CS 70 Discrete Mathematics and Probability Theory
Fall 2022 Satish Rao and Babak Ayazifar

HW 02

Due: Saturday, 9/10, 4:00 PM
Grace period until Saturday, 9/10, 6:00 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.)

I did not work with anybody to complete this homework.

1 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: We solve this by induction. First, construct the base case with 3 airports. First we label the airports to be A, B and C and the routes connecting them a, b, c. Then, without loss of generality, we can let $a < b < c$. Clearly, since each airport is connected with the others, there will be no airplane that travels along route c.

Now the inductive step: we assume that with $2n+1$ airports, there exists an aiport with no airplanes destined for it, call this airport P. Now we wish to prove that this works for $2n+3$ airports.

We know that we must connect the two airports such that they conect to airport P, so that there exists a plane destined for P. However, notice that by adding another two airports, we've again made a triangle (with airports P, and the two that we just introduced), which as we know from our base case, will always have an airport that has no plane destined for it. Thus, we have proven our inductive step and we are done. ■

2 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: There is the formula for the finite sum of a geometric series:

$$S_n = \frac{a(1-r^n)}{1-r}$$

Substituting $a = \frac{1}{3}$ and $r = \frac{1}{3}$ then we get:

$$S_n = \frac{1}{2}\left(1 - \frac{1}{3^n}\right)$$

Notice that $1 - \frac{1}{3^n} < 1$ for all n, thus this sum is always less than $\frac{1}{2}$. We now prove that this is the correct formula to evaluate S_n, using induction.

Base case: $n = 1$. We have one term in our geometric series, so we know that the left hand side is $\frac{1}{3}$. Computing S_1:

$$S_1 = \frac{1}{2} \cdot \frac{2}{3} = \frac{1}{3}$$

Thus our base case is true.

Inductive Hypothesis: Suppose that the formula for S_n is:

$$S_n = \frac{1}{2}\left(1 - \frac{1}{3^n}\right)$$

Our goal will be to show that this is true for S_{n+1} as well. Notice that we can write:

$$\begin{aligned}
\frac{1}{3}+\cdots+\frac{1}{3^{n+1}} &= S_n + \frac{1}{3^{n+1}} \\
&= \frac{1}{2}\left(1-\frac{1}{3^n}\right)+\frac{1}{3^{n+1}} \\
&= \frac{1}{2}-\frac{1}{3^n}\left(\frac{1}{2}-\frac{1}{3}\right) \\
&= \frac{1}{2}-\frac{1}{3^n}\cdot\frac{1}{6} \\
&= \frac{1}{2}\left(1-\frac{1}{3^n}\cdot\frac{1}{3}\right) \\
&= \frac{1}{2}\left(1-\frac{1}{3^{n+1}}\right)
\end{aligned}$$

Which is exactly what we wanted to show. Thus, this is a valid formula for the sum of our series on the left. We've also shown earlier that $S_n < \frac{1}{2}$ for all n, and so we are done. ■

3 Inductive Charging

There are n cars on a circular track. Among all of them, they have exactly enough fuel (in total) for one car to circle the track. Two cars at the same location may transfer fuel between them. More formally:

- Order cars clockwise around the track, starting at some arbitrary car.
- Let the fuel in car i be f_i liters, where 1 liter of gas corresponds to 1 kilometer of travel.
- Let the track be D kilometers around, and let the distance between car i and car $i+1$ (in the modular sense) be d_i kilometers.

(a) Prove, using whatever method you want, that there exists at least one car that has enough fuel to reach the next car along the track.

Solution: Suppose that there exists a configuration of n cars such that no car can make it to the next car on the track. Then, have every car drive in the direction of the car in front. Since $f_i < d_i$ for all i, then the sum total distance they travel must be less than D, since $\sum d_i = D$ and $f_i < d_i$ for all i. Thus, we have reached a contradiction, since the sum of the fuel in all cars should be able to travel a distance D. Therefore, there will always exist at least one car that has enough fuel to reach the next car along the track. ■

(b) Prove that there exists one car that can circle the track, by gathering fuel from other cars along the way. (That is, one car moving and all others stopped). Hint: Use the previous part. [1]

Solution: We prove this by induction.

Base case: Suppose we have 2 cars. We skip the case for 1 car since it's too trivial. In a situation with 2 cars, we already know from part (a) that one car can make it to the next. Then, after collecting the other car's fuel, it now has all the alloted fuel, and thus can circle the track.

Inductive Hypothesis: Assume that for a configuration of n cars, there exists a car that can make it all the way around the track. We will snow that this is also true for $n+1$ cars.

In the case with $n+1$ cars, we still know from part (a) that at least one of these cars can make it to the next car. Let this car be car i, and have it make its way to the next car and gather the fuel from that car. We are now reduced to n cars (since one of them has no fuel, and thus basically doesn't exist), in which we know from our inductive hypothesis that there exists a car that can circle the track.

Now it remains to prove that car i is now the only car that can circle the track. We do this by by observing that the total remaining fuel is $D - d_i$, since car i travelled a distance d_i to reach the next car. Similarly, car i only has a distance $D - d_i$ left before it circles the track is car i, and since it is the only car that has moved so far, it is the also the only car that has this distance remaining.

