The Structure of Digital Computing

From Mainframes to Big Data

by

Robert L. Grossman

Open Data Press LLC
2012

Published by Open Data Press LLC
400 Lathrop Ave, Suite 90
River Forest, IL 60305, USA

First Printing: June, 2012

Library of Congress Control Number: 2012908445

ISBN 978-1-936298-00-6

Contents

1 The Five Eras of Computing **1**
1.1 Introduction 1
1.2 The Main Themes 2
1.3 A Billion IP Addresses 5
1.4 The SAD History of Computing 9
1.5 Why Symbols Matter 11
1.6 Algorithms as Recipes 15
1.7 Computing Devices 17
1.8 Case Study: Slide Rule 19
1.9 From Mainframes to Devices 23
1.10 Mainframe Era 26
1.11 Case Study: Punch Cards 27
1.12 Personal Computer Era 28
1.13 The Web Era 30
1.14 Case Study: SMTP 31
1.15 Clouds of Devices 34
1.16 Case Study: Routers 36
1.17 The First Half Century of Computing . . . 38
1.18 The Commoditization of Data 43

2 Commoditization **45**
2.1 Christmas and Easter 45
2.2 The Commoditization of Time 47
2.3 The Commoditization of Space 49
2.4 Moore's Law 52
2.5 Commoditization is All Around Us 54
2.6 The Doubling Game 58

2.7 Transforming Technologies 61
2.8 Storage and Johnson's Law 62
2.9 Bandwidth and Gilder's Law 66
2.10 Software and Stallman's Law 67
2.11 Data and the Bermuda Principles 74
2.12 Network Effects 75

3 Technical Innovation vs. Market Clutter 81
3.1 Innovation vs. Clutter 81
3.2 Approximating Solutions to Equations . . . 83
3.3 Case Study: Business Intelligence 87
3.4 Views of Technical Innovation 88
3.5 The Imperative to be in the Upper Right . 91
3.6 Why Clutter Is Inevitable 93
3.7 Who Clutters 96
3.8 Sources of Clutter: Features 98
3.9 Case Study: Databases 101
3.10 Case Study: Searching for Primes 109
3.11 Case Study: Routing Packets 111

4 Technology Adoption Cycles 119
4.1 Forces Effecting Technology Adoption . . . 119
4.2 The Basic Equation of Marketing 122
4.3 Getting to Main Street 127
4.4 Case Study: The Nike Pegasus 130
4.5 Technology Roadmaps 133
4.6 Case Study: Clusters 137
4.7 Context 140
4.8 Case Study: Relational Databases 145
4.9 Technology Pain Points 149
4.10 Case Study: Adoption of Linux 156

5 The Era of Data 161
5.1 Introduction 161
5.2 Thinking about Big Data 162
5.3 The Commoditization of Data 164
5.4 The Data Gap 168
5.5 Extracting Knowledge from Data 172

5.6 Kepler's Law and Brahe's Data 178
5.7 Pearson's Law 182
5.8 The Bermuda Principles 188
5.9 World Population 190
5.10 The Shape of Data 193
5.11 Case Study: Consumer Databases 199
5.12 Creating Digital Data 203
5.13 Using Data to Make Decisions 207
5.14 Case Study: Mammograms 211
5.15 Events, Profiles and Alerts 214
5.16 Case Study: NASA's EOS 219

Notes **225**

References **249**

Preface

This book is about the structure of digital computing: what is significant, what is novel, what endures, and why it is all so confusing. The book tries to balance two point of views: digital computing as viewed from a business perspective, where the focus is on marketing and selling, and digital computing from a more technical perspective, where the focus is on developing new technology.

My goal was to write a short book about digital computing that takes a long term point of view and integrates to some extent these two perspectives.

The book is shaped by my personal experience in these two worlds: From 1996–2001, I was the Founder and the CEO of a company called Magnify, Inc. that developed and marketed software for managing and analyzing big data. Prior to this, from 1988–1996, I was faculty member at the University of Illinois at Chicago (UIC), where I did research on data intensive and distributed computing. From 1996–2010, I remained at UIC as a part time faculty member.

I wrote the sections in this book over an approximately eight year period from 2001 to 2008, with most of the writing done during 2001–2003. I have left the older sections by and large as they were originally written.

Although there have been some changes since 2003 (for example, computers are faster, there are more web sites, and phones are smarter), hopefully as the book will make clear, at a more fundamental level, we are still on the same fifty or so year trajectory today that we were on in 2003.

Robert L. Grossman

Chapter 1

The Five Eras of Computing

1.1 Introduction

This book is about the structure of digital computing: it is concerned with what is significant, what is novel, what endures, and why it is all so confusing.

Computing and communication technologies have gotten a bad name for being hard to predict and difficult to understand. In this book, I take the opposing point of view: that many of the most important phenomena that underlie computing have been remarkably regular and predictable over the past fifty years.

For example, the remarkable growth of processing power exemplified by Moore's Law has followed a regular pattern for over forty years. To put it simply, for most applications, processing power is a commodity and no harder to get than other commodities, such as electrical power. What is sometimes not appreciated is that a variety of other underlying processes that form the basis for today's computational and communications infrastructure have also been commoditized. For example, software and network bandwidth have been commoditized and show a similar regularity.

On the other hand, it is easy to lose sight of this regularity and predictability given the market clutter created by the many players with financial interests in computing and related fields. One of themes of this book is that technical innovation is generally masked by market clutter.

Understanding technology is often confused with the challenge of predicting which of the several thousand technology vendors will be around in five years and what their sales and profitability will be. This is a much harder problem and not one of the subjects of this book. It may be helpful to think of the survival of a vendor over five years as being modeled by a random walk, in much the same way that the stock market is often modeled by a random walk.

True computing innovations have a beauty and a longevity that creates regularity and simplicity in the historical narrative of computing. Although technical innovations are rare and cannot be predicted, they are usually recognized relatively quickly.

In this book, we try to focus on some of the underlying ideas and principles which have been fundamental drivers for computing and communications technology. These tend to be simple, rather than complex; long-lived, rather than short-lived; and easy to understand, but not easy to anticipate. These drivers have broad applicability rather than narrow applicability. Although they may have been introduced by individuals, they are brought to market by a variety of vendors using a variety of business models over a number of business cycles.

1.2 The Main Themes

> Everything should be made as simple as possible, but no simpler.
>
> Albert Einstein

This book has five main themes:

Theme 1. Our computing environment is shaped by commoditization, which governs the progress from one computing era to the next. The most familiar example of commoditization is Moore's law describing the rapid increase in power of integrated circuits at the same time that the unit cost has been stable or decreased. More generally, commoditization is the phenomenon in which unit capacity of a core technology grows exponentially, while unit cost is stable or decreases.

In this book we also examine the commoditization of several other critical technologies including storage, networking, and software. One of the themes in this book is that the commoditization of a handful of critical technologies has shaped our computing and communications infrastructure. We call these transforming technologies. Most other computing related technologies merely fill in the details.

The process of adopting and using transforming technologies is regular, lasts for decades, and is relatively easy to forecast. It is also not new. The printing press commoditized books and the telephone commoditized global person to person communication.

Chapter 1 provides an introduction to five different eras of computing, each one shaped by the commoditization of a different component of technology. Chapter 2 discusses commoditization in more detail.

Theme 2. Technical innovation is rare. It is also difficult to predict. Although technical innovation is important and critical, given its rarity, there is ample time to detect it and to understand it when it does occur.

