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.

Solution:

(a) TRUE. Suppose $P(1)$ is true. That provides the base case of the induction, and together with the inductive step this implies that $P(n)$ is true for all n, including for $n = 29$. But this is a contradiction, as $P(29)$ is false, so $P(1)$ must be false.

(b) TRUE. Since $P(29)$ is false, and there is no base case to start the induction. So $P(50)$ is not necessarily true. But there is nothing preventing $P(50)$ from being true, so it could either betrue or false.

(c) FALSE. Consider lists J_1: $C_1 > C_2$, J_2: $C_2 > C_1$, C_1: $J_1 > J_2$, C_2: $J_1 > J_2$

(d) TRUE. if J is the job at the top and its favorite candidate is C, then if (J,C') and (C,J') are in a pairing, (J,C) are a rogue couple because they mutually prefer each other.

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.

Solution:

For $n = 1$, let the 3 airports be A, B, C and let their distance be $|AB|, |AC|, |BC|$. Without loss of generality suppose B,C is the closest pair of airports (which is well defined since all distances are different). Then the airplanes departing from B and C are flying towards each other. Since the airplane from A must fly to somewhere else, no airplanes are destined for airport A.

Now suppose the statement is proven for $n = k$, i.e. when there are $2k+1$ airports. For $n = k+1$, i.e. when there are $2k+3$ airports, the airplanes departing from the closest two airports must be destined for each other's starting airports. Removing these two airports reduce the problem to $2k+1$ airports. From the inductive hypothesis, we know that among the $2k+1$ airports remaining, there is an airport with no incoming flights which we call airport Z. When we add back the two airports that we removed, the airplane flights may change; in particular, it is possible that an airplane will now choose to fly to one of these two airports (because the airports that were added may be closer than the airport to which the airplane was previously flying), but observe that none of the airplanes will is destined for the airport Z. Also, the two airports that were added back will have airplanes destined for each other, so they too will not be destined for airport Z. We conclude that the airport Z will continue to have no incoming flights when we add back the two airports, and so the statement holds for $n = k+1$. By induction, the claim holds for all $n \geq 3$.

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?)

Solution:

Show that induction based on this claim doesn't get us anywhere. Try a few cases and come up with a stronger inductive hypothesis. For example:

- $\frac{1}{3} = \frac{1}{2} - \frac{1}{6}$
- $\frac{1}{3} + \frac{1}{9} = \frac{1}{2} - \frac{1}{18}$
- $\frac{1}{3} + \frac{1}{9} + \frac{1}{27} = \frac{1}{2} - \frac{1}{54}$

One possible statement is

$$\frac{1}{3^1} + \frac{1}{3^2} + \ldots + \frac{1}{3^n} = \frac{1}{2} - \frac{1}{2 \cdot 3^n}$$

- *Base Case:* $n = 1$. $\frac{1}{3} = \frac{1}{2} - \frac{1}{6}$. True.

- *Inductive Hypothesis:* Assume the statement holds for $n \geq 1$.
- *Inductive Step:* Starting from the left hand side,

$$\begin{aligned}\frac{1}{3^1}+\frac{1}{3^2}+\ldots+\frac{1}{3^n}+\frac{1}{3^{n+1}} &= \frac{1}{2}-\frac{1}{2\cdot 3^n}+\frac{1}{3^{n+1}}\\ &= \frac{1}{2}-\frac{3-2}{2\cdot 3^{n+1}}\\ &= \frac{1}{2}-\frac{1}{2\cdot 3^{n+1}}.\end{aligned}$$

Therefore, $\frac{1}{3^1}+\frac{1}{3^2}+\ldots+\frac{1}{3^n}=\frac{1}{2}-\frac{1}{2\cdot 3^n}<\frac{1}{2}$.

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 $i \leq k$.

Solution:

Prove by strong induction on n.

The key insight here is that if n is divisible by 2, then it is easy to get a bit string representation of $(n+1)$ from that of n. However, if n is not divisible by 2, then $(n+1)$ will be, and its binary representation will be more easily derived from that of $(n+1)/2$. More formally:

- Base Case: $n = 1$ can be written as 1×2^0.
- Inductive Step: Assume that the statement is true for all $1 \leq m \leq n$, where n is arbitrary. Now, we need to consider $n+1$. If $n+1$ is divisible by 2, then we can apply our inductive hypothesis to $(n+1)/2$ and use its representation to express $n+1$ in the desired form.

