CS 70 Discrete Mathematics and Probability Theory
Fall 2021

HW 2

Due: Friday 9/10, 10:00 PM
Grace period until Friday 9/10, 11:59 PM

Sundry

Before you start writing your final homework submission, state briefly how you worked on it. Who else did you work with? List names and email addresses. (In case of homework party, you can just describe the group.)

1 True or False?

For each of the questions, answer TRUE or FALSE and justify your answers.

(a) Suppose you proved the inductive step for a statement $P(n)$ but then discovered that $P(29)$ is false. Thus, $P(1)$ has to be false.

(b) Suppose you proved the inductive step for a statement $P(n)$ but then discovered that $P(29)$ is false. Then, we cannot say anything about P(50).

(c) In a stable matching instance where there is a job at the bottom of each candidate's preference list, the job is paired with its least favorite candidate in every stable pairing.

(d) In a stable matching instance where there is a job at the top of each candidate's preference list, the job is paired with its favorite candidate in every stable pairing.

2 Airport

Suppose that there are $2n+1$ airports where n is a positive integer. The distances between any two airports are all different. For each airport, exactly one airplane departs from it and is destined for the closest airport. Prove by induction that there is an airport which has no airplanes destined for it.

3 Proving Inequality

For all positive integers $n \geq 1$, prove that

$$\frac{1}{3^1} + \frac{1}{3^2} + \ldots + \frac{1}{3^n} < \frac{1}{2}.$$

(Note: while you can use formula for an infinite geometric series to prove this, we would like you to use induction. If you're having trouble with the inductive step, try strengthening the inductive hypothesis. Can you prove an equality statement instead of an inequality?)

4 Binary Numbers

Prove that every positive integer n can be written in binary. In other words, prove that we can write

$$n = c_k \cdot 2^k + c_{k-1} \cdot 2^{k-1} + \cdots + c_1 \cdot 2^1 + c_0 \cdot 2^0,$$

where $k \in \mathbb{N}$ and $c_i \in \{0,1\}$ for all $i \leq k$.

5 AM-GM

For nonnegative real numbers $a_1, \cdots, a_n$, the arithmetic mean, or average, is defined by

$$\frac{a_1 + \cdots + a_n}{n},$$

and the geometric mean is defined by

$$\sqrt[n]{a_1 \cdots a_n}.$$

In this problem, we will prove the "AM-GM" inequality. More precisely, for all positive integers $n \geq 2$, given any nonnegative real numbers $a_1, \cdots, a_n$, we will show that

$$\frac{a_1 + \cdots + a_n}{n} \geq \sqrt[n]{a_1 \cdots a_n}.$$

We will do so by induction on n, but in an unusual way.

(a) Prove that the inequality holds for $n = 2$. In other words, for nonnegative real numbers a_1 and a_2, show that

$$\frac{a_1 + a_2}{2} \geq \sqrt{a_1 a_2}.$$

(This equation might be of use: $(\sqrt{a} - \sqrt{b})^2 = a - 2\sqrt{ab} + b$)

(b) For some positive integer k, suppose that the AM-GM inequality holds for $n = 2^k$. Show that the AM-GM inequality holds for $n = 2^{k+1}$. (Hint: Think about how the AM-GM inequality for $n = 2$ could be used here.)

(c) For some positive integer $k \geq 2$, suppose that the AM-GM inequality holds for $n = k$. Show that the AM-GM inequality holds for $n = k - 1$. (Hint: In the AM-GM expression for $n = k$, consider substituting $a_k = \frac{a_1 + \cdots + a_{k-1}}{k-1}$.)

(d) Argue why parts a) - c) imply that the AM-GM inequality holds for all positive integers $n \geq 2$.

6 Pairing Up

Prove that for every even $n \geq 2$, there exists an instance of the stable matching problem with n jobs and n candidates such that the instance has at least $2^{n/2}$ distinct stable matchings.

7 Stable Matching for Classes!

Let's consider the system for getting into classes. For simplicity, we will start with the problem assigning students to lab sections first, since it is clear that there are a finite number of seats. We are given n students and m lab sections. Each lab section ℓ has some number, q_ℓ of seats, and we assume that the total number of students is larger than the total number of seats (i.e. $\sum_{\ell=1}^{m} q_\ell < n$) and so some students are going to end up with no lab. Each student ranks the m lab sections in order of preference, and the instructor for each lab ranks the n students. Our goal is to find an assignment of students to seats (one student per seat) that is *stable* in the following sense:

- There is no student-lab pair (s, ℓ) such that s prefers ℓ to her allocated lab section and the instructor for ℓ prefers s to one of the students assigned to ℓ. (This is like the stability criterion you have seen for jobs: it says there is no student-lab pair that would induce that lab instructor to kick out an existing student and take this new student instead.)
- There is no lab section ℓ for which the instructor prefers some unassigned student s to one of the students assigned to ℓ. (This extends the stability criterion to take account of the fact that some students are not assigned to labs.)

Note that this problem is almost the same as the Stable Matching problem for jobs/internships presented in the lecture note, with two differences: (i) there are more students than seats; and (ii) each lab section can have more than one seat.

Perhaps you will agree that this extended model is more realistic, even for the jobs context!

(a) Explain how to modify the propose-and-reject algorithm so that it finds a stable assignment of students to seats. [*Hint*: What roles of students/instructors will be in the propose-and-reject algorithm? What does "candidates have a job offer in hand (on a string)" mean in this setting?]

(b) State a version of the Improvement Lemma (see the Stable Matchings Lecture Note) that applies to your algorithm, and prove that it holds.

(c) Use your Improvement Lemma to give a proof that your algorithm terminates, that every seat is filled, and that the assignment your algorithm returns is stable.

(d) Let us consider the potential of a pair of students wishing to swap lab sections, subject to global stability (i.e. the swap can't make the matching as a whole unstable). Either prove that your algorithm will not have any such swap requests or modify it to have no such swap requests and prove that the modified one will not have any pair of student wanting to stably-swap sections.