## Summary

This chapter examines some of the pillars of theoretical computer science, including runtime complexity analysis, computability, algorithm design, and algorithm optimization. As you have seen, algorithm optimization is not restricted to the ivory tower of academia; there are real-life situations where choosing the proper algorithm or using a compression technique can make a considerable performance difference. Specifically, the sections on dynamic programming, index storage, and approximation algorithms contain recipes that you can adapt for your own applications.

The examples in this chapter are not even the tip of the iceberg of complexity theory and algorithms research. Our primary hope is that, by reading this chapter, you learned to appreciate some of the ideas behind theoretical computer science and practical algorithms used by real applications. We know that many .NET developers rarely encounter the need to invent a completely new algorithm, but we still think it is important to understand the taxonomy of the field and have some examples of algorithms on your tool belt.