You can think of innovation as being at the bottom of an inverted pyramid. Look it this way: A technical innovation requires at least 10 engineering advances. A hundred companies will try to bring products to market commercializing these advances. These companies over the lifetime of the products will produce 1000 marketing campaigns. Analysts and pundits will write 10,000 articles analyzing these campaigns and products. It is more efficient to understand the single innovation than to deconstruct the 10,000 articles.

Theme 3. Market clutter is rampant. The large number of players with financial stakes in technology produces a large amount of market clutter. It is hard to see through the market clutter. Pundits and industry analysts are part of the system and instead of helping, they make things worse. There is not much to do about this except to ignore it.

As already mentioned, in this book we tend to view the survival of a technology company over a five year period as a random walk: looking at five year intervals, some technology vendors will grow, some will shrink, and some will be absorbed by other vendors; but to first order, which vendor does which can best be modeled as random walk.

Chapter 3 is about the rarity of technical innovation and the confusion caused by market clutter.

Theme 4. Technology takes time. The process by which technology is created and adopted takes years. The process begins in the laboratory and ends with sales and marketing. This process is relatively well understood and relatively regular. On the other hand, the particular vendors that bring a new technology to market are not so easy to predict. Which vendors survive and which fail is perhaps best viewed as a random walk, as we have mentioned. The adoption of new technology is complicated by several different cycles involved: technology adoption cycles usually last a decade or more, while the life cycle of many technology companies is 3-7 years, and marketing cycles and fashions last 1 to 2 years. You can think of this as a tricycle with three different size wheels. Steering is obviously quite hard.

Chapter 4 is about technology adoption cycles.

Theme 5. We are entering the era of big data. We are currently entering an era defined by the commoditization of data. More precisely, over the next decade or so, we will be entering a new era of discovery driven by vast amounts of new data being produced and archived. From a broad historical perspective, having a surplus of unanalyzed data is unusual. For example, Darwin spent 17 years collecting data before publishing one of his papers. On the other hand, today an individual with a laptop and a web connection can try to create new drugs by accessing human genetic sequences, three dimensional protein data, and the chemical properties of various compounds. All this is available from the web today at no cost.

Chapter 5 is about this emerging era of data.

How the book is organized. The chapters are meant to be read in order. As just mentioned, Chapter 1 provides a framework for understanding the most important broad trends in computing. Chapter 2 is about commoditization. Chapter 3 is about the rarity of innovation and the prevalence of market clutter. It contains lots of examples, the purpose of which is to give the reader some practice so that it is easier to separate technical innovation from market clutter. Chapter 4 describes the pattern with which new technology is typically adopted by the market and the many years this usually requires. Chapter 5 is about the Era of Big Data.

Each chapter contains several extended examples and case studies, some of which are quite detailed. Feel free to skim or to ignore the ones that you don't find interesting. These examples and case studies are included since many people find it easiest to learn through concrete examples.

1.3 Case Study: A Billion IP Addresses

When you use your home computer to buy an airline ticket on the web site www.united.com, your computer looks up the web site www.united.com to get the number 209.87.-112.90. This is called an *IP address* and in some ways is somewhat similar to a telephone number. The current format for IP addresses was defined in 1981 and provided roughly 4 billion of them. Since the 1981 world population

```
9  44 40 40  209.247.34.166  internap-ne.chicago1.level3.net
10 38 37 37  64.94.32.11     border6.po1-bbnet1.chg.pnap.net
11 49 40 41  64.94.34.74     mypoints10.border6.chg.pnap.net
12 45 40 40  209.87.127.111  -
13 42 40 43  209.87.112.90   www.united.com
```

Figure 1.1: The Linux command traceroute provides the IP addresses of intermediate points between your computer and hosts on the Internet, such as www.united.com. This is fragment of a traceroute to www.united.com, showing the last portion of the route to www.united.com. The fifth column is the IP address of the intermediate points along the way to www.united.com.

was roughly 4.5 billion, since only a handful of people had access to computers, and since only some of these had network access, this seem a reasonable number of addresses.

To connect to the Internet, a company such as United Airlines needs an IP address such as 209.87.112.90. Once it has an IP address, it can provide a variety of services, such as serving web pages describing flights between Chicago and Hawaii and offering airline tickets for sale.

Beginning in 1999, a new type of Internet address became available, called IPv6. An example of an IPv6 address is

`1080:0:0:0:8:800:200C:417A.`

IPv6 addresses are longer than IPv4 addresses. IPv4 addresses are 32 bits long, while IPv6 addresses are 128 bits long.

Today, not only can computers connect to the Internet, but so can mobile phones. This means that it is useful for a mobile phone to have an IP number. For over a decade it has been clear that there were not enough IPv4 addresses for each device, such as a mobile telephone, to have its own IPv4 number. The IPv6 addresses were introduced in part so that each device could have its own IP number and easily connect to the Internet.

A manufacturer of mobile telephones, such as Nokia or Ericsson, is assigned large blocks of IPv6 addresses to burn in to the telephones they build. Blocks are assigned in units called /48's. For example, Nokia would request a /48 from European Registry for the delegation of Internet Numbers or ERIN. The interesting thing is that a /48 has enough IP numbers to set up 2^{16} separate networks, with each network having as many as 2^{64} separate computers or other IP devices [62].

Since the world population is about 6 billion (or about 2^{30}, this may seem somewhat excessive. On the other hand, there were about 1.18 billion new telephones sold in 2008, about 39% of the them by Nokia [109]. Today, we are in the midst of a transition from a computing infrastructure in which computers are connected to form networks that are in turn aggregated to form the Internet, to an infrastructure in which mobile devices supplying services are connected to form networks that are in turn connected to create clouds of services. A Nokia phone may require several IP addresses, each for a separate service in the cloud, such as talking, browsing the web, GPS location, etc. From this perspective, a billion addresses doesn't go as far as once did.

The Internet interconnects millions of different networks and billions of different computers and devices. Each computer which is directly on the Internet has a unique IP address. The simplest of are obtained by concatenating a network ID and a host ID. The network ID identifies which network it is on, while the host ID distinguishes different computers on the same network.

Actually, it is a bit more complicated. Just as some large offices use private branch exchanges (PBXs) so that individual phones have extension numbers and not unique telephone numbers, many large companies use a similar scheme so that the company itself has an IP address and individual companies have what are essentially extension numbers, which are used within the company's internal network.

Here is another way to think about the transition from

IPV4 to IPV6. IPV4 provides about 4 billion different addresses, which was originally more than enough to create a world wide network of computers, but which today is running out of space. IPV6 provides about 340 trillion, trillion, trillion (3.4×10^{38}) addresses, which is enough today to create a world wide network of devices, but which in twenty years may not be enough. With IPV6 there are enough addresses so that your phones, games, cars, and cameras can each have several different addresses.

The transition from a world wide network of computers to a world wide network of devices is natural and gradual and part of broader transition that is over fifty years old. Indeed, over the past fifty years, computer hardware, computer software, and networking have progressed in a relatively predictable fashion over a trajectory which will soon create a world in which your cell phone and car are just as much part of a network as your Gmail account.

