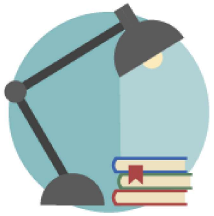
TidBIT

Did you know that the “O” in Big-O stands for “order”? When an algorithm runs in Big-O time, we are saying that it is “on the order of” a function. In other words, we don’t care about the actual running time but just a function that will approximate the running time. Because of that, we ignore constants. O(3n) = O(n) or, to take it to an extreme, O(100n) = O(n). Obviously the actual running time will be different, but the algorithmic running times are the same.



Required Resources

Textbook: *Data Structures and Algorithms*, Chapter 3 (/d2l/common/dialogs/quickLink/quickLink.d2l?ou=1860222&type=lti&rcode=snhu-2534460&srcou=1040994) (Sections 3.7, 3.11, 3.12, and 3.13)

This zyBooks reading will provide you with information on the following topics. Note that you have read these sections previously.

- Constant time operations, growth of functions and complexity, Big-O notation, algorithm analysis, and analyzing the time complexity of recursive algorithms