Chapter 2
Divide and conquer

How can ancient Sumerian history help us solve problems of our time?

From Sumerian times, and maybe before, every empire solved a hard problem – how to maintain dominion over resentful subjects. The obvious solution, brute force, costs too much: If you spend the riches of the empire just to retain it, why have an empire? But what if the resentful subjects would expend their energy fighting one another instead of uniting against their rulers? This strategy was summarized by Machiavelli [20, Book VI]:

> A Captain ought. . . endeavor with every art to divide the forces of the enemy, either by making him suspicious of his men in whom he trusted, or *by giving him cause that he has to separate his forces,* and, because of this, become weaker. [my italics]

Or, in imperial application, divide the resentful subjects into tiny tribes, each too small to discomfort the empire. (For extra credit, reduce the discomfort by convincing the tribes to fight one another.)

Divide and conquer! As an everyday illustration of its importance, imagine taking all the files on your computer – mine claims to have 1901629 files – and stuffing all of them into one folder. Dividing those million odd files into a hierarchy is the only hope for conquering so much complexity.

This reasoning tool dissolves difficult problems into manageable pieces. It is a universal solvent for problems social, mathematical, engineering, and scientific.

To master a physical tool, we use and analyze it. We see what it can do, how it works, and maybe study the principles underlying its design. Similarly, the reasoning tool of divide and conquer, the subject of this chapter, is introduced using a mix of examples and theory. There are four examples: CDROM design, oil imports, bank robbery, and the UNIX operating system.

There are three sections of theory: how to increase confidence in estimates; how to represent divide-and-conquer reasoning graphically; and how to explain the uncanny accuracy of divide-and-conquer reasoning.

2.1 Example 1: CDROM design

Our first use of the divide-and-conquer tool is from electrical engineering and information theory:

▶ *How far apart are the pits on a compact disc (CD) or CDROM?*

Divide the finding the spacing into two subproblems: (1) estimating the CD's area and (2) estimating its data capacity. The area is roughly $(10\,\text{cm})^2$ because each side is roughly 10 cm long. The actual length, according to a nearby ruler, is 12 cm; so 10 cm is an underestimate. However, (1) the hole in the center reduces the disc's effective area; and (2) the disc is circular rather than square. So $(10\,\text{cm})^2$ is a reasonable and simple estimate of the disc's pitted area.

The data capacity, according to a nearby box of CDROM's, is 700 megabytes (MB). Each byte is 8 bits, so here is the capacity in bits:

$$700 \cdot 10^6 \text{ bytes} \times \frac{8\,\text{bits}}{1\,\text{byte}} \sim 5 \cdot 10^9 \text{ bits}.$$

Each bit is stored in one pit, so their spacing is a result of arranging them into a lattice that covers the $(10\,\text{cm})^2$ area. 10^{10} pits would need 10^5 rows and 10^5 columns, so the spacing between pits is roughly

$$d \sim \frac{10\,\text{cm}}{10^5} \sim 1\,\mu\text{m}.$$

That calculation was simplified by rounding up the number of bits from $5 \cdot 10^9$ to 10^{10}. The factor of 2 increase means that $1\,\mu\text{m}$ underestimates the spacing by a factor of $\sqrt{2}$, which is roughly 1.4: The estimated spacing is $1.4\,\mu\text{m}$.

Finding the capacity on a box of CDROM's was a stroke of luck. But fortune favors the prepared mind. To prepare the mind, here is a divide-and-conquer estimate for the capacity of a CDROM – or of an audio CD, because data and audio discs differ only in how we interpret the information. An audio CD's capacity can be estimated from three quantities: the playing time, the sampling rate, and the sample size (number of bits per sample).

▶ *Estimate the playing time, sampling rate, and sample size.*

Here are estimates for the three quantities:

1. *Playing time.* A typical CD holds about 20 popular-music songs each lasting 3 minutes, so it plays for about 1 hour. Confirming this estimate is the following piece of history. Legend, or urban legend, says that the designers of the CD ensured that it could record Beethoven's Ninth Symphony. At most tempos, the symphony lasts 70 minutes.

2. *Sampling rate.* I remember the rate: 44 kHz. This number can be made plausible using information theory and acoustics.

   First, acoustics. Our ears can hear frequencies up to 20 kHz (slightly higher in youth, slightly lower in old age). To reproduce audible sounds with high fidelity, the audio CD is designed to store frequencies up to 20 kHz: Why ensure that Beethoven's Ninth Symphony can be recorded if, by skimping on the high frequencies, it sounds like was played through a telephone line?

   Second, information theory. Its fundamental theorem, the Nyquist–Shannon sampling theorem, says that reconstructing a 20 kHz signal requires sampling at 40 kHz – or higher. High rates simplify the antialias filter, an essential part of the CD recording system. However, even an 80 kHz sampling rate exceeded the speed of inexpensive electronics when the CD was designed. As a compromise, the sampling-rate margin was set at 4 kHz, giving a sampling rate of 44 kHz.