$$\begin{aligned}(n+1)/2 &= c_k \cdot 2^k + c_{k-1} \cdot 2^{k-1} + \cdots + c_1 \cdot 2^1 + c_0 \cdot 2^0\\ n+1 = 2\cdot(n+1)/2 &= c_k \cdot 2^{k+1} + c_{k-1} \cdot 2^k + \cdots + c_1 \cdot 2^2 + c_0 \cdot 2^1 + 0 \cdot 2^0.\end{aligned}$$

  Otherwise, n must be divisible by 2 and thus have $c_0 = 0$. We can obtain the representation of $n+1$ from n as follows:

$$\begin{aligned}n &= c_k \cdot 2^k + c_{k-1} \cdot 2^{k-1} + \cdots + c_1 \cdot 2^1 + 0 \cdot 2^0\\ n+1 &= c_k \cdot 2^k + c_{k-1} \cdot 2^{k-1} + \cdots + c_1 \cdot 2^1 + 1 \cdot 2^0\end{aligned}$$

  Therefore, the statement is true.

Here is another alternate solution emulating the algorithm of converting a decimal number to a binary number.

- Base Case: $n = 1$ can be written as 1×2^0.
- Inductive Step: Assume that the statement is true for all $1 \leq m \leq n$, for arbitrary n. We show that the statement holds for $n+1$. Let 2^m be the largest power of 2 such that $n+1 \geq 2^m$. Thus, $n+1 < 2^{m+1}$. We examine the number $(n+1)-2^m$. Since $(n+1)-2^m < n+1$, the inductive hypothesis holds, so we have a binary representation for $(n+1)-2^m$. Also, since $n+1 < 2^{m+1}$, $(n+1)-2^m < 2^m$, so the largest power of 2 in the representation of $(n+1)-2^m$ is 2^{m-1}. Thus, by the inductive hypothesis,

  $$(n+1)-2^m = c_{m-1} \cdot 2^{m-1} + c_{m-2} \cdot 2^{m-2} + \cdots + c_1 \cdot 2^1 + c_0 \cdot 2^0,$$

  and adding 2^m to both sides gives

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

  which is a binary representation for $n+1$. Thus, the induction is complete.

Another intuition is that if x has a binary representation, $2x$ and $2x+1$ do as well: shift the bits and possibly place 1 in the last bit. The above induction could then have proceeded from n and used the binary representation of $\lfloor n/2 \rfloor$, shifting and possibly setting the first bit depending on whether n is odd or even.

Note: In proofs using simple induction, we only use $P(n)$ in order to prove $P(n+1)$. Simple induction gets stuck here because in order to prove $P(n+1)$ in the inductive step, we need to assume more than just $P(n)$. This is because it is not immediately clear how to get a representation for $P(n+1)$ using just $P(n)$, particularly in the case that $n+1$ is divisible by 2. As a result, we assume the statement to be true for all of $1, 2, \ldots, n$ in order to prove it for $P(n+1)$.

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 $n \geq 2$.

Solution:

(a) We can manipulate the equation:

$$\begin{aligned} \frac{a+b}{2} &\geq \sqrt{ab} \\ \iff a+b &\geq 2\sqrt{ab} \\ \iff a-2\sqrt{ab}+b &\geq 0 \\ \iff (\sqrt{a}-\sqrt{b})^2 &\geq 0, \end{aligned}$$

where we know the last inequality is true because the square of a real number is nonnegative. Thus, our initial inequality holds.

(b) Let $a_1, \cdots, a_{2^k}, a_{2^k+1}, \cdots, a_{2^{k+1}}$ be any nonnegative real numbers. By our hypothesis, we know that

$$l_1 = \frac{a_1+\cdots+a_{2^k}}{2^k} \geq \sqrt[2^k]{a_1\cdots a_{2^k}} = r_1$$

and

$$l_2 = \frac{a_{2^k+1}+\cdots+a_{2^{k+1}}}{2^k} \geq \sqrt[2^k]{a_{2^k+1}\cdots a_{2^{k+1}}} = r_2.$$

For ease of notation, we define l_1, l_2, r_1, and r_2 as above. Then applying part a), we know that

$$\frac{l_1+l_2}{2} \geq \sqrt{l_1 l_2} \geq \sqrt{r_1 r_2}.$$

On the left hand side, we have that

$$\frac{l_1+l_2}{2} = \frac{\frac{a_1+\cdots+a_{2^k}}{2^k} + \frac{a_{2^k+1}+\cdots+a_{2^{k+1}}}{2^k}}{2} = \frac{a_1+\cdots+a_{2^{k+1}}}{2^{k+1}},$$

and on the right hand side, we have that

$$\sqrt{r_1 r_2} = \sqrt{\sqrt[2^k]{a_1\cdots a_{2^k}}\ \sqrt[2^k]{a_{2^k+1}\cdots a_{2^{k+1}}}} = \sqrt[2^{k+1}]{a_1\cdots a_{2^{k+1}}},$$

so we conclude that

$$\frac{a_1+\cdots+a_{2^{k+1}}}{2^{k+1}} \geq \sqrt[2^{k+1}]{a_1\cdots a_{2^{k+1}}},$$

which is AM-GM for $n = 2^{k+1}$, as desired.

