Algorithms Lab 1

The in-lab problems are to be solved during the lab time. Work with your team, but write your solutions individually. You do not need to turn in your answers, only discuss them with me. Ask me for feedback on how you write your answers as well.

The homework problem set is due on Wed 9/17. Work with your team, but write your solutions individually. List the people with whom you discussed the problems.

1 In lab exercises

1. Consider the problem of finding the index of the smallest element in an array of n elements. We discussed an algorithm for solving this problem in class. Express, as a function of n, the best-case and worst-case running time of this algorithm.

2. Express, as a function of n, the best-case and worst-case running time of selection sort. Use the answer from (1). It's ok to leave the answer as a summation.

3. Algorithm A uses $10n \lg n$ operations, while algorithm B uses n^2 operations. Determine the value n_0 such that A is better than B for $n \geq n_0$.

4. Let $f(n) = \lg n$ and assume that we have an algorithm whose running time is $f(n)$ microseconds. Determine the largest size of a problem that can be solved by the algorithm in: (a) 1 second; (b) 1 hour; (c) 1 month; (d) 1 century.

   Same problem for $f(n) = \sqrt{n}$ and $f(n) = 2^n$.

2 Homework problems

1. (GT C-1.11) Describe a method for finding both the minimum and the maximum of n numbers with fewer than $3n/2$ comparisons.

2. Arrange the following functions in ascending order of growth rate. For each pair of consecutive functions, give a brief justification on why they are in this order. For e.g., if you ordered A, B, C, you need to justify that 1. $A = O(B)$; and 2. $B = O(C)$.

$$2^{\sqrt{\log n}}, 2^n, n^{4/3}, n(\log n)^3, n^{\log n}, 2^{2^n}, 2^{n^2}$$

3. (GT C-1.9) Give an example of a positive function $f(n)$ such that $f(n)$ is neither $O(n)$ nor $\Omega(n)$.