3. *Sample size.* Each sample requires 32 bits: two channels (stereo) each needing 16 bits per sample. Sixteen bits per sample is a compromise between the utopia of exact volume encoding (infinity bits per sample per channel) and the utopia of minimal storage (1 bit per sample per channel). Why compromise at 16 bits rather than, say, 50 bits? Because those bits would be wasted unless the analog components were accurate to 1 part in 2^{50}. Whereas using 16 bits requires an accuracy of only 1 part in 2^{16} (roughly 10^5) – attainable with reasonably priced electronics.

The preceding three estimates – for playing time, sampling rate, and sample size – combine to give the following estimate:

$$\text{capacity} \sim 1\,\text{hr} \times \frac{3600\,\text{s}}{1\,\text{hr}} \times \frac{4.4 \times 10^4\,\text{samples}}{1\,\text{s}} \times \frac{32\,\text{bits}}{1\,\text{sample}}.$$

This calculation is an example of a conversion. The starting point is the 1 hr playing time. It is converted into the number of bits stepwise. Each step is a multiplication by unity – in a convenient form. For example, the first form of unity is 3600 s/1 hr; in other words, 3600 s = 1 hr. This equivalence is a truth generally acknowledged. Whereas a particular truth is the second factor of unity, $4.4 \cdot 10^4$ samples/1 s, because the equivalence between 1 s and $4.4 \cdot 10^4$ samples is particular to this example.

Problem 2.1 General or particular?

In the conversion from playing time to bits, is the third factor a general or particular form of unity?

Problem 2.2 US energy usage

In 2005, the US economy used 100 quads. One quad is one quadrillion (10^{15}) British thermal units (BTU's); one BTU is the amount of energy required to raise the temperature of one pound of liquid water by one degree Fahrenheit. Using that information, stepwise convert the US energy usage into familiar units such as kilowatt–hours.

What is the corresponding power consumption (in Watts)?

To evaluate the capacity product in your head, divide it into two subproblems – the power of ten and everything else:

1. *Powers of ten.* They are, in most estimates, the big contributor; so, I always handle powers of ten first. There are eight of them: The factor of 3600 contributes three powers of ten; the 4.4×10^4 contributes four; and the 2×16 contributes one.
2. *Everything else.* What remains are the mantissas – the numbers in front of the power of ten. These moderately sized numbers contribute the product $3.6 \times 4.4 \times 3.2$. The mental multiplication is eased by collapsing mantissas into two numbers: 1 and 'few'. This number system is designed so that 'few' is halfway between 1 and 10; therefore, the only interesting multiplication fact is that $(\text{few})^2 = 10$. In other words, 'few' is approximately 3. In $3.6 \times 4.4 \times 3.2$, each factor is roughly a 'few', so $3.6 \times 4.4 \times 3.2$ is approximately $(few)^3$, which is 30: one power of 10 and one 'few'. However, this value is an underestimate because each factor in the product is slightly larger than 3. So instead of 30, I guess 50 (the true answer is 50.688). The mantissa's contribution of 50 combines with the eight powers of ten to give a capacity of $5 \cdot 10^9$ bits – in surprising agreement with the capacity figure on a box of CDROM's.

▶ *Find the examples of divide-and-conquer reasoning in this section.*

Divide-and-conquer reasoning appeared three times in this section:

1. spacing dissolved into capacity and area;
2. capacity dissolved into playing time, sampling rate, and sample size; and
3. numbers dissolved into mantissas and powers of ten.

These uses illustrate important maneuvers using the divide-and-conquer tool. Further practice with the tool comes in subsequent sections and in the problems. However, we have already used the tool enough to consider how to use it with finesse. So, the next two sections are theoretical, in a practical way.

2.2 Theory 1: Multiple estimates

After estimating the pit spacing, it is natural to wonder: How much can we trust the estimate? Did we make an embarrassingly large mistake? Making reliable estimates is the subject of this section.

In a familiar instance of searching for reliability, when we mentally add a list of numbers we often add the numbers first from top to bottom. For example: *12 plus 15 is 27; 27 plus 18 is 45.* Then, to check the result, we add the numbers in reverse: *18 plus 15 is 33; 33 plus 12 is 45.* When the two totals agree, as they do here, each is probably correct: The chance is low that both additions contain an error of exactly the same amount.

Redundancy, it seems, reduces errors. Mindless redundancy, however, offers little protection. As an example, if we repeatedly add the numbers from top to bottom, we are likely to repeat our mistakes from the first attempt. Similarly, reading your rough drafts several times usually means repeatedly overlooking the same spelling, grammar, or logic faults. Instead, put the draft in a drawer for a week, then look at it, or ask a colleague or friend – in both cases, use fresh eyes.

This robustness heuristic was in the Laser Interferometric Gravitational Observatory (LIGO), an extremely sensitive system to detect gravitational waves. It contains one detector in Washington and a second in Louisiana. The LIGO fact sheet explains the redundancy:

> Local phenomena such as micro-earthquakes, acoustic noise, and laser fluctuations can cause a disturbance at one site, simulating a gravitational wave event, but such disturbances are unlikely to happen simultaneously at widely separated sites.

Robustness, in short, comes from *intelligent* redundancy.