CS7800: Advanced Algorithms **Fall 2025**
Homework 1 **Due: 09/12/25, 11:59pm**

Jonathan Ullman

Philosophy/Policies

This is a PhD-level core class, and I think of it as having two goals: (1) Ensure an appropriate working knowledge of algorithms and algorithmic reasoning for a PhD computer scientist. (2) Give you a "mental workout" by exposing you to new skills and challenging problems. These assignments and policies are structured to balance these two goals. Towards this end, assignments have two types of problems:

- *Assigned problems* for you to complete and submit by the due date (unless using late days). These will be graded and checked for consistency with the class honesty policy. I will provide full solutions for these problems.
- *Suggested problems* that you do not need to complete, but will be useful for guiding your practice and stretching your ability. You may work on these problems any way you like, with no honesty policy. I can't promise full solutions for all of these problems but am happy to discuss them in office hours.

Unless I indicate otherwise, both assigned and suggested problems are related to the lecture material and are considered "fair game" for exams.

All solutions must be typed in LaTeX! You'll need to learn LaTeXeventually to write papers and there are many resources to get started. I'd suggest starting with the source code for this assignment.

This is a PhD-level course and I trust that you will figure out how to use these assignments to perform well on the exams and get what you need out of the course. Thus, I have kept the honesty policies as lightweight as possible.

- *Collaboration is allowed and encouraged!* To make sure you are contributing, limit collaboration to groups of at most three. Identify your collaborators on your solutions.
- *You're adults and scholars, so act like it!* It would be easy to solve these problems using AI or other sources, but you wouldn't learn anything or exercise your brain, and it would waste my time giving feedback on solutions you didn't write. However, policing AI-based cheating is time consuming at best, and hopeless at worst, so I simply won't bother. I ask that you respect me and yourself enough to do the work yourself.

Assigned Problems

Problem 1 This question will test your comfort with asymptotic notation, which we lean on when analyzing the running time of algorithms. If you need to review this material, I recommend KT Chapter 2.

Put the following functions in asymptotic order from smallest to largest by asymptotic ("big O") notation and labels which asymptotic relationships are strict ("little o"). More specifically, given the eight functions below, put them in order $f_1 \preceq f_2 \preceq \cdots \preceq f_8$ so that $f_i = O(f_{i+1})$ for each i and indicate whether the pair also satisfies $f_i = o(f_{i+1})$.

$$\sum_{i=1}^{n} 2^i \qquad \sum_{i=1}^{n} i \qquad 2.2^n \qquad \log_2^2 n \qquad n \qquad \log_2(n^3) \qquad \sum_{i=1}^{n} \frac{1}{i} \qquad 5^{\log_4 n}$$

Problem 2 These questions will test your comfort with some common proof strategies we will use: proof-by-induction and proof-by-contradiction. If you need to review this material, I recommend Appendix I in Erickson.[1]

2.1 Prove by contradiction that any *tree*[2] $G = (V, E)$ contains at least two *leaves*.[3]

2.2 Prove by induction that for every real number $x \geq 0$ and every integer $n \geq 0$, $(1 + x)^n \geq 1 + nx$.

2.3 Your friend tries to convince you of the following obviously erroneous theorem, and gives the following proof by induction.

Theorem 1. *In every set of $n \geq 1$ students, every student uses the same chatbot to do their homework.*

Proof. Let $H(n)$ be the statement "In every set of $1 \leq n \leq k$ students, every student uses the same chatbot to do their homework" We will prove that $H(n)$ is true for every n. For the base case $H(1)$ is true since any single student uses the same chatbot as itself. For the inductive step, we will prove that $H(n-1) \implies H(n)$ for every $n \geq 2$. Consider any set of n students. By the inductive hypothesis $H(n-1)$, the first $n-1$ students use the same chatbot

$$\underbrace{s_1 \;\; s_2 \;\; s_3 \;\; \ldots \;\; s_{n-1}}_{\text{same chatbot}} \;\; s_n$$

Also by the inductive hypothesis, the last $n-1$ students use the same chatbot

$$s_1 \;\; \underbrace{s_2 \;\; s_3 \;\; \ldots \;\; s_{n-1} \;\; s_n}_{\text{same chatbot}}$$

By transitivity, all n of the students use the smae chatbot. The proof of the theorem now follows by induction. □

Since the statement is obviously false, there must be a specific flaw in the proof. What is it?

[1] https://jeffe.cs.illinois.edu/teaching/algorithms/
[2] A *tree* is an undirected graph that is *connected* and *acyclic*.
[3] A *leaf* in a tree is a node with exactly one neighbor.

Problem 3 This question will test your knowledge of undergraduate-level algorithms and algorithmic concepts, as well as your ability to write algorithms in pseudocode and to analyze those algorithms' properties. If you need to review this material, I suggest reading KT Chapters 2, 3, and 5.

You are looking for an apartment and you'd like to find one that is in good condition. Unfortunately, Boston apartments are old, and landlords play lots of tricks with their photographs, so you won't know the condition of an apartment until you go see it. Since you are so busy with algorithms homework, you won't have time to visit every apartment, but you decide that you'll be satisfied as long as you can find an apartment that is in better condition than those of your neighbors. Fortunately, we will show you that you only need to visit a very small fraction of the apartments in Boston to find a place to live.

More precisely, there are n apartments on a single street, and there are numbers $c_1, \ldots, c_n$ that represent the condition of the apartment. Your goal is to find an apartment that is in better condition than your neighbors, meaning apartment i such that $c_i \geq c_{i-1}$ and $c_i \geq c_{i+1}$. If $i = 1$ then it's enough to have $c_1 \geq c_2$ and if $i = n$ then it's enough to have $c_n \geq c_{n-1}$.

Design and analyze an algorithm that finds an apartment that is in better condition than its neighbors, and visits only $O(\log n)$ apartments.[4] Your solution should include

- A clear *pseudocode* description of your algorithm.
- A clear argument of its correctness. Your argument doesn't need to be overly formal, as long as it's logical and convincing.
- An analysis of the asymptotic ("big O") worst-case number of apartments visited.

Problem 4 This series of questions will test your comfort with discrete probability, which will be important later in the course when we study randomized algorithms. If you need to review this material, I suggest reading the Discrete Probability chapter of Erickson.

A fair die with n sides has the numbers $1, 2, \ldots, n$ written on it, and when you roll it each of the numbers comes up with equal probability $1/n$. Suppose you have a set of six fair dice with 4, 6, 8, 10, 12, and 20 sides. You roll each of the six dice, and define the random variables $X_4, X_6, X_8, X_{10}, X_{12}$, and X_{20} to be the value that comes up on each of the dice. Calculate the following expected values:

4.1 $\mathbb{E}(X_4)$

4.2 $\mathbb{E}(X_4 + X_6 + X_8 + X_{10} + X_{12} + X_{20})$

4.3 $\mathbb{E}(X_4^2)$

4.4 $\mathbb{E}(X_8^2 \mid X_4 + X_8 = 7)$

4.5 $\mathbb{E}(X_4 X_6 \mid X_4 + X_6 = 5)$

4.6 $\mathbb{E}(X_{10} X_{12} X_{20} \mid X_4 + X_6 = 3)$

[4] **Hint:** Think about binary search!