland, tried the query “weather.com Philadelphia,” the new version gave a high ranking to a map showing the location of the long-defunct main office of Bell Telephone of Pennsylvania. That was strange and a big loss. This result spurred a vigorous discussion. Someone figured it out: probably, in some earlier period of technology when Bell Telephone was a sort of search engine, that office was the source of the dial-up phone service that told you the weather. Buried on the web somewhere was that factoid, and the alteration to the algorithm had somehow routed it out of its obscurity. In 2009, Google search engineers made more than six hundred changes to improve search quality.

It was no coincidence that the man who eventually headed Google’s research division was the coauthor of *Artificial Intelligence: A Modern Approach*, the standard textbook in the field. Peter Norvig had been in charge of the Computational Science Division at NASA’s facility in Ames, not far from Google. At the end of 2000, it was clear to Norvig that turmoil in the agency had put his programs in jeopardy, so he figured it was a good time to move. He had seen Larry Page speak some months before and sensed that Google’s obsession with data might present an opportunity for him. He sent an email to Page and got a quick reply—Norvig’s AI book had been assigned reading for one of Page’s courses. After arriving at Google, Norvig hired about a half-dozen people fairly quickly and put them to work on projects. He felt it would be ludicous to have a separate division at Google that specialized in things like machine learning—instead, artificial intelligence should be spread everywhere in the company.

One of the things high on Google’s to-do list was translation, rendering the billions of words appearing online into the native language of any user in the world. By 2001, Google.com was already available in twenty-six languages. Page and Brin believed that artificial barriers such as language should not stand in the way of people’s access to information. Their thoughts were along the lines of the pioneer of machine translation, Warren Weaver, who said, “When I look at an article in Russian, I say, ‘This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.’” Google, in their minds, would decode every language on the planet.

There had been previous attempts at online translation, notably a service dubbed Babel Fish that first appeared in 1995. Google’s own project, begun in 2001, had at its core a translation system licensed from another company—basically the same system that Yahoo and other competitors used. But the system was often so inaccurate that it seemed as though the translated words had been selected by throwing darts at a dictionary. Sergey Brin highlighted the problems at a 2004 meeting when he provided Google’s translation of a South Korean email from an enthusiastic fan of the company’s search technology. It read, “The sliced raw fish shoes it wishes. Google green onion thing.”

By the time Brin expressed his frustration with the email, Google had already identified a hiring target who would lead the company’s translations efforts—in a manner that solidified the artificial intelligence focus that Norvig saw early on at Google. Franz Och had focused on machine translations while earning his doctorate in computer science from the RWTH Aachen University in his native Germany and was continuing his work at the University of Southern California. After he gave a talk at Google in 2003, the company made him an offer. Och’s biggest worry was that Google was primarily a search company and its interest in machine translation was merely a flirtation. A conversation with Larry Page dissolved those worries. Google, Page told him, was committed to organizing all the information in the world, and translation was a necessary component. Och wasn’t sure how far you could push the system—could you really build for twenty language pairs? (In other words, if your system had twenty languages, could it translate any of those to any other?) That would be unprecedented. Page assured him that Google intended to invest heavily. “I said okay,” says Och, who joined Google in April 2004. “Now we have 506 language pairs, so it turned out it was worthwhile.”

Earlier efforts at machine translation usually began with human experts who knew both languages that would be involved in the transformation. They would incorporate the rules and structure of each language so they could break down the original input and know how to recast it in the second tongue. “That’s very time-consuming and very hard, because natural language is so complex and diverse and there are so many nuances to it,” says Och. But in the late 1980s some IBM computer scientists devised a new approach, called statistical machine translation, which Och embraced. “The basic idea is to learn from data,” he explains. “Provide the computer with large amounts of monolingual text, and the computer should figure out himself what those structures are.” The idea is to feed the computer massive amounts of data and let him (to adopt Och’s anthropomorphic
pronoun) do the thinking. Essentially Google's system created a "language model" for each tongue Och's team examined. The next step was to work with texts in different languages that had already been translated and let the machines figure out the implicit algorithms that dictate how one language converts to another. "There are specific algorithms that learn how words and sentences correspond, that detect nuances in text and produce translation. The key thing is that the more data you have, the better the quality of the system," says Och.

The most important data were pairs of documents that were skillfully translated from one language to another. Before the Internet, the main source material for these translations had been corpuses such as UN documents that had been translated into multiple languages. But the web had produced an unbelievable treasure trove—and Google's indexes made it easy for its engineers to mine billions of documents, unearthing even the most obscure efforts at translating one document or blog post from one language to another. Even an amateurish translation could provide some degree of knowledge, but Google's algorithms could figure out which translations were the best by using the same principles that Google used to identify important websites. "At Google," says Och, with dry understatement, "we have large amounts of data and the corresponding computational resources we need to build very, very, very good systems."

