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.

This principle helps us make reliable, robust estimates. Not only should we use several methods, we should make the methods different from one another; for example, make the methods use unrelated knowledge and information. This approach is another use of divide and conquer (which may explain why the approach belongs in this chapter): The hard problem of making a robust estimate becomes several simpler subproblems – one per estimation method.

So, to supplement the divide-and-conquer estimate for the pit spacing (**Section 2.1**), here are two intelligently redundant methods:

1. An optics method is based on turning over a CD to enjoy and explain the brilliant, shimmering colors. The colors are caused by how the pits diffract different wavelengths of light. (Diffraction is beautifully explained in Feynman's *QED* [13].) For a pristine example of diffraction, find a red-light laser pointer, the kind often used for presentations. When you shine it onto the back of a CD, you'll see several red dots on the wall. These dots are separated by the diffraction angle. This angle, we learn from optics, depends on the wavelength (or color): It is λ/D, where λ is the wavelength and D is the pit spacing. Since light contains a spectrum of colors, each color diffracts by its own angle. Tilting the disc changes the mix of spots – of colors – that reach your eye, creating the shimmering colors.

   Their brilliance hints that the diffraction angles are significant – meaning that they are comparable to 1 rad. To estimate the angle more precisely, and lacking a laser pointer, I took a CD to a sunny spot and noted what appeared on the nearest wall: There was a sunny circle, the reflected image of the CD, surrounded by a diffracted rainbow. Relative to the reflected image, the rainbow appeared at an angle of roughly 30° or 0.5 rad. This data along with the diffraction relation $\theta \sim \lambda/D$ implies that the pit spacing is approximately 2λ. Since visible-light wavelengths range from 0.35 µm to 0.7 µm – let's call it 0.5 µm – I estimate the pit spacing to be 1 µm.

2. A hardware method is based on how a CD player or a CDROM drive reads data. It scans the disc with a tiny laser that emits – I seem to remember – near-infrared radiation. The *infrared* means that the radiation's wavelength is longer than the wavelength of red light; the *near* indicates that its wavelength is close to the wavelength of red light. Therefore, *near infrared* means that the wavelength is only slightly longer than the wavelength of red light. For the laser to read the pits, its wavelength should be smaller than the pit spacing or size. Since red light has

a wavelength of roughly 700 nm, I'll guess that the laser has a wavelength of 800 nm or 1000 nm and that the pit spacing is slightly larger – 1 μm. (The actual wavelength is 780 nm.)

Three significantly different methods give comparable estimates: 1.4 μm (capacity), 1 μm (optics), and 1 μm (hardware). Therefore, we have probably not committed a blunder in any method. To make that argument concrete, imagine that the true spacing is 0.1 μm. Then three *independent* methods all contain an error of a factor of 10 – and each time producing an overestimate. Such a coincidence is not common. Although any method can contain errors – the world is infinite but our abilities are finite – the errors would not often agree in sign (being an over- or underestimate) and magnitude.

The lesson – that intelligent redundancy produces robustness – seems plausible now, I hope. But the proof of the pudding is in the eating: What is the true pit spacing? It depends whether you mean the radial or the transverse spacing. The data pits lie on a tremendously long spiral track whose 'rings' lie 1.6 μm apart. Along the track, the pits lie 0.9 μm apart. So, the spacing is between 0.9 and 1.6 μm; if you want just one value, let it be the midpoint, 1.3 μm. We made a tasty pudding!

Problem 2.3 Robust addition

The text offered addition as an example of intelligent redundancy: We often verify an addition by by redoing the sum from bottom to top. Analyze this practice using simple probability models. Is it indeed an example of intelligent redundancy?

Problem 2.4 Intelligent redundancy

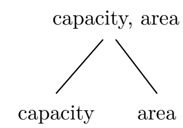
Think of and describe a few real-life examples of intelligent redundancy.

2.3 Theory 2: Tree representations

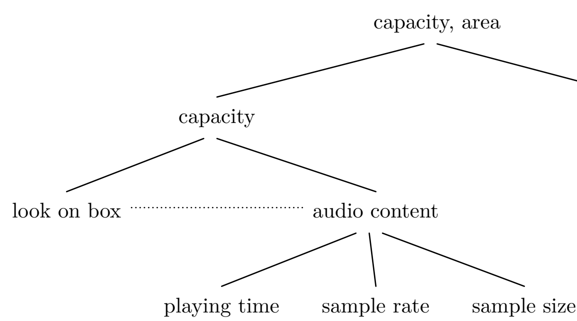
Tasty though the estimation pudding may be, its recipe is long and detailed. It is hard to follow – even for its author. Although I wrote the analysis, I cannot quickly recall all its pieces; rather, I must remind myself of the pieces by looking over the text. As I do, I am reminded that sentences, paragraphs, and pages do not compactly represent a divide-and-conquer estimate.

Linear, sequential information does not match the estimate's structure. Its structure is hierarchical – with answers constructed from solving smaller problems, which might be constructed from even solving still smaller problems – and its most compact representation is as a tree.

As an example, let's construct the tree representing the elaborate divide-and-conquer estimate for a CDROM's pit spacing (**Section 2.1**). The tree's root is 'capacity, area', a two-word tag reminding us of the method underlying the estimate. The estimate dissolves into finding two quantities – the capacity and area – so the tree's root sprouts two branches.



