Module Six

Learning Objectives

By the end of this module, you will meet these learning objectives:

- Explain the complexity of an algorithm
- Analyze the runtime requirements of an algorithm
- Analyze the memory requirements of an algorithm
- Analyze the amount of memory that is needed to store data for a given algorithm
- Analyze the input, output, and data structures needed in programming applications

Module Overview

Welcome to Module Six! When analyzing algorithms, we are not typically concerned with the actual running time since that is based on the computer on which the code is running, the operating system, the number of other applications that are running, the number of CPUs, the amount of memory, other hardware configuration, and so on. Instead, we are interested in how the running time changes as the input size changes.

For example, if an algorithm that sorts a set of input values takes 10ms to run when the input size is 1,000, how long will it take when the input size is 2,000? Put more generally, if the input size is n, how long will it take to run (in terms of n)? That is called the "algorithmic running time." We analyze the running time of algorithms using a worst-case bound, which is called Big-O notation. Going back to our sorting example, if the sorting algorithm has an algorithmic running time of O(n2), that means that the actual running time as the input size changes from 1,000 to 2,000 (which would be changing from n=1,000 to 2n=2,000) would be O((2n)2) = O(4n2). As you see, when the input size doubles, the running time quadruples!

We often want to get algorithms to run in polynomial or sub-polynomial time, meaning that we don't want an algorithm to run slower than O(nk) where k is a constant value. **Note:** Even though O(n100) is considered polynomial time, it is a very slow algorithm. However, as the value of n>100, O(nn) will be larger, which makes for a terribly slow running time.

Big-O notation is the most common running time analysis, and is used for determining the worst-case running time. Big-O notation will give us an upper bound on the algorithm, but there is also Big-Omega (best-case running time) and Big-Theta (average-case running time).

Module at a Glance

This is the recommended plan for completing the reading assignments and activities within the module. Additional information can be found in the module Resources section and on the module table of contents page.

1. Review the Module Six resources.
2. Complete the Module Six activities in zyBooks.

3 Submit Project One.