Och began with a small team that used the latter part of 2004 and early 2005 to build its systems and craft the algorithms. For the next few years, in fact, Google launched a minicrusade to sweep up the best minds in machine learning, essentially bolstering what was becoming an AI stronghold in the company. Och's official role was as a scientist in Google's research group, but it is indicative of Google's view of research that no step was required to move beyond study into actual product implementation.

Because Och and his colleagues knew they would have access to an unprecedented amount of data, they worked from the ground up to create a new translation system. "One of the things we did was to build very, very, very large language models, much larger than anyone has ever built in the history of mankind." Then they began to train the system. To measure progress, they used a statistical model that, given a series of words, would predict the word that came next. Each time they doubled the amount of training data, they got a 5 percent boost in the metrics that measured success in the results. "So we just doubled it a bunch of times." In order to get a reasonable translation, Och would say, you might feed something like a billion words to the model. But Google didn't stop at a billion.

By mid-2005, Google's team was ready to participate in the annual machine translation contest sponsored by the National Institute of Standards and Technology. At the beginning of the event, each competing team was given a series of texts and then had a couple of days for its computers to do the translation, while government computers ran evaluations and scored the results. For some reason, NIST didn't characterize the contest as one in which a participant is crowned champion, so Och was careful not to declare Google the winner. Instead, he says, "Our scores were better than the scores of everyone else." One of the language pairs it was tested on involved Arabic. "We didn't have an Arabic speaker on the team but did the very best machine translation."

By not requiring native speakers, Google was free to provide translations to the most obscure language pairs. "You can always translate French to English or English to Spanish, but where else can you translate Hindi to Danish or Finnish or Norwegian?"

A long-term problem in computer science had been speech recognition—the ability of computers to hear and understand natural language. Google applied Och's techniques to teaching its vast clusters of computers how to make sense of the things humans said. It set up a telephone number, 1-800-GOOG-411, and offered a free version of what the phone companies used to call directory assistance. You would say the name and city of the business you wanted to call, and Google would give the result and ask if you wanted to be connected. But it was not a one-way exchange. In return for giving you the number, Google learned how people spoke, and since it could tell if its guess was successful, it had feedback that told it where it went wrong. Just as with its search engine, Google was letting its users teach it about the world.

"What convinced me to join Google was its ability to process large-scale information, particularly the feedback we get from users," says Alfred Spector, who joined in 2008 to head Google's research division. "That kind of machine learning has just not happened like it's happened at Google."

Over the years Google has evolved what it calls "a practical large scale machine learning system" that it has dubbed "Seti." The name comes from the Search for Extra Terrestrial Intelligence, which scans the universe for evidence of life outside Earth; Google's system also works on the scale of the universe as it searches for signals in its mirror world. Google's indexes almost absurdly dwarf the biggest data sets formerly used in machine learning experiments. The most ambitious machine learning effort in the UCI KDD Archive of Large Data Sets for Data Mining Research and Experi-
mentation is a set of 4 million instances used to detect fraud and intrusion detection. Google's Seti learning system uses data sets with a mean training set size of 100 billion instances.

Google’s researchers would acknowledge that working with a learning system of this size put them into uncharted territory. The steady improvement of its learning system flirted with the consequences postulated by scientist and philosopher Raymond Kurzweil, who speculated about an impending “singularity” that would come when a massive computer system evolves its way to intelligence. Larry Page was an enthusiastic follower of Kurzweil and a key supporter of Kurzweil-inspired Singularity University, an educational enterprise that anticipates a day when humans will pass the consciousness baton to our inorganic progeny.

What does it mean to say that Google “knows” something? Does Google's Seti system tell us that in the search for nonhuman intelligence we should not look to the skies but to the million-plus servers in Google's data centers?

"That's a very deep question," says Spector. "Humans, really, are big bags of mostly water walking around with a lot of tubes and some neurons and all. But we're knowledgeable. So now look at the Google cluster computing system. It's a set of many heuristics, so it knows 'vehicle' is a synonym for 'automobile,' and it knows that in French it's 'voiture,' and it knows it in German and every language. It knows these things. And it knows many more things that it's learned from what people type." He cited other things that Google knows: for example, Google had just introduced a new heuristic where it determined from your searches whether you might be contemplating suicide, in which case it would provide you with information on sources of aid. In this case, Google's engine gleaned predictive clues from its observations of human behavior. They are formulated in Google's virtual brain just as neurons are formed in our own wetware. Spector promised that Google would learn much, much more in coming years.

"Do these things rise to the level of knowledge?" he asks rhetorically. "My ten-year-olds believe it. They think Google knows a lot. If you asked anyone in their grade school class, I think the kids would say yes."

What did Spector, a scientist, think?