If this seems surprising, it is simply because news about computing is reported as consisting of a series of breakthroughs. In fact, computing can more fruitfully be thought of as a trajectory in which the slope changes from time to time as the result of innovation. Innovation is hard to predict but is quickly recognized and does not change the fundamental pattern. This doesn't mean everything is easy to predict in computing. For example, predicting which company will supply the component technology for your cell phone and car is not so easy. Predicting vendors in this fashion perhaps may be best considered as the result of a random walk. News about which vendors are ahead and which are behind is part of the market clutter which makes learning about technology confusing. On the other hand, unless you have invested in one of these companies, you don't care all that much as long as one company survives.

We are currently in an era of computing dominated by web browsers surfing the Internet. Over the next decade we will transition to an era of computing in which large numbers of devices will all connect to networks, many of which will be wireless, and all of which will be able to communicate with each other. Over the past decade we have emerged from an era of computing characterized by PCs using office applications, such as word processors and spread sheets. Prior to that, computing was dominated by terminals connected to mainframes. These four eras span over half a century. From this perspective, change has been relatively easy to predict. This is the perspective of this book.

In this chapter, we describe these four major eras of computing that span the past fifty years in more detail. We will also describe an emerging fifth era of computing in which data and information will become just as commoditized as have computer cycles, storage, bandwidth and software in the proceeding eras.

1.4 The SAD History of Computing

> The river where you set your foot just now is gone —
> those waters giving way to this, now this.
>
> Heraclitus

> What has been, that will be; what has been done, that will be done. Nothing is new under the sun. Even the thing of which we say, "See, this is new!" has already existed in the ages that preceded us.
>
> Ecclesiastes

In this book, we take a broad perspective on computing and communication, trying to understand digital computing and communication from the perspective of decades, rather than the perspective of a media cycle, in which there is pressure for each new print edition or each new broadcast to announce something innovative. To do this, in Section 1.9 we will divide computing and communication technologies into five eras: the mainframe era, the personal computer era, the web era, the device era, and the data era.

Before we begin though, it is instructive to take an even longer perspective and to divide computing and communicating technologies into several epochs, the last of which is the digital epoch. (Think of epochs as much longer than eras.)

From this broader perspective, progress in computing and communications can be viewed along three dimensions: the *symbols* used, the *algorithms* which govern how we manipulate the symbols in order to compute something, and the *devices* used to manipulate the symbols. It might be helpful to remember the acronym SAD for **S**ymbols, **A**lgorithms and **D**evices.

A good example of this perspective is provided by the slide rule. The symbols used by slide rules are the same symbols that children learn in elementary school today. What is important to remember is that the positional number system in which 1402 = 1 x 1000 + 4 x 100 + 0 x 10 + 2 was not available to the Greeks, and was only fully developed in India during the 9th century. Because of this, computation was much more difficult for the Greeks.

Algorithms for multiplying and dividing numbers using slide rules rely on two formulas involving logarithms that developed in the 17th century and turn multiplication and division into addition and subtraction.

A simple slide rule consists of two ruled pieces of wood, one of which can be moved relative to the other. This is the device and it is a simple but effective analog computer. The slide rule as a computing device is so elegant and effective that it dominated computation for over 300 years.

As another example, today's digital computers use symbols representing binary numbers, employ algorithms that add, multiply, and perform other operations on these symbols, and rely on devices that implement these algorithms using transistors and integrated circuits. Progress is typically measured simply by how many operations can be performed in one second. In some sense this is like measuring the beauty of a painting by how many brushstrokes are used.

From a broad (SAD) perspective, as we will see, much of today's computing infrastructure is based upon symbols from the 18th century, algorithms from the 19th century, and devices from the 20th century. Of course the marketing is from the 21st century, so people today can be proud of something.

1.5 Why Symbols Matter

> All is Number.
>
> Attributed to Pythagoras

> Euclid alone
> Has looked on Beauty bare.
>
> Edna St. Vincent Millay (1892–1950)

Computation is intimately tied up with the symbols we use. Symbols and computation are so interwoven that it is easy to take for granted the power provided by innovative symbols. In this section, we discuss a few examples of how innovations involving symbols can dramatically simplify computations.

By the time a student today enters high school, she has written both large numbers (e.g. $299,792,458$) and fractions (e.g. 3.1415926535) using positional notation; she has used scientific notation (e.g. 6.38×10^{27}) for computations; and she has used symbols to represent numbers (e.g. let x be a number whose square is 16) and geometric quantities (e.g. let x be the radius of a circle whose circumference is 10).

What a student doesn't always appreciate is how relatively recent some of these innovations are: for example, the transition in Europe from Roman numerals to the positional Hindu-Arabic system took several centuries and did not become common until the 16th century; logarithms were not introduced until the 17th century; and symbols to represent numeric and geometric quantities were also introduced in the 17th century.

In classical antiquity, the Babylonians, Greeks, Egyptians, and Romans each used different symbols and number systems. The Babylonians used a positional number system with a base of sixty, which we have inherited to this day for measurements involving navigation, astronomy, and time. For example, there are sixty seconds in a minute and sixty minutes in an hour. The sun was observed to require about 360 days to complete a circle, so a circle was divided into 360 degrees, and each degree was divided into 60 minutes ('), each minute into 60 seconds ("), and each second into sixty thirds ("').

Archimedes (c. 287 BC – c. 212 BC) did not have the use of the positional number system. He wrote a paper called the Sand Reckoner in which he tried to estimate the number of grains of sand that would fill the earth. It was not an easy computation and would have been much simpler if he had used the positional number system that we take for granted today.

The Greeks during the time of Archimedes used the letters to represent number as follows: $1, 2, 3, \ldots, 9$ were represented by the letters alpha through theta; $10, 20, 30, \ldots 90$ were represented by the letters iota through koppa (koppa is not part of the current Greek alphabet); $100, 200, 300, \ldots, 900$ were represented by the letters rho through san (san is not part of the current Greek alphabet). For example, the number 222 was written sigma kappa beta.

To represent numbers greater than 999, subscripts and superscripts were used. For example, adding the Greek letter iota as a subscript or superscript to the letters alpha, beta, ..., theta, produced the numbers 1000, 2000, $\ldots, 9000$.

Rules for adding and multiplying using this type of alphabetic system were more complicated than the familiar rules today. Just think for a moment how hard it would be to estimate the number of grains of sand in the earth using this number system.

Numbers in the fifth century BC were thought of geometrically. This way thinking can be seen in Euclid's El-

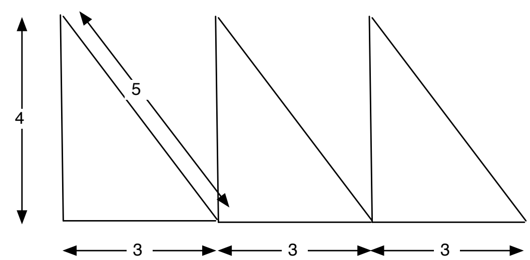

Figure 1.2: The Greek Parthenon is 69.5 meters longs, 30.88 meters wide and 13.72 meters tall, exactly the same proportions that you get if you take three 3x4 rectangles and lay them end to end as indicated. Note that the length of the diagonal of a 3x4 rectangle is 5.

ements, which captured the geometry of the 4th century B.C. Greeks. This was a tremendous achievement and provided fundamental insights and algorithms for a variety of different problems, both theoretical and practical.