Of the two new leaves, the area is easy to estimate without explicitly subdividing into smaller problems, so the 'area' node remains a leaf. To estimate the capacity – rather, to estimate the capacity reliably – we used intelligent redundancy: (1) looking on a CDROM box; and (2) estimating how many bits are required to represent the music that fits on an audio CD. The second method subdivided into three estimates: for the playing time, sample rate, and sample size. Accordingly, the 'capacity' node sprouts new branches – and a new connector:



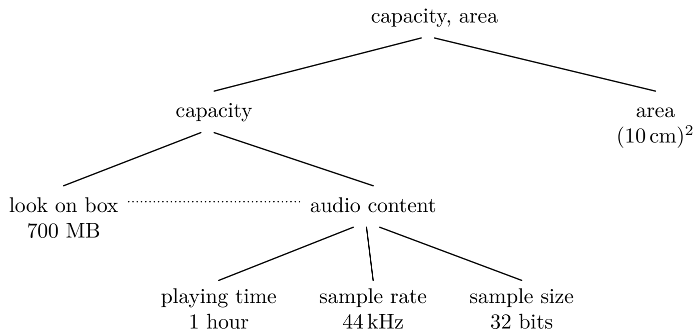
The dotted horizontal line indicates that its endpoints redundantly evaluate their common parent (see **Section 2.2**). Just as a crossbar strengthens a structure, the crossing line indicates the extra reliability of an estimate based on redundant methods.

The next step in representing the estimate is to include estimates at the five leaves:

1. capacity on a box of CDROM's: 700 MB;
2. playing time: roughly one hour;
3. sampling rate: 44 kHz;

4. sample size: 32 bits;
5. area: $(10\,\text{cm})^2$.

Here is the quantified tree:



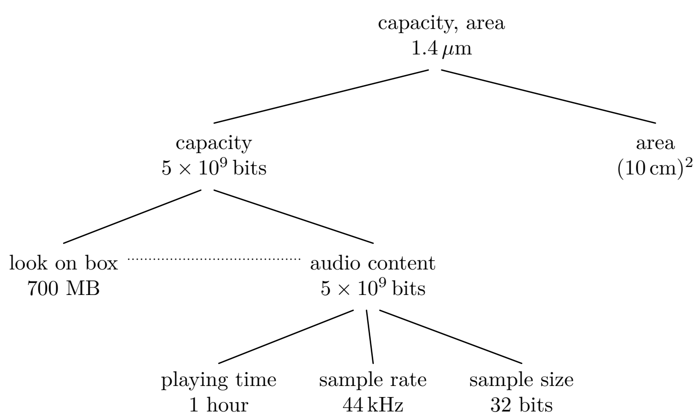
The final step is to propagate estimates upward, from children to parent, until reaching the root.

Draw the resulting tree.

Here are estimates for the nonleaf nodes:

1. *audio content.* It is the product of playing time, sample rate, and sample size: $5 \cdot 10^9$ bits.
2. *capacity.* The look-on-box and audio-content methods agree on the capacity: $5 \cdot 10^9$ bits.
3. *pit spacing computed from capacity and area.* At last, the root node! The pit spacing is $\sqrt{A/N}$, where A is the area and N is the capacity. The spacing, using that formula, is roughly $1.4\,\mu$.

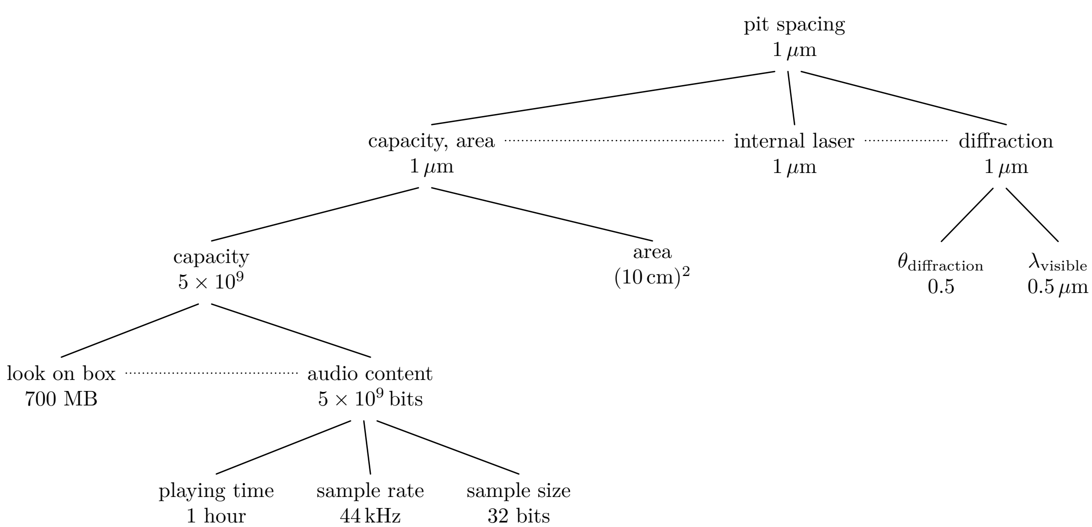
Propagating estimates from leaf to root gives the following tree:



This tree is far more compact than the sentences, equations, and paragraphs of the original analysis in **Section 2.1**. The comparison becomes even stronger by including the alternative estimation methods in **Section 2.2**: (1) the wavelength of the internal laser, and (2) diffraction to explain the shimmering colors of a CD.

▶ *Draw a tree that includes these methods.*

The wavelength method depends on just quantity, the wavelength of the laser, so its tree has just that one node. The diffraction method depends on two quantities, the diffraction angle and the wavelength of visible light, so its tree has those two nodes as children. All three trees combine into a larger tree that represents the entire analysis:



This tree summarizes the whole analysis of **Section 2.1** and **Section 2.2** – in one figure. The compact representation make it possible to grasp the analysis in one glance. It makes the whole analysis easier to understand, evaluate, and perhaps improve.

2.4 Example 2: Oil imports

For practice, here is a divide-and-conquer estimate using trees throughout:

▶ *How much oil does the United States import (in barrels per year)?*

One method is to subdivide the problem into three quantities: