Problem Set 1

This problem set is due **at 10:00pm** on **Wednesday, September 12, 2018.**

Please make note of the following instructions:

- This assignment, like later assignments, consists of *exercises* and *problems*. **Hand in solutions to the problems only.** However, we strongly advise that you work out the exercises for yourself, since they will help you learn the course material. You are responsible for the material they cover.
- Remember that the problem set must be submitted on Gradescope. If you haven't done so already, please signup for 6.046 Fall 2018 on Gradescope, with the entry code MG2K5P, to submit this assignment.
- We require that the solution to the problems is submitted as a PDF file, **typeset on LaTeX**, using the template available in the course materials. Each submitted solution should start with your name, the course number, the problem number, your recitation section, the date, and the names of any students with whom you collaborated.
- We will often ask you to "give an algorithm" to solve a problem. Your write-up should take the form of a short essay. Start by defining the problem you are solving and stating what your results are. Then provide: (a) a description of the algorithm in English and, if helpful, pseudo-code; (b) a proof (or proof sketch) for the correctness of the algorithm; and (c) an analysis of the running time. We will give full credit only for correct solutions that are described clearly and convincingly.

EXERCISES (NOT TO BE TURNED IN)

Asymptotic Analysis, Recursion, and Master Theorem

- Do Exercise 4.3-7 in CLRS on page 87.
- Do Exercise 4.3-9 in CLRS on page 88.

Divide and Conquer Algorithms

- Do Exercise 4.2-3 in CLRS on page 82.
- Do Exercise 9.3-1 in CLRS on page 223.

Problem 1-1. Recurrences and Asymptotics [40 points]

Let $T(n)$ be the time complexity of an algorithm to solve a problem of size n. Assume $T(n)$ is $O(1)$ for any n less than 3. Solve the following recurrence relations for $T(n)$. Please show your work in deriving your solutions.

(a) [5 points] $T(n) = 5T\left(\frac{n}{4}\right) + n$

(b) [5 points] $T(n) = 9T\left(\frac{n}{3}\right) + n^2$

(c) [5 points] $T(n) = T(n-2) + n$

(d) [5 points] $T(n) = 3T(\sqrt[2]{n}) + \log n$.

(e) [5 points] $T(n) = 2T(n/4) + T(n/3) + \Theta(n)$.

(f) [15 points] Consider the following functions. Within each group, sort the functions in asymptotically increasing order, showing strict orderings as necessary. For example, we may sort $n^3, n, 2n$ as $2n = O(n) = o(n^3)$.

1. [5 points] $\log n$, $\sqrt{n}$, $\log n^2$, $\log_5 n$, $\log^2 n$, $\log \log n$.
2. [5 points] $n^{4/3}$, $n \log n$, n^2, 2^n, $\log(n!)$.
3. [5 points] n^n, n^{n-1}, e^n, $2^{\log n^{\log \log n}}$.

Problem 1-2. Ben Bitdiddle vs. Xers [60 points] When a group of humans from Earth-X (the Earth of a parallel universe) find themselves on the Earth of our universe, they assimilate immediately into the incoming MIT Class of 2050. Unfortunately, scientists discover that their presence on Earth is causing our universe to become unstable. Ben Bitdiddle, an intern at NASA, is charged with identifying the Earth-X humans (or Xers), and returning them to their universe. Ben knows that over half of the students in the MIT Class of 2050 are from Earth-X. As his first order of business, he gathers the entire incoming class in a room together.

(a) [15 points] Formally, there are n people in a room. We know that more than $n/2$ of them are from Earth-X, and the rest are from Earth. However, the only way to get any information about who is from where, is by asking someone the following question: "Are person A and person B from the same universe?". One of the people in the question (Person A or Person B) can be the person who is answering the question.

Assuming everyone is honest, design an algorithm to identify someone from Earth-X.

For full credit, your algorithm should run in $O(n)$ time, assuming that each query takes $O(1)$ time.

(b) [10 points] Ben tries out the algorithm from part (a), but quickly realizes that there is a problem. The Xers don't want to leave, so they choose not to answer the questions honestly. That is, a person from Earth will always answer Ben's questions honestly, while a person from Earth-X may choose to lie or tell the truth. You may assume that the Xers can cooperate to fool Ben.

Ben now questions two students at a time, and asks them what planet the other student is from. For example, in a single query, Ben can ask a pair of students Student A and Student B the following:

To student A: "What planet is student B from?"

To student B: "What planet is student A from?"

Show that it's impossible for Ben to identify the set of students from Earth.

(c) [10 points] Assume that Ben asks a single query of a pair of students (A, B), as described in part (b). For each possible response, describe whether the pair (A, B) must contain 0, 1, or 2 Xers.

Note that, from a single query, it might not be possible to figure out exactly how many Xers are in the pair. However, Ben might learn that there must be either 0 or 1 Xer in the pair. For full credit, all responses must be justified.

(d) [25 points] Luckily for Ben, his manager at NASA comes by and identifies a few Xers for him, and removes them from the room. Now, less than half of the remaining students are from Planet-X. Design a divide-and-conquer algorithm for Ben to find all the Xers, in $O(n)$ time. For full credit, your solution must include a clear and rigorous proof of correctness and runtime analysis.

Hint: Keep in mind that the problem is not solvable when more than half the people in the group are Xers - consider making this an invariant, for all the subproblems, as well.