(c) Let $a_1, \cdots, a_k$ be any nonnegative real numbers. By our hypothesis, we know that

$$\frac{a_1 + \cdots + a_k}{k} \geq \sqrt[k]{a_1 \cdots a_k}.$$

Let $a_k = \frac{a_1 + \cdots + a_{k-1}}{k-1}$. Then we have that

$$\begin{aligned}
\frac{a_1 + \cdots + \frac{a_1 + \cdots + a_{k-1}}{k-1}}{k} &= \frac{(a_1 + \cdots + a_{k-1}) + \frac{a_1 + \cdots + a_{k-1}}{k-1}}{k} \\
&= \frac{k\frac{a_1 + \cdots + a_{k-1}}{k-1}}{k} \\
&= \frac{a_1 + \cdots + a_{k-1}}{k-1},
\end{aligned}$$

and

$$\sqrt[k]{a_1 \cdots a_k} = \sqrt[k]{a_1 \cdots a_{k-1} \frac{a_1 + \cdots + a_{k-1}}{k-1}},$$

so

$$\frac{a_1 + \cdots + a_{k-1}}{k-1} \geq \sqrt[k]{a_1 \cdots a_{k-1} \frac{a_1 + \cdots + a_{k-1}}{k-1}}.$$

Raising both sides to the k power and dividing both sides by $\frac{a_1 + \cdots + a_{k-1}}{k-1}$ gives

$$\left(\frac{a_1 + \cdots + a_{k-1}}{k-1}\right)^{k-1} \geq a_1 \cdots a_{k-1},$$

so taking the $k-1$th root of both sides gives

$$\frac{a_1 + \cdots + a_{k-1}}{k-1} \geq \sqrt[k-1]{a_1 \cdots a_{k-1}},$$

which is AM-GM for $n = k-1$, as desired.

(d) For any positive integer k, there exists a power of 2 2^ℓ greater or equal to k. Thus, starting at the base case in part a), we can use part b) to show that AM-GM applies for $n = 2^\ell$, then use part c) to show that AM-GM applies to $n = k$.

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.

Solution:

To prove that there exists such a stable matching instance for any even $n \geq 2$, we just need to show how to construct such an instance.

The idea here is that we can create pairs of jobs and pairs of candidates: pair up job $2k-1$ and $2k$ into a pair and candidate $2k-1$ and $2k$ into a pair, for $1 \leq k \leq n/2$ (you might come to this idea since we are asked to prove this for *even* n).

For n, we have $n/2$ pairs. Choose the preference lists such that the kth pair of jobs rank the kth pair of candidates just higher than the $(k+1)$th pair of candidates (the pairs wrap around from the last pair to the first pair), and the kth pair of candidates rank the kth pair of jobs just higher than the $(k+1)$th pair of jobs. Within each pair of pairs (j, j') and (c, c'), let j prefer c, let j' prefer c', let c prefer j', and let c' prefer j.

Each match will have jobs in the kth pair paired to candidates in the kth pair for $1 \leq k \leq n/2$.

A job j in pair k will never form a rogue couple with any candidate c in pair $m \neq k$. If $m > k$, then c prefers her current partner in the kth pair to j. If $m < k$, then j prefers its current partner in the kth pair to c. Then a rogue couple could only exist in the same pair - but this is impossible since exactly one of either j or c must be matched to their preferred choice in the pair.

Since each job in pair k can be stably matched to either candidate in pair k, and there are $n/2$ total pairs, the number of stable matchings is $2^{n/2}$.

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.

Solution:

(a) We will extend the propose-and-reject algorithm given in the lecture notes. Students will play the role of jobs and discussion section instructors will play the role of candidates. Instead of keeping a single person as in the original algorithm, each discussion section instructor will keep a *waitlist* of size equal to its quota.

Note that there are other valid ways to modify the propose-and-reject algorithm such that a stable assignment is produced. One way is to have the instructors playing the role of jobs and the students playing the role of candidates, with the difference that now we have jobs proposing to up to q_u candidates each day. It is also possible to "expand" each instructor by the size of his or her quota by making q_u copies of instructor u and adding empty discussions for the unassigned students.