A good example of how the Greeks viewed numbers as geometric lengths and ratios is provided by the Parthenon [115]. The Parthenon is 69.5 meters long, 30.88 meters wide, and 13.72 meters high. This means that the ratio of the width to the length is 30.88/69.5 or about 4/9, while the ratio of the height to the width is 13.72/30.88 or about 4/9. These are the dimensions you would get if you took three rectangles of length 3×4 and placed them side by side, as in the diagram below. Note that the diagonal of a 3×4 rectangle is of length 5, since by the Pythagorean Theorem $3 \times 3 + 4 \times 4 = 5 \times 5$.

Another significant advance in computing which is easy to take for granted is the introduction in the seventeenth century of symbols for unknown quantities, such as the variable x. With these types of symbols, equations such as 2.2 x = 32.8 could be easily represented, as could geometric objects such as the circles, ellipses, and hyperbolas.

Just as today's positional number system enables computations that would be simply impractical with the Greek or Roman number system, it is instructive to imagine new types of symbols that might provide a similar advantage over today's use of binary symbols to represent positional numbers.

Here is a simple example. For over thirty years, computer scientists have been building systems for what is called *symbolic computation*. These systems manipulate symbolic as opposed to numeric entities. Using such a system, one can multiply two polynomials like $x+2$ and $x+3$ to compute their product x^2+5x+6. Simple versions of these systems are now found in calculators.

Another example is in logic programming, where systems work with assertions and rules to draw conclusions from them.

In Chapter 5, we discuss some of the ways multimedia digital data, such as images and audio files, are created. Working with these digitally requires symbols for encoding images and sounds, as well as devices for taking light waves and audio waves and producing discrete symbols, such as the symbols (135, 206, 235) or #82CAFF to represent the color sky blue in HTML documents.

Today we use symbols from different alphabets to form expressions which fill files to record a wide variety of things, including shapes (vector graphics SVG files), colors (JPEG files), sounds (MP3 files), and video (MPEG and DVD files).

From this perspective, our ability to compute depends critically upon the richness of the symbols we use and the power of the algorithms we use for transforming the symbols.

1.6 Algorithms as Recipes for Manipulating Symbols

> Science is what we understand well enough to explain to a computer. Art is everything else we do.
>
> Donald Knuth

Having fixed a collection of symbols, the next question is what do we do with them? What are the rules and systems for manipulating them? Algorithms may be thought of as formal procedures for manipulating symbols. For many problems, such as predicting the trajectory of a projectile, factoring prime numbers, or predicting tomorrow's weather, algorithms affect the speed of computation just as much, or more, than the hardware of the computing platform.

It is easy to explain the basic idea of an algorithm using an example. Recall that x is called a square of a in case x times x is equal to a. For example, 2 is the square root of 4, 3 is the square root of 9, and 5 is the square root of 25.

Here is a simple algorithm for computing the square root of a number a.

1. Begin with a guess x for the square root.
2. Replace x by the average of x and a/x.
3. Go to Step 2.

Here are some approximations to the square roots of 2, 5 and $5,934,939$ using a five line Python program that you can find in the notes:

```
sqrt of 2
1.5
1.41666666667
1.41421568627
```

```
1.41421356237
1.41421356237
...

sqrt of 5
2.25
2.23611111111
2.23606797792
2.2360679775
2.2360679775
...


sqrt of 5,934,939
1483735.75
741869.874999
370938.937486
185477.46863
92754.7333983
46409.3593468
23268.6208599
11761.8413875
6133.21703089
3550.44423902
2611.0244305
2442.02762481
2436.18004142
2436.17302342
2436.17302341
2436.17302341
...
```

The three dots indicate that the last number repeats.

As can be seen from this example, an algorithm consists of a series of steps each of which can be carried out explicitly once the previous steps are completed. More or less, one can think of an algorithm as anything that can be expressed by a computer program.

The notes for this section contains another algorithm called Newton's Method, which can be used to find the solutions of a wide class of equations.

1.7 Computing Devices

> [Mathematical] Tables have been with us for some 4500 years. For at least the last two millennia they have been the main calculation aid, and in dynamic form remain important today. Their importance as a central component and generator of scientific advance over that period can be underestimated by sheer familiarity. Like other apparently simple technological or conceptual advances (such as writing, numerals, or money) their influence on history is very deep.
>
> The History of Mathematical Tables: From Sumer to Spreadsheets, edited by M. Campbell-Kelly, M. Croarken, R. Flood and E. Robson.

It is easy to take paper, pens and pencils for granted, but they are important computing devices. Twenty five hundred years ago, Babylonian computation was done in clay, Chinese computation was done using bark and bamboo, and Egyptian computation was done using papyrus.

For perspective, Homer's Iliad and Odyssey were not initially written down, but passed on instead as an oral tradition. Indeed, they were composed at a time when the Greek alphabet was still emerging and when papyrus was scarce.

Papyrus was an early form of paper made from the papyrus plant, which grew mainly in Egypt and had to be imported by other regions. Egyptian papyri document some of the earliest algorithms that we know of. For example, what is sometime called the Rhind papyrus was written about 1650 BC by the scribe Ahmes. It is about 20 feet long by 1 foot wide and contains 87 problems, including problems requiring multiplying numbers and working with fractions. For example, Problem 4 shows how to divide 7 loaves of bread between 10 men.

Parchment, which is made from animal skins, was also used for writing, and may have been invented in part because of the difficulty obtaining papyrus outside of Egypt. Parchment can be made from a number of animal skins, including those of calves, sheep or goats. Although humans have used animal skins since paleolithic times, the preparation and use of animal skins for writing is much more recent. Parchment was invented about third or second century BCE in Pergamum (modern day Turkey). Parchment became generally available in the Hellenistic world during the first century AD. Parchment was used extensively through the middle ages, and is still sometimes used to this day for diplomas or other special documents.

Euclid's Elements is one of the most important books containing Greek mathematics. Early versions were probably written in papyrus, with later versions written in parchment. The earliest version still extant is in parchment and was written in 888 AD, almost 1200 years after the original.

Paper was invented in China about 105 AD. Paper is made from fibers extracted from wood pulp, from trees such as spruce or pine trees. Paper can also be made from fibers extracted from other sources, such as cotton or rice. In Europe, paper began to replace parchment during the middle ages.

Moving from clay and wood to parchment and paper was one of the first significant advances in computing devices. On the other hand, the Babylonian computations preserved on clay proved to be more durable, which is why we know a bit more about how the Babylonians computed during this time than how some of their contemporaries did.

Paper is still a wonderful device for computing. It is one of the most flexible devices and supports not only numerical computation, but also computation involving algebraic, geometric and logical symbols. Over the last few hundred years, it has been augmented by wooden devices, such as slide rules; mechanical devices, such as adding machines built from gears; and electronic devices, such as a calculators and computers built from vacuum tubes, transistors, and integrated circuits.

There is no reason to expect the development of new computing devices to slow down. Not only do integrated circuits continue to improve, but so does the exploration of novel devices, such as devices that use genes to compute or that exploit quantum mechanics.

1.8 Case Study: Slide Rule

> When I was research head of General Motors and wanted a problem solved, I'd place a table outside the meeting room with a sign: LEAVE SLIDE RULES HERE! If I didn't do that, I'd find some engineer reaching for his slide rule.
>
> Charles F. Kettering (1876-1958)

The slide rule is a computing device that enormously sped up the computation of a sequence of multiplications and divisions, as well as a variety of other computations, such as extracting square roots. Slides rules were introduced in the 17th century and were an important computing device for over three hundred years, a reign significantly longer than the modern digital computer.

Prior to slide rules, the device most often used for mathematical computations was probably a mathematical table. Mathematical tables are one of the earliest computing devices. We have examples of tables used by Babylonians dating from about 2000 BC.

Tables are still used in mathematics to this day, and so are in the running for one of the computing devices with the longest staying power. Tables are easy to use. For example, using the table of sines below, we can read off the sine of 80 degrees as 0.98481 by reading the table from left to right. We can also read the table from right to left to see that the

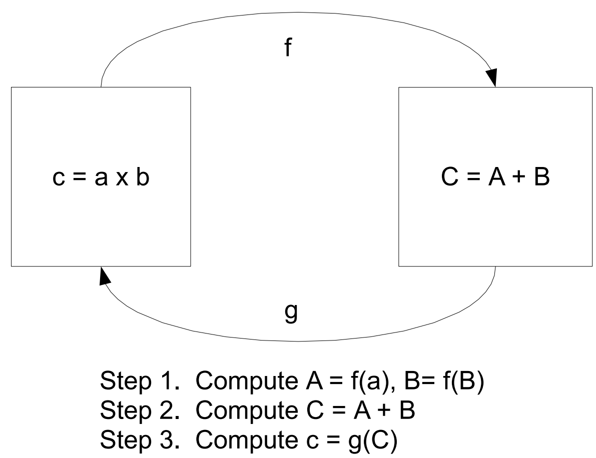

Figure 1.3: The first idea underlying the slide rule is that there are certain functions (f and g in the diagram) that transform multiplication and division into addition and subtraction. John Napier discovered such a pair of functions in the early 17th century.

number whose sine is -0.34202 is 200 degrees. Reading the table from right to left computes a function which undoes the sine (the function is called the arcsine). To compute the sine of 50 degrees, we know from the table that it is between 0.64279 and 0.86603 (the sines of 40 degrees and 60 degrees). If we take the midpoint of these two numbers we get 0.75441, which is close but not exactly the sine of 50 degrees which is 0.76604. Over the years, a number of formulas have been developed to interpolate between two values in a table to get more accurate answers.

There are two basic ideas underlying the slide rule:

The first idea is that certain functions can reduce multiplication and division to addition and subtraction. In 1614, John Napier (1550-1617) introduced a very nice function (a variant of today's logarithm function) with the property of reducing multiplications to additions. See Figure 1.3.

The second idea is due to Edmund Gunter (1581-1626).

degrees	**sine**
0	0.00000
20	0.34202
40	0.64279
60	0.86603
80	0.98481
100	0.98481
120	0.86603
140	0.64279
160	0.34202
180	0.00000
200	-0.34202
220	-0.64279
240	-0.86603
260	-0.98481
280	-0.98481
300	-0.86603
320	-0.64279
340	-0.34202
360	0.00000

Table 1.1: This is a simple mathematical table. If you read from left to right, you can compute the sine of various numbers. If you read from right to left, you can compute another function, called arcsine, that is the inverse of the sine function.

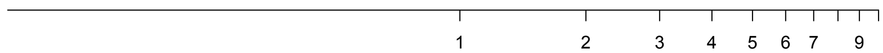

Figure 1.4: Edmund Gunter positioned numbers along a piece of wood according to their logarithms. The Gunter scale was used by seamen to simplify computations for navigation.

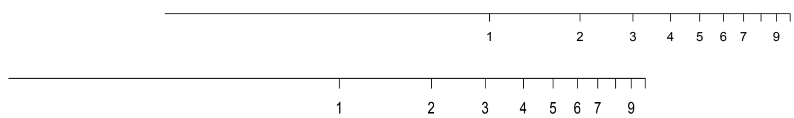

Figure 1.5: About 1625, William Oughtred put together two Gunter scales, allowing one to slide next to the other to create the slide rule.

His idea was to create a scale on wood in which numbers were marked according to their logarithms. See Figure 1.4

About 1625, William Oughtred (1575-1660) put together two Gunter scales so that one could slide next to the other, creating the slide rule. Slide rules were one of the staples of computation until the introduction of handheld calculators in the 1970s.

To end the section, let us try to characterize the slide rule from the symbols-algorithms-devices (SAD) perspective:

First, let us think of the slide rule as a device. From this perspective, it was a true revolution. Prior to slide rules, computations were done using tables, for example, tables of sines, cosines, square roots, and logarithms. The slide rule allowed one device (mathematical tables) and several manual look-ups to be replaced by another device (the slide rule) and two operations (moving one wooden rule with respect to the other) and moving a reference line along the two slides. In other words, this elegant device replaced large mathematical tables and error prone manual look-ups. In fact, multiple tables could be encoded on the same device, so that a single slide rule could replace tables of square roots, cube roots, and logarithms.

Second, consider the algorithms involved. From this perspective, the slide rule is less innovative. Indeed, the algorithms used for a slide rule are essentially the same algorithms used previously with mathematical tables. The first algorithm is a simple look-up (reading a table from left to right) or reverse look up (reading a table from right to left). The second algorithm is using a function like the logarithm to replace multiplications and divisions by additions and subtractions.

Third, consider the symbols involved. Again, from this perspective, the symbols used in mathematical tables and the symbols used in slide rules are essentially the same: positional numbers (like 1.414213562).

Although the computing devices we describe next don't have the staying power of mathematical tables (4000+ years) or slide rules (300+ years), they do have the advantage that they can also be used for playing music and writing emails to your family and friends. On the other hand, neither mathematical tables nor slide rules require batteries or electricity and both function perfectly well a decade after production.

1.9 From Mainframes to Devices

> The longer you look back, the further you can look forward.
>
> — Winston Churchill

> It is difficult to tell a short-sighted man how to get somewhere. Because you cannot say to him: “Look at that church tower ten miles away and go in that direction.”
>
> Ludwig Wittgenstein

One of the themes of this book is that digital computing for the past fifty years or so has been shaped in large part by the process of commoditization. To understand commoditization better, it is useful to divide the past fifty years into four eras, consisting of overlapping 20 year periods.

1. The Mainframe Era (1965 - 1985). In 1965, IBM shipped the System 360, its first computer based on integrated circuits. By 1968, it had installed over 14,000 System 360 systems, at an average price of over $1,000,000 per system, generating over $14 billion dollars of revenue for IBM. During the Mainframe Era, computing cycles were limited. Departments were built around the mainframes to operate them, ration their cycles, and provide services to the users.

2. The Personal Computer (PC) Era (1980-2000). In 1977, Apple, Commodore and Tandy began selling personal computers. In 1981, IBM entered the market, effectively setting the standard, with a business model which encouraged third party manufacturers. For example, Compaq shipped its first IBM clone in 1983 and set a record by selling $111 million of PCs, one of the largest first year product sales in the history of American business. By the end of this era, over 50 percent of the workers in some metropolitan areas had PCs. PCs had also became building blocks for specialized computing needs: for example, rather than build specialized devices for supercomputing, supercomputers were being built by putting together hundreds or thousands of PCs into what are

called clusters. During the PC Era, computer hardware became a commodity.

3. The Web Era (1995-2015). In 1993, Mosaic, the first graphics based web browser, was released. It was developed at the National Center for Supercomputing Applications at the University of Illinois in Urbana Champaign. In 1995, NSF turned over the networking backbone for the Internet to commercial vendors and, at the same time, introduced a research network called the very High Speed Backbone Network Service or vBNS, that became the foundation for the next generation of the Internet. By 1996, the number of hosts on the Internet exceeded 10 million. By 2000, the number exceeded 75 million hosts. With the Internet, it became much easier to develop and deploy software applications that essentially used the Internet as the operating system. Companies began to give away software. For example, Hotmail was a free email program and had over 30 million users two and half years after its launch in 1996. During the Web Era, software became a commodity.

