CS 410 Summer 2000
Homework 1
Due 11:30 AM, Monday July 3

Important notes about all exercises for this course:

- No late homework will be accepted.
- Justify your answers. Answers without brief (but adequate) justification may be considered incomplete.
- You may use pseudocode, Java, C, or C++ to write your algorithms. You do not need to compile or run your code for written exercises.
- When presenting algorithms, also provide a brief English description of the pseudocode and an explanation of why the algorithm (and your analysis of it) is correct.

Reading

Read CLR chapters 1, 2, 3, 4.1, 4.2, 4.3, 7, 8.

Written Exercises

1. CLR 1.2-1. !!!Some additional instructions here. Then answer the following questions:

   (a) Compare selection sort with bubble sort? How are they different? How are they similar? Give a benefit of each.

   (b) How does the asymptotic growth of selection sort compare with that of bubble sort?

   (c) How would you modify selection sort to use only one (the original) array?

2. CLR 1.4-2

3. Rank the following functions by order of growth. That is, find an arrangement $f_1, f_2, \ldots, f_n$ of the functions satisfying $f_1 = O(f_2), f_2 = O(f_3)$, etc. In each case indicate whether $f_i = \Theta(f_{i+1})$

   $n,\ \sqrt{n},\ \lg n,\ \lg(\lg n),\ n/(\lg n),\ 10^{\sqrt{n}},\ e^n,\ 2^n,\ n^2,\ \lg(n^2),\ n^2 + n \lg n.$

   You do **not** need to show any proof for this question.

4. Let $f(n)$ and $g(n)$ be asymptotically positive functions. Prove or disprove each of the following conjectures. Note that to disprove a conjecture you only have to produce a single counterexample.

   (a) $f(n) = O(g(n))$ implies $g(n) = O(f(n))$.

   (b) $f(n) + g(n) = O(min(f(n), g(n)))$.

   (c) $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

   (d) $f(n) = O(g(n))$ implies $2f(n) = O(2g(n))$.

   (e) $f(n) = O(f(n)/2)$.

   (f) $f(n) = O(f(n/2))$.

   (g) $2^{n+1} = O(2^n)$.

   (h) $2^{2n} = O(2^n)$.

5. You might not think the difference between n^2 and $n \lg n$ is such a big deal, but consider this. The human genome is a sequence of roughly 3.5 billion nucleotides, each of which is one of four types A,C,G,T. Representing each of A,C,G,T as a string of 2 bits, the entire genome string would take up roughly .875 gigabytes of storage, thus would fit comfortably on a laptop hard disk.

   (a) Suppose your computer can perform 10^9 operations per second. Say you wanted to analyze the genome running a naive string matching algorithm known to take n^2 operations on inputs of length n. Comment on the feasibility of this.

   (b) Suppose that by using a more appropriate data structure, your algorithm could perform the same task using only $n \lg n$ operations on inputs of length n. What then?

6. (a) Show that $o(f(n)) \subset O(f(n))$.

   (b) Show that for real numbers a and b with $a > 1$, $n^b = o(a^n)$.

7. CLR 3.2-4

8. Assume $a_1 + a_2 < 1$. Prove $\mathrm{T}(n) = \mathrm{T}(a_1 n) + \mathrm{T}(a_2 n) + bn$ is $O(n)$.
   (Hint: Use induction and express c in terms of a_1, a_2, and b. Assume $\mathrm{T}(1) = 1$. Where do you use the fact that $a_1 + a_2 < 1$?)

9. CLR 4.2-3

10. CLR 4.3-2

11. Let T(n) = T(n-1) + T(n-2) (the Fibonacci sequence).
    Prove that it is $\Theta(\varphi^n)$, where φ, the golden mean, is $\frac{1+\sqrt{5}}{2}$.