Worksheet 10: Using Big-Oh to Estimate Wall Clock Time

In preparation: Read Chapter 4 to learn more about the concept of big-oh notation.

As you learned in Chapter 4, a big-Oh description of an algorithm is a characterization of the change in execution time as the input size changes. If you have actual execution timings ("wall clock time") for an algorithm with one input size, you can use the big-Oh to estimate the execution time for a different input size. The fundamental equation says that the ratio of the big-Oh's is equal to the ratio of the execution times. If an algorithm is O(f(n)), and you know that on input n_1 it takes time t_1, and you want to find the time t_2 it will take to process an input of size n_2, you create the equation

$$f(n_1) / f(n_2) = t_1 / t_2$$

To illustrate, suppose you want to actually perform the mind experiment from Lesson 7. You ask a friend to search for the phone number for "Chris Smith" in 8 pages of a phone book. Your friend does this in 10 seconds. From this, you can estimate how long it would take to search a 256 page phone book. Remembering that binary search is O(log n), you set up the following equation:

$$\log(8)/\log(256), \text{ which is } 3 / 8 = 10 / X$$

Solving for X gives you the answer that your friend should be able to find the number in about 24 seconds. Now you time your friend perform a search for the name attached to a given number in the same 8 pages. This time your friend takes 2 minutes. Recalling that a linear search is O(n), this tells you that to search a 256 page phone could would require:

$$8/256 = 2 / X$$

Solving for X tells you that your friend would need about 64 minutes, or about an hour. So a binary search is really faster than a linear search.

Linear search	O(n)
Binary search	O(log n)
countOccurrences	O(n)
isPrime	O(Sqrt(n))
printPrimes	O(nSqrt(n))
matMult	O(n^3)
SelectionSort	O(n^2)

Recall from the last lesson the table of execution times for common algorithms. Using these, estimate the actual execution times for various tasks in the following problems.

Worksheet 10: Using Big-Oh to Estimate Wall Clock Time Name:

Problem 1. Warehouse video keeps a list of titles for their 7000 item inventory in a simple unsorted list. To find out how many copies of “Kill Bill” they have using the countOccurrences algorithm takes about 45 seconds. They recently acquired a competing video store, and now their inventory has 43000 items. How long will it take to search?

Problem 2. Suppose you can multiply two 17 by 17 matrices in 33 seconds. How long will it take to multiply two 51 by 51 element matrices. (By the way 51 is 17 times 3).

Problem 3. If you can print all the primes between 2 and 10000 in 92 seconds, how long will it take to print all the primes between 2 and 160000?