4. The Device Era (2005-2025). There is no generally accepted name for the post-web era. In this era, PCs connected to the web will be supplemented by a wide variety of networked devices, many of which are wireless. These devices include mobile phones, personal digital assistants (PDAs), devices for listening to music, cars emitting diagnostic information, cameras, home stereos, as well as a variety of other devices not yet prototyped. At the beginning of the device era, wireless Internet is available at coffee shops and bookstores, and some cities are providing free wireless Internet. During the Device Era, networks will become a commodity.

Dividing the digital age into four eras simply provides convenient markers for certain inflection points. For example, although the web era is defined as the twenty year period from 1995-2015, the roots of the web era date back at least to the sixties. For example, the first paper on packet switching, which is one of the main ideas for the Internet's network architecture, was published by Leonard Kleinrock from MIT in 1961. In 1967, Larry Roberts of ARPA organized the first meeting to design what would become ARPANET, the predecessor of the Internet. In 1969, ARPANET began operations with nodes at UCLA, the Stanford Research Institute (SRI), the University of California at Santa Barbara (UCSB) and the University of Utah.

In other words, the divisions between the different eras should be thought of as simply convenient markers denoting a time in which the prior era's technology is widely diffused, and the next era is well on its way. For example, by 1996, near the beginning of the web era, there were over 10 million hosts on the Internet. By this time, your father and your grandfather were likely to have heard of the Internet.

1.10 The First Era: Mainframes

From today's viewpoint, life with the first computers seems to belong to a simpler, more romantic time: there was basically one type of computer — the mainframes — and one dominant company — IBM. Of course, if you look deeper, there were specialized computers for specialized purposes, and several companies trying to gain the dominant market share.

More importantly, though, the computing was centralized, and the control of the system was firmly in the hands of the professional computing staff. Individuals accessed information through terminals that were networked to the mainframe. To get new information from the system or old information presented in a new way, one submitted a written request which eventually either led to what was wanted or to a further request. On the positive side, the mainframe with its centralized structure meant that the information was secure and that operations were managed by professionals. One also knew who to blame when things went wrong.

During this period, mainframe computers primarily managed corporate level information, such as employee records, accounting data, and product data.

In a sense that will become clearer as we consider the second and third eras, the first era can be characterized as being hardware-limited. The challenge was to provide the appropriate hardware to the customer. Companies such as IBM, which provided this hardware and the associated services, grew to be quite large.

1.11 Case Study: Punch Cards

Old technologies are a bit like photographs — you never really throw them away, you just spend less time with them. In the second and third eras, the era of PCs and the Web, many companies still use mainframes for important business processes, such as the preparation of payrolls.

Not only are mainframes still critical in producing payrolls but so are punch cards. In the 1880's, the punch card was invented by Herman Hollerith to automate the US Census. Holes in a punch card are translated by a punch card reader into mechanical or electrical signals that can be counted or further processed.

One of the most common uses of punch cards was to compute payroll. For example, the garment industry used punch cards as follows. A punch card was created each time an individual working on a garment performed a particular operation in the manufacture of the garment. If a worker performed 100 operations per day for five days, this generated 500 punch cards. One hundred employees would generate 5000 punch cards per week, which would be tabulated to produce the week's payroll.

In the 1970's, disk drives began replacing the punch cards as the main storage mechanism for data. Today, it is very difficult even to find a punch card reader. On the other hand, systems using them as the basic form of data entry still persist. In 2002, the New York Times ran a story describing how the 23,000 employees of the Los Angeles County Department of Health Services still use punch cards for their time card and payroll system [83]. Punch cards are still used because the system still works and moving all the hospitals and clinics to a new system would be too expensive.

In 1967, an estimated 260 billion punch cards were used in the US or 1,300 for each American. Today, there are only three companies in the US that manufacture punch cards. On the other hand, punch cards are still used for many voting machines, as became apparent in the 2000 Presidential election controversy.

1.12 The Second Era: PCs

The emergence of microprocessor-based computers ushered in a new era because for the first time hardware became a commodity. During this era, personal computers covered more and more desk tops. Companies that provided useful desktop applications such as word processors, spreadsheets, and databases grew. Hardware was distributed and computing was based on the client-server model in which larger systems (servers) provided data and information to smaller systems, such as personal computers (clients) networked to them.

Individuals could suddenly buy, install, and use desktop applications without the lengthy involvement of a centralized computing department. On the other hand, individuals were suddenly responsible for managing and servicing their desktop systems and protecting their desktop data.

The information on the desktop was, by and large, department level information generated by individuals: reports, presentations, budgets, and other work related data.

The VisiCalc spreadsheet was one of the first new types applications to be designed for the PC. VisiCalc was designed and developed by Dan Bricklin and Bob Frankston. Bricklin and Frankston formed Software Arts, Inc. on January 2, 1979, and worked out of their attics. There was very little infrastructure for developing applications on the PC in those days. The first version of VisiCalc was written for the Apple II in assembler, a low level language that required quite a bit of code for such things as reading input from the keyboard, saving information to a file, or reading information from a file. With higher level languages these days, these operations can all be done with a single line of code.

If you grew up using spreadsheets, it may be a bit difficult to imagine how business was conducted before they were introduced. The first business application of VisiCalc took place while Dan Bricklin was getting an MBA at the Harvard Business School in 1979. MBAs at Harvard learned about business through case studies. Bricklin used an early version of VisiCalc to analyze the Pepsi Challenge marketing campaign for the Pepsi Corporation case study. While the other students in the class used a Texas Instrument Business Analyst calculator, Bricklin used VisiCalc, allowing him to do five year forecasts and vary many of the assumptions. When the professor teaching the class asked Bricklin how he had done all the projections, Bricklin answered in a way that didn't disclose that he had used an entirely new type of product, since he wanted to keep the VisiCalc secret until it was closer to being released.

A few years later, spreadsheets would transform many business processes. For example, Kohlberg Kravis Roberts & Company, commonly referred to as KKR, is a New York City-based company that popularized the leveraged buyout. One of their secret weapons was using spreadsheets to analyze quickly the cash generated under various scenarios when the acquired company was split apart in different ways. This was very important, since KKR used the acquired company's own cash for the buyout, and financed the rest by issuing high yield (also known as junk) bonds.

In some sense, the PC Era was limited by the availability of the appropriate desktop software and the challenge was to provide this software to the customer. Companies such as Microsoft that met this challenge became the icons for this era.

1.13 The Third Era: The Web

A typical software application during the PC Era was Microsoft Word, a word processor. A typical software application in the Web Era is Google's GMail, an email program. Software from the PC Era resides on your desktop computer or your laptop. Software from the Web Era is often on another machine and you access it through a web browser over a network.

The significance of the world wide web is not only that one can click to bring up an interesting picture or that one can buy a book over the web, but rather that information and services from millions of servers on the network are just as easily available as if they were your own desktop PC.

The icons of this age are not the hardware vendors — IBM sold its PC business in 2004 — nor the software vendors — much of the software infrastructure, such as the Apache web server, is open source and not sold by a software vendor but instead developed by a distributed team of volunteers — but rather providers of web-based services, such as Amazon, eBay, and Google.

