CS 381, Fall 2014; Instructor: Elena Grigorescu
TAs: Nabeel, Akash
Due September 12, 12:01 pm sharp, in hard copy.

Problem set 1

You may discuss the problem sets with other students in the class, however you *must* write up the solutions yourself. If you collaborate, specify who you worked with. No other solution sources are allowed. The solutions may be hand-written.

1 Bookeeping

1. Read the chapters from CLRS for the material covered specified on the class website.
2. Sign up on Piazza (see link on class website)
3. Submit a recent photo of yourself to the TAs, as a lastname.jpg file. I might use this photo to remember you in the future if you need a reference letter.

2 Solve but do not turn in (they might appear in exams/quizzes)

1. Recall the insertion sort algorithm from class (book). If the input array looks like

$$\langle 3, 2, 1, 6, 5, 4, 9, 8, 7, ..., 3n, 3n-1, 3n-2 \rangle,$$

   what is the running time of the algorithm?

2. What is the running time of merge sort on an already sorted array? What is the runtime on an inversely sorted array?

3 Solve and turn in

Problem 1 Rank the following functions by increasing order of growth (i.e., the slowest-growing first, the fastest-growing last):

$(\log n)^6$, $n!$, $\sqrt{n}$, n^{21}, $n(\log n)^4$, $n \log n$, 2^n, n^2, $\log n^n$, $(\log n)^{0.3}$

where all the logarithms are to the base 2. If two functions have equal orders of growth then list them grouped together, e.g., between brackets {like this}.

Problem 2

Consider the recurrence $T(n) = 8T(n/2) + n^2$ and $T(1) = 1$.

1. Use the recursion tree method to 'guess' the solution to the recurrence relation. Does your quess coincide with the Master theorem solution?
2. Use induction to give an alternative proof of the solution you found in the previous step.

Problem 3 Alice needs to climb n stairs to get to her room. In a single step Alice can climb 1 stair or 2 stairs. What is the number of ways in which Alice can climb the stairs if, say n = 12? Two walk sequences are considered different if the sequence of steps taken is not identical. (Hint: Do you recognize the sequence of integers described by the resulting recurrence?)

Problem 4

Given an array of n integers and a target integer D, describe the best (in terms of running time) algorithm you can, that finds two elements of the array whose difference is D or outputs that no such elements exist.

Problem 5

1. Describe an algorithm that given an array of n integers, finds the 3rd smallest one in $O(n)$ time.
2. Describe an algorithm that given an array of n integers, finds the $\sqrt{n}$th smallest one in $O(n \log n)$ time.
3. (Think about it, but do not turn in. It's a harder problem.) Can you achieve an asymptotically better running time than $n \log n$ for finding the $\sqrt{n}$th smallest element?