CS4810 / CS7800: Advanced Algorithms — Fall 2022
Homework 1: Due Friday, September 16, 2022

Mahsa Derakhshan and Jonathan Ullman

Assigned Problems

Problem 1 This question will test your comfort with asymptotic notation, which we use heavily when analyzing properties 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 $f_i = o(f_{i+1})$.

$$2^{\log_3 n} \qquad 3^{\log_2 n} \qquad \sum_{i=1}^{n} i^2 \qquad \sum_{i=1}^{n} i \qquad 1.1^n \qquad \log_2^3 n \qquad n \qquad 2n^2$$

Problem 2 These questions will test your comfort with 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, all students have the same favorite song.*

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

$$\underbrace{s_1 \;\; s_2 \;\; s_3 \;\; \ldots \;\; s_{k-1}}_{\text{same favorite song}} \;\; s_k$$

Also by the inductive hypothesis, the last $k-1$ students have the same favorite song.

$$s_1 \;\; \underbrace{s_2 \;\; s_3 \;\; \ldots \;\; s_{k-1} \;\; s_k}_{\text{same favorite song}}$$

[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.

By transitivity, all k of the students must have the same favorite song. 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?

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 clear 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.

4.1 Suppose you have a set of six fair dice with 4, 6, 8, 10, 12, and 20 sides where a die with n sides is numbered $1, 2, \ldots, n$. 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:

(a) $\mathbb{E}(X_4)$
(b) $\mathbb{E}(X_4 + X_6 + X_8 + X_{10} + X_{12} + X_{20})$
(c) $\mathbb{E}(X_4^2)$
(d) $\mathbb{E}(X_8^2 \mid X_4 + X_8 = 7)$
(e) $\mathbb{E}(X_4 X_6 \mid X_4 + X_6 = 5)$
(f) $\mathbb{E}(X_{10} X_{12} X_{20})$

4.2 Suppose you have a fair coin that comes up heads or tails. You first toss the coin until you get heads. Let X be a random variable for the number of times you got tails before you get heads. Then

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

you toss the coin again until you see heads X times. For example, if you start out with H you stop, and if you start out with TTH then you flip again until you see heads twice. Let Y be a random variable for the total number of times you flip the coin. Calculate $\mathbb{E}(Y)$.