Think of it this way: someone buying a book on Amazon is more aware of how slow her network connection is than what the model of her PC is or which software vendor developed her web browser. Over a two year period, the book buyer will easily pay more for her DSL-based network connection (2 years x 12 months per year x 50 dollars per month or $1200) than for her PC clone ($800) or her open source Mozilla Fire Fox web browser ($0). During the web era, hardware and software are commodities, but network access and web based business services are still a scarce resource.

During the web era not only is network access a scarce commodity but so is information. This is a bit counter-intuitive, since suddenly not only is everything on your own computer accessible, but, via the web, so is much of the information on the computers of colleagues and strangers. The problem, though, is that this information is rarely organized well enough so that you can readily use it.

Think for a moment about how information is organized on your own desktop or laptop computer. For most of us, organizing information on our computer is put on a to-do list almost as frequently, and no more effectively, than losing ten extra pounds of weight. Suddenly, not only must your computer be organized, but so must the computers of all your colleagues, as well as those of strangers you have never met.

In other words, although the amount of digital data is growing, the amount of useful information is not keeping up. The technology to screen, sort, and extract useful information from large amounts of digital data is not yet ready for everyday use. This is the challenge of the next era. Companies that succeed with these challenges, such as Google, are candidates to be the icons of the next era.

1.14 Case Study: SMTP

This section contains a case study of the Internet protocol that powers email. An Internet protocol is roughly speaking a “language” that allows two computers on the Internet to communicate. There are a large number of different Internet protocols, including those that support email, web browsing, and setting up telephone calls over the Internet. Email is commonly considered as the *killer application* or killer app that sparked the adoption of the Internet among consumers.

Many users access email through an email application such as Microsoft's Outlook or Outlook Express. Outlook is a proprietary client that not only interfaces to proprietary Microsoft products and services but also interfaces to open protocols and services, such as the Simple Mail Transfer Protocol or SMTP. Although most people have never heard of it, SMTP is one of the main reasons that email became a killer app.

SMTP is a computer-to-computer (C2C) protocol that allows two computers on the Internet to exchange email messages [126]. Here is an example from the original 1982 description of SMTP [126]:

```
S: MAIL FROM:
R: 250 OK

S: RCPT TO:
R: 250 OK

S: RCPT TO:
R: 550 No such user here

S: RCPT TO:
R: 250 OK

S: DATA
R: 354 Start mail input; end with .
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK
```

EMail is exchanged when an email client connects to a SMTP server. The SMTP server is a computer that provides an Internet or web service. In particular it listens for requests from SMTP clients. Clients include proprietary Microsoft products such Outlook, as well as browser-based email, such as those provided by Google, Yahoo or Hotmail.

In the computer-to-computer conversation above, Smith (the sender S) at the computer alpha.arpa tries to send email to Jones (the receiver R) at beta.arpa and Green (the receiver R) at beta.arpa. He succeeds with Jones but not with Green. The email is sent as an ASCII text stream.

The computer-to-computer conversation takes place between two different computer programs residing on two different nodes on the Internet without human intervention. A special case is when both programs reside on the same node. The conversation between the SMTP-Sender and the SMTP-Receiver is pretty simple and consists of three steps.

In the first step, the SMTP-Sender sends a mail command:

```
MAIL  FROM:
```

This command tells the SMTP-Receiver that a new message is being sent. The reverse path is used so that the SMTP-Receiver can return a 250 OK reply after successfully processing the command. In the second step, the SMTP-Sender sends a RCPT command:

```
RCPT  FROM:
```

This command gives the address of the one recipient of the message. If the SMTP-Receiver accepts the recipient, it returns a “250 OK”; if not, it returns a “550 No such user here”. This step is repeated once for each user the message is sent to.

In the third step, the SMTP-Sender sends a DATA command containing the message itself:

```
DATA
```

If the SMTP-Receiver accepts this commands it returns a “354 Intermediate Reply”. Once the SMTP-Sender receives the 354 command, it begins sending data as a simple ASCII stream. The end of the ASCII text stream is indicated by sending a single line consisting of a single period (“.”) followed by a carriage return and line feed. Once the SMTP-Receiver receives the period, it sends a final 250 OK command to finish the computer-to-computer session.

Notice that the FROM, TO, DATE, SUBJECT, CC, BCC, and other fields in a standard email message are all sent as part of the data and have no special meaning to the SMTP-Sender or SMTP-Receiver. The mail client program, such as Outlook or a Yahoo web mail, extracts the mail addresses from these fields and passes them to the SMTP-Sender.

Over time, mail was used to send a variety of attachments in a variety of formats, from Microsoft Word documents to jpg images. To handle this, a standard was developed called the Multimedia Internet Mail Extensions (MIME) Encoding. MIME is a way for binary data such as Microsoft Word documents or graphic images to be encoded as ASCII text. Once encoded in this way, it can be sent using standard SMTP-Senders and SMTP-Receivers. The same MIME encoding is used today by both SMTP and HTTP.

1.15 The Fourth Era: Clouds of Devices

In the fourth era, hardware cycles and software applications have already become commodities. This era is about the transition from browser-based applications to devices; from copper-based networks to fiber-based and wireless networks; and from an application-based software model to a service-based software model.

The icon of the third era is a PC running a web browser. The problem is that 1990's style PCs were complicated to operate, had to be tethered to networks, and came in basically one style and color. The icon of the fourth era is a wireless pocket device supporting email and instant messaging. The device doesn't need to be booted, simply turned on; it doesn't need a network cable, simply a network; and it doesn't need an operations manual, simply an operator.

In the fourth era, rather than use a single PC running desk top applications in your office, you are more likely to carry several small independent devices, such as email devices, cell phones, music and video players, and games, all connected on wireless IP networks and providing various services. In the same way that a telephone today provides a simple service (you dial a phone number and talk to someone), these devices will also provide equally simple services (you enter an email address, a short note, and push send).

By the end of the fourth era computing devices with embedded wireless networking and general positioning systems (GPS) capabilities will be smaller than postage stamps and cost about the same. They will be included in automobiles to provide early warnings of engine breakdown, attached to bridges to aid in maintenance, and used to keep track of your young children.

In the first two eras, the computer was firmly at the center of the model. Attached to the computer were various peripheral devices, such as terminals, printers and disk drives. In the third era this began to change, with the network moving towards the center. By the fourth era, this transition should be complete, with computer routers and switches firmly at the center of the model, and with CPUs, disk drives, and wireless devices simply peripherals. Sometimes this model is called the "hollowed out computer."

There is no agreed upon name for the fourth era. The fourth era is full of *devices* providing clouds of services over an ever present and ubiquitous IP network. For simplicity, in this book, we will refer to it simply as the Device Era. You should expect this *name* to seem quaint and old fashioned by the time you are reading this book, but I would be surprised if the essence of the era had changed in a significant way.

What is scarce in the fourth era is a way to manage the data and the information produced by the myriad devices and their associated services. The upcoming fifth era is based upon the emerging ability to extract useful information from this data and to structure it in a way that leads to useful decisions.

1.16 Case Study: Routers

This case study is about a piece of hardware called a router that allows two or more computer networks to connect together. The development of the router was one of the critical events that enabled the creation of the Internet.

Viewing a web page on the Internet requires that your computer send messages back and forth with another computer, which is called the web server. In an earlier section, we learned that just as every telephone in the world has a unique phone number, every computer on the Internet has a unique address, called its Internet or IP address.

