CSCI-2270
Recitation 3
Instructor: Boese

Algorithm Analysis

Objectives

- Reinforce your understanding of algorithm analysis

Write down your answers (or print this out and bring to lab to fill out).
The TA will walk you through a discussion of each.

True/False

1. True or False – In terms of efficiency, all algorithms are equivalent.

2. True or False – Algorithm analysis uses a mathematical notation called Big-O to analyze any given algorithm.

3. True or False – Big-O is really the Greek symbol Theta.

4. True or False – What really determines how efficient an algorithm is is the processor speed of the computer on which it is running.

5. True or False – Given a SEQUENTIAL algorithm (no branches or loops) with ten lines of code, $\Theta = 10c$, where c is some machine dependent constant we won't care about.

Calculations

6. If you had an algorithm (developed by some twisted person!) that included THREE nested loops, what would the algorithm complexity be?
    a. 3n
    b. n^3
    c. 3n + 30
    d. There is no way of knowing

7. One of the most frequently deployed algorithms in Computer Science applications is a Search. Which search algorithm looks at each element in a list until it finds a match? For example, you have a key chain with 50 keys and you need to find a specific key to unlock a door…
    a. Binary
    b. Random
    c. Sequential
    d. Quick

8. On average (ie, you run your key search 100 times), how long will it take you to locate a match using a Sequential Search?
    a. 1 time
    b. n times
    c. 2n times
    d. n/2 times

9. What is the worst case scenario on how long it will take you to locate a match using a Sequential Search?

n times

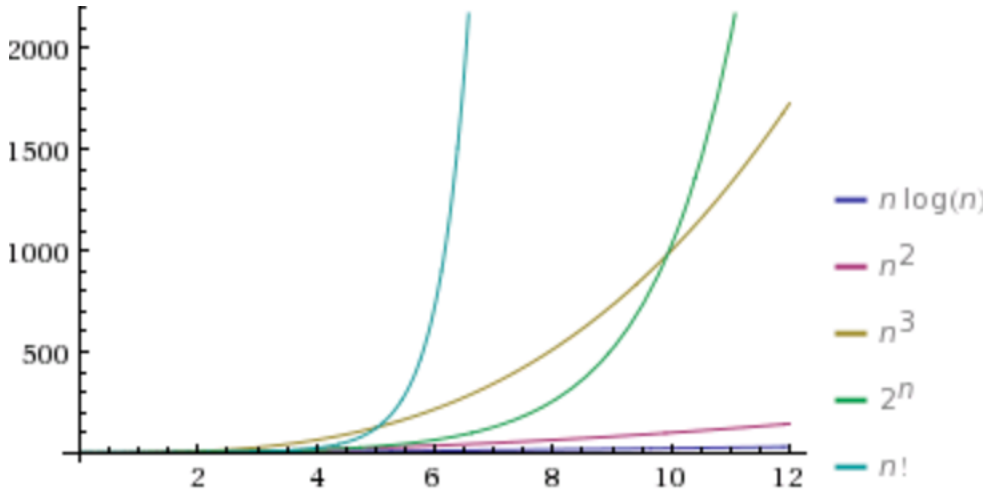
10. What are the pre-conditions of when you can use a Binary Search on a list?

has to be an ordered list, however your data structure has to support theta 1 data access

11. Why do we drop the constants from Big-O notation?

because they don't matter

12. What happens when the Big-O of an algorithm is more efficient for smaller values of n but less efficient for large values of n? *(e.g.,* $O(2^n)$ *vs* $O(n^3)$ *)*



Big o is important for larger values of n

To get credit for this lab exercise, show the TA and sign the lab's completion log.