Chapter 5 gives a very basic introduction to the ideas underlying algorithm design and analysis. The author discusses such riveting, novel topics as merge sort, the fact that $\mathcal{O}\left(n^2\right)$ is worse than $\mathcal{O}\left(n \lg n\right)$, and the concept of a greedy heuristic. Sorry if I sound overly contemptuous here, but as someone well-versed in at least the fundamentals of computer science, this chapter was particularly uninformative.

Chapter 6 provides some ideas about information theory, by way of data compression and encryption. We begin with a simple definition of information and a discussion of text compression methods that target this definition by compressing areas of data regularity in the text. We then move on to other forms of data such as images and digitized analog signals, including the use of lossy compression. The discussion of image compression leads to the classic philosophical conundrum that occurs when one tries to define information as the amount of irregularity in some data—namely that this is incongruent with how most people intuit and reason about information. This leads to a new definition referring to the size of a computer program required to generate the information. We then get two brief sections covering the basics of public-key encryption, and the use of parity bits for error-checking and -correcting, respectively.

While I of course appreciate the importance of the ideas presented in chapter 5, there really is nothing new there, at least for me. I've encountered literally all of this in previous Mines courses, especially *Algorithms* and *Discrete Mathematics*. The only interesting idea I got from this chapter is to start bringing in a deck of cards to my office hours for when I need to teach students the intuition behind merge sort (this is, admittedly, a pretty fun idea). It also contained my favorite quote from the book so far: "If you don't remember what logarithms are, never mind. They are all small numbers, so they can be safely ignored."

I'm also familiar with the topics from chapter 6, but I happen to be quite fond of them, so I always enjoy reiteration. It's fun to see that the author apparently attempted to compress the book itself using the methods described, or at least did enough calculations to get approximate values. I also am able to appreciate the final definition of information they landed on more than when I've seen it in the past, now that I've learned the basics of computation theory. Understanding that all computers really are identical in their capabilities gives this definition more credence, though of course philosophers are never satisfied and so the debate rages on.