Very roughly, the Internet, from the network perspective, is a collection of local area networks that are connected together with routers and that communicate with a common set of protocols. These protocols divide data into chunks called packets and attach the IP address of the source and the IP address of the destination to each packet.

As more and more local area networks connected to the Internet, more and more routers were required. The company CISCO was an early supplier of routers. It was founded in 1984. It sold approximately 5,000 routers in 1990 and over 900,000 in 1997. By about 2002, routers could process millions of packets per second, and by 2006, they could process billions of packets per second. Routers contain tables that may have over a hundred thousand entries describing how to route an incoming packet.

Here is how they work in a bit more detail:

Suppose that Computer A in local area network L1 wants to send a message to Computer B on local area network L2. Suppose also that the two local area networks are connected together with a router R having three ports. In this context, a port is a number used to distinguish two or more physical connections of a network to a router. Port P1 is connected to local area network L1, Port P2 is connected to local area network L2, and Port P3 is connected to the local area network of an Internet Service Provider.

Here is what router R does every time it processes a packet.

1. Computer A first breaks the message into units which are each 1500 Bytes long. For simplicity, assume that the message is short and consists of a single 1500 Byte data packet.

2. Given the name of Computer B, Computer A looks up the corresponding IP number for Computer B using a network service called the Domain Name Service or DNS.

3. Computer A then assembles a packet [IP(A), IP(B), data], where IP(A) is the source Internet address of computer A, say a.b.c.d, IP(B) is the destination Internet address of Computer B, say e.f.g.h, and data is the data packet for the message.

4. Computer A looks at the destination address e.f.g.h and determines that it is is not a packet on its own local area network. If it were, the last part of the address would be of the form b.c.d. It therefore sends it to a specific computer called the default gateway, which is specified in the network configuration software of Computer A. Call the default gateway Router R.

5. To send the packet from A to the Router R, the Computer A wraps the IP packet in a frame of the type required by the local area network. For example, if the local area network uses Ethernet, then [MAC(A), MAC(R), IP(A), IP(B), data, CRC]. Here MAC(A) and MAC(R) are the MAC address of the network interfaces for Computer A and Router R on the local area network L1.

6. When Router R gets the packet, it removes the Ethernet Frame to get [IP(A), IP(B), Data]. Router R looks at its router table and finds that IP addresses like B are sent to Port P2 on Router R.

7. To send the packet from Router R to Computer B on local area network L2, Router R places [IP(A), IP(B), Data] into a frame for local area network L2.

8. Computer B on local area network L2 receives the frame, extracts the IP Packet, and then extracts the data.

1.17 The First Half Century of Computing

Study the past if you would define the future.

Confucius

In this section, we broaden our point of view a bit and consider computing *platforms*, which include not only the computer itself, but the broader infrastructure required for computing, including operating systems, applications, storage devices, networks, network services, displays, peripherals, and various other devices that we use when computing.

Viewing computing platforms from the perspective of fifty years or so is difficult for several reasons:

- The different components of a computer and of a computing platform become commoditized at different rates. This is the subject of Chapter 2.

- We are surrounded by marketing clutter and the amount and type of clutter varies from component to component in the computing platform. This is the subject of Chapter 3.

- New technology gets adopted over a period of time. Different factors affect the rate at which different components of a computing platform are adopted. This is the subject of Chapter 4.

In this section, we briefly consider each of the difficulties in turn from the perspective of about fifty years.

Computers vs. Computing Platforms. As we just mentioned, computers are the most visible component of the computing infrastructure, but not the only component. What we do with computers depends upon the software applications that run on them, the operating systems they employ, the displays and peripherals they use, the networks that connect them, the data that flows through them, and the human-computer interfaces by which we interact with them. One way to to understand the differences between different computing platforms is to answer the following questions:

1. What is the hardware?
2. What is the software?
3. What is the network?
4. What is the user interface?
5. Where is the data?

Table 1.2 summarizes how each computing era has answered these questions differently. It is important to keep two things in mind:

1. First, there is no sharp division between one era and the next. Indeed, it is usually several years into the new era before there is a broad understanding of the nature of the technology of the new era, and a corresponding disappointment in all the utterances by the pundits about the transition.

2. Second, the computing platforms from prior eras never really fade away; instead, they continue to hang around, continue to be used, and continue to evolve. It is simply human nature to focus on what is new.

Scarce and plentiful resources. A simple means of distinguishing the different computing eras is to ask what component of the computing platform is the bottleneck, and hence is rationed by high prices, and what component is becoming a commodity, characterized by falling prices and rapidly increasing capacity. This is the subject of the second chapter. Table 1.3 provides a high level summary.

Technology adoption. Despite their name, computers do much more than compute. In fact, relatively few people use them for computing. Today a consumer at home is more likely to use a computer to send email, to buy books, to play games, or to listen to music than to perform a computation. Similarly, a business today is more likely to use a computer to do their accounting, to pay their bills, to manage their inventory, to keep the loyalty of their customers, or to create a marketing brochure.

Each time a new application or function appears, it usually takes much longer than is initially predicted to be adopted. In Chapter 4, we examine some issues that arise and limit the spread of new technology. The term *technology adoption life cycle* is often used to describe the process that limits the spread of new technology. The good news is that you can pay consultants to tell you that the adoption of new technology can be challenging. If you pay them enough, they will tell you that the color of your marketing brochure is wrong and that the tag line you are using in the brochure is not catchy enough. Unfortunately, sometimes the problems are more fundamental.

Question	First Era	Second Era	Third Era	Fourth Era
When?	1965-1985	1980-2000	1995-2015	2005-2025
What is the hardware?	mainframes	servers & PCs	personal computers	devices
What is the software?	back office applications, such as payroll	PC applications, such as word processors and spreadsheets	web applications, such as Amazon and Facebook	services on devices, such as listening to music or taking pictures with cell phones
What is the network?	terminals on serial lines	local area networks	wide area networks	wireless networks
What is the user interface?	terminals	PCs with windows, menus & mice	clicking buttons on browsers	pushing buttons on devices; voice
Where is the data?	in the mainframes	on servers	on desktops	in your pocket

Table 1.2: The first four eras of computing.

	Mainframe Era	**PC Era**	**Web Era**	**Device Era**
What is the bottleneck?	computer cycles	application software	network bandwidth	data
What is becoming commoditized?		computer cycles	application software	network bandwidth

Table 1.3: Viewing the four eras of computing by what is the bottleneck and what is a commodity.

1.18 The Fifth Era: The Commoditization of Data

The majority of books about technology that predict the future age very quickly and become irrelevant within a few years. With this in mind, and with the fifth of era of computing just beginning to emerge, it will be difficult to say very much about it. On the other hand, extrapolating the perspective of the section above, we can provide a rough characterization of some aspects of it:

- Just as the second era commoditized cycles and cylinders, the third era application software, and the fourth era bandwidth, the fifth will commoditize data.
- Once data is commoditized, new types of discovery in science and new types of decision support applications in business will become common.
- The scarce resource in the fifth era of computing will be those individuals with knowledge of how to leverage the technology — everything else will be commoditized or outsourced.

The Fifth Era of Computing will be the subject of Chapter 5. Chapter 5 will discuss the commoditization of data and the emergence of a computing platform that promises to change data-driven decision-making in the same way that during the Fourth Era, the Apple iPod changed how teenagers listened to music.