"I'm afraid that it's not a question that is amenable to a scientific answer," he says. "I do think, however, loosely speaking, Google is knowledgeable. The question is, will we build a general-purpose intelligence which just sits there, looks around, then develops all those skills unto itself, no matter what they are, whether it's medical diagnosis or ..." Spector pauses. "That's a long way off," he says. "That will probably not be done within my career at Google." (Spector was fifty-five at the time of the conversation in early 2010.)

"I think Larry would very much like to see that happen," he adds.

In fact, Page had been thinking about such things for some time. Back in 2004, I asked Page and Brin what they saw as the future of Google search. "It will be included in people's brains," said Page. "When you think about something and don't really know much about it, you will automatically get information."

"That's true," said Brin. "Ultimately I view Google as a way to augment your brain with the knowledge of the world. Right now you go into your computer and type a phrase, but you can imagine that it could be easier in the future, that you can have just devices you talk into, or you can have computers that pay attention to what's going on around them and suggest useful information."

"Somebody introduces themselves to you, and your watch goes to your web page," said Page. "Or if you met this person two years ago, this is what they said to you." Later in the conversation Page said, "Eventually you'll have the implant, where if you think about a fact, it will just tell you the answer."

It was a fantastic vision, straight out of science fiction. But Page was making remarkable progress—except for the implant. When asked in early 2010 what will come next for search, he said that Google will know about your preferences and find you things that you don't know about but want to know about. So even if you don't know what you're looking for, Google will tell you.

What Page didn't mention was how far along Google was on that path. Ben Gomes, one of the original search rock stars, showed a visitor something he was working on called "Search-as-You-Type." Other internal names for it were "psychic" and "Miss Cleo," in tribute to a television fortune-teller. As the more prosaic name implied, this feature enables search to start delivering results even before you finish typing the query. He started typing "finger shoes"—the term that people often use to describe the kind of footwear Sergey Brin often sports, rubberized slippers with individual sleeves that fit toes the way gloves fit your fingers. Of course, Google search, with all the synonyms and knowledge fed to it by billions of searchers who clicked long and those who clicked short, knew what he
was talking about. Gomes hadn’t finished typing the second word before the page filled with links—and ads!—confidently assuming that he wanted information, and maybe a buying opportunity, involving “Vibram Five Fingers, the barefoot alternative.” “It’s a weird connection between your brain and the results,” Gomes said. (In September 2010, Google introduced this product as “Google Instant.”)

“Search is going to get more and more magical,” says search engineer Johanna Wright. “We’re going to get so much better at it that we’ll do things that people can’t even imagine.” She mentioned one example of a demo being passed around. “Say you type in ‘hamburger.’ Right now, Google will show you hamburger recipes. But we’re going to show you menus and reviews of where you can get a hamburger near you, which is great for anyone living in a place where there are restaurants. I call this project Blueberry Pancakes because if I want to check those out, it’ll tell me about the pancake house in Los Altos, and I’ll go there. It’s just another example of where we’re going—Google’s just going to really understand you better and solve many, many, many more of your needs.”

That would put Google in the driver’s seat on many decisions, large and small, that people make in the course of a day and their lives. Remember, more than 70 percent of searches in the United States are Google searches, and in some countries the percentage is higher. That represents a lot of power for the company founded by two graduate students just over a decade ago. “In some sense we’re responsible for people finding what they need,” says Udi Manber. “Whenever they don’t find it, it’s our fault. It’s a huge responsibility. It’s like we’re doctors who are responsible for life.”

Maybe, it was suggested to Manber, however well intentioned Google’s brainiacs were, it was not necessarily a good thing for any single entity to have the answer, whether it was hardwired to your brain or not.

“It may surprise you,” says Udi Manber, “but I completely agree with that. And it scares the hell out of me.”

PART TWO

GOOGLENOMICS
Cracking the Code on Internet Profits

“What’s a business plan?”

Google CEO Eric Schmidt called it “the hiding strategy.” It was Google’s biggest secret, maybe even better protected than the secrets behind search. Those who knew the secret—virtually everyone working at Google—were instructed quite firmly to keep their mouths shut about it. Outsiders who suspected the secret were given no winks of confirmation. What made this information easier to keep is that almost none of the experts tracking the business of the Internet believed that Google’s secret was even possible.

What Google was hiding was how it had cracked the code to making money on the Internet. Google had invented one of the most successful products in corporate history, and the company was swimming in black ink.

David Krane, who joined Google in 2000 as one of its first press reps, was charged with maintaining the hide and thwarting the seek. Every company he’d ever worked for previously had been more than eager to emphasize the positive when it came to financial results. But at Google, his job was misdirecting journalists away from good news. “We’d cracked one of the unsolved puzzles on the Internet—making money at scale in a way that