The extended procedure for students proposing and instructors keeping a waitlist works as follows:

- All students apply to their first-choice discussion section.
- Each discussion section u with a quota of q_u, then places on its waitlist the q_u applicants who rank highest (or all the applicants if there are fewer than q_u of them) and rejects all the rest.
- Rejected applicants then apply to their second-choice discussion section, and again each discussion section instructor u selects the top q_u students from among the new applicants AND those on its waitlist; it puts the selected students on its new waitlist, and rejects the rest of its applicants (including those who were previously on its waitlist but now are not).
- The above procedure is repeated until every applicant is either on a waitlist or has been rejected from every discussion section. At this point, each discussion section admits everyone on its waitlist.

(b) Improvement Lemma: Assume that discussion sections maintain their waitlist in decreasing order of their preference for the students on the lists. Let $\oplus$ be the "null" element, which we will use as a placeholder in waitlist positions not yet assigned to a student. For any discussion section u, let q_u be its quota and let $s_i^k \in \{\text{Students}\} \cup \{\oplus\}$ be the student in the i'th position on the waitlist after the k'th round of the algorithm. (If there are fewer than q_u students on the list, we fill the bottom of the list with $\oplus$'s.) Then for all i and k, the discussion section instructor likes s_i^{k+1} at least as much as s_i^k. (Here we assume that the discussion section instructor prefers any student to no student.)

In short, the lemma says that no position in the list ever gets worse for the discussion section instructor as the algorithm proceeds. As in the Lecture Notes, we use the Well-Ordering Principle and prove the lemma by contradiction:

Suppose that the jth day, where $j > k$, is the first counterexample where, for index $i \leq q_u$, discussion section u, has either nobody or some student $\hat{s}$ inferior to s_i^k. Then on day $j-1$, the instructor has some student $\tilde{s}$ on a string that they like at least as much as s_i^k. Following the algorithm, $\tilde{s}$ still "proposes" to the instructor of section u on day j since they said "maybe" the previous day. Therefore, the instructor has the choice of at least one student on the jth day, and his or her best option is at least as good as $\tilde{s}$, so the instructor would have chosen $\tilde{s}$ over $\hat{s}$. Therefore on day j, the instructor *does* have a student that they like at least as much as s_i^k in waitlist position $i \leq q_u$. This contradicts our initial assumption.

(c) First, the algorithm terminates. This follows by similar reasoning to the original propose-and-reject algorithm: in each round (except the last), at least one discussion section is crossed off the list of some rejected students.

Second, every seat is filled. Suppose some discussion section u has an unfilled seat at the end. Then the total number of students who applied to u must have been fewer than its quota q_u (since the Improvement Lemma in Part (b) ensures that a waitlist slot, once filled, will never later be unfilled). But the only students who do *not* apply to u are those who find a slot in some other discussion section. And since we are told that there are more students than seats, the number of students applying to u must be at least q_u.

Finally, the assignment is stable:

- Suppose there is a student-section pair (s,u) such that s prefers u to her discussion section u' in the final allocation. Then s must have proposed to u prior to proposing to u', and was rejected by u. Thus immediately after rejecting s, u must have had a full waitlist in which every student was preferred to s. By the Improvement Lemma, the same holds at all future times and hence at the end. Thus u does not prefer s to any of its assigned students, as required.
- Suppose s is a student left unassigned at the end. Consider any discussion section u. Since s must have applied to and have been rejected from all discussion sections, this holds in particular for u. Reasoning similar as in the previous case, using the Improvement Lemma, we see that in the final allocation, u prefers all its students to s. Therefore u does not prefer s to any of its assigned students, as required.

This concludes the proof that our algorithm finds a stable assignment when terminates.

(d) The intuition here is that the group proposing in the Propose-and-Reject algorithm is the group for which the resulting stable matchings are optimal. We proceed with a proof.

Assume that there is a pair (s,u) and a pair (s',u') such that s' prefers u to u' and s prefers u' to u. If this is the case, then s must have proposed to u' before u, and gotten rejected, and s' must have proposed to u before u', and gotten rejected.

Without loss of generality, assume s was rejected first.

If u' rejected any student, s'' that it prefers to s throughout the algorithm, then if s and s' swap, s" and u' would become a rogue pair which means swapping makes for an unstable pairing and we are done.

Thus, we can assume u' never rejects any student it prefers to s. This and the fact that everyone on u''s waitlist was preferred to s when s was rejected by u', implies that the waitlist for u' on that day must be the exact students that are in u' wailtist at the end of the algorithm. Thus, s' must be on u''s waitlist at that time and therefore must have been rejected by u previously.

This contradicts the assumption that s was rejected first. Thus, switching the pairs creates an unstable matching.