[1] To think about, after you complete this problem: Is your proof constructive or non-constructive? (That is, does it actually point to the exact car that can complete the track, or does it just prove that one such car must exist?) If it's non-constructive, then how do we actually find this car? (Can we write a program to do this, faster than actually trying every car?)

As a result, car i is now the only car that can circle the track, since it is the only car that can feasibly achieve this considering the amount of fuel remaining. Combining this with our inductive hypothesis that in any arrangement of n cars there must exist one car that can circle the track, the inductive hypothesis is proved for $n+1$ cars, and we are done. Note that our inductive hypothesis still applies, since after car i has moved the situation is identical to the original problem except with a new track distance of $D-d_i$ and fuel $D-d_i$, which has nothing to do with our indcutive hypothesis since D was chosen arbitrarily anyway. ■

4 A Coin Game

Your "friend" Stanley Ford suggests you play the following game with him. You each start with a single stack of n coins. On each of your turns, you select one of your stacks of coins (that has at least two coins) and split it into two stacks, each with at least one coin. Your score for that turn is the product of the sizes of the two resulting stacks (for example, if you split a stack of 5 coins into a stack of 3 coins and a stack of 2 coins, your score would be $3 \cdot 2 = 6$). You continue taking turns until all your stacks have only one coin in them. Stan then plays the same game with his stack of n coins, and whoever ends up with the largest total score over all their turns wins.

Prove that no matter how you choose to split the stacks, your total score will always be $\frac{n(n-1)}{2}$. (This means that you and Stan will end up with the same score no matter what happens, so the game is rather pointless.)

Solution: We prove this using induction.

Base case: $n = 2$. No matter how we split this stack we will get a total sum of $1 \cdot 1 = 1 = \frac{2 \cdot 1}{2}$. (Note that $n = 1$ is not that helpful of an example since there aren't any coins to split, so I chose $n = 2$ instead.)

Induction Hypothesis: Assume that for a stack of n coins, the sum will always be $\frac{n(n-1)}{2}$, and assume this relation holds for all values less than n as well.

Now we want to show that this holds for $n+1$. Suppose that we take away k coins from this stack. Then our two piles have coin amounts $n+1-k$ and k coins each. Notice that since k is a nonzero positive number, these two values are guaranteed to be less than or equal to n, and thus our induction hypothesis applies. Thus, our total score will be the sum of each of the parts, plus the product $k(n+1-k)$ (the product for the initial division).

Note I will be skipping a lot of the intermediate steps in the algebra, simply becuase I'm too lazy to type them out.

$$\begin{aligned}
T &= \frac{(n+1-k)(n-k)}{2} + \frac{k(k-1)}{2} + k(n+1-k) \\
&= \frac{1}{2}(k^2 + n^2 - 2kn + n - k) + \frac{1}{2}(k^2 - k) + k(n+1-k) \\
&= \frac{1}{2}(2k^2 + n^2 - 2kn + n - 2k) + k(n+1-k) \\
&= \frac{1}{2}(2k(k-n-1) + n(n+1)) + k(n+1-k) \\
&= k(k-(n+1)) + \frac{n(n+1)}{2} + k(n+1-k)
\end{aligned}$$

Notice that $k(k-(n+1)) + k(n+1-k) = 0$, so the total product is $\frac{n(n+1)}{2}$, and thus we are done. ■

5 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: We prove this using induction. We start with the base case of 2 jobs and 2 candidates. Suppose we have the following table:

J_1	J_2	c_1	c_2
c_1	c_2	J_2	J_1
c_2	c_1	J_1	J_2

This instance has 2 stable matchings: $(J_1c_1),(J_2,c_2)$ and $(J_1,c_2),(J_2,c_1)$. Thus, the base case is done.

Induction Hypothesis: Assume that for n jobs and n candidates, there are $2^{n/2}$ distinct stable matchings.

We now add two jobs and two candidates. We know that on its own, 2 candidates and 2 jobs give us 2 distinct stable matchings, from our base case. If we can find a way to isolate the matchings for the n candidates and these extra two, then we can multiply them together and get $2^{n/2+1}$, which would prove our inductive step.

To do this, call the new jobs J_{n+1} and J_{n+2} and the new candidates c_{n+1} and c_{n+2}. Then, we place jobs J_{n+1} and J_{n+2} above all jobs $J_1,\ldots,J_n$ in the preference list for new candidates, and likewise we place c_{n+1} and c_{n+2} above $c_1,\ldots,c_n$ on the preference list for the new jobs. This will guarantee that the new jobs and candidates will choose each other over everybody else in a stable matching.

Similarly, for $J_1,\ldots,J_n$, place c_{n+1} and c_{n+2} lowest on the candidate preferences, and place J_{n+1} and J_{n+2} lowest on the job preferences. As a result, the n original candidates will still choose each other over the new introduced jobs.

Now, since the job pairings will be independent of each other, we can multiply the two together, obtaining $2^{n/2+1}$ total possibilities, proving our inductive step. ■

6 Nothing Can Be Better Than Something

In the stable matching problem, suppose that some jobs and candidates have hard requirements and might not be able to just settle for anything. In other words, each job/candidate prefers being unmatched rather than be matched with those below a certain point in their preference list. Let the term "entity" refer to a candidate/job. A matching could ultimately have to be partial, i.e., some entities would and should remain unmatched.

Consequently, the notion of stability here should be adjusted a little bit to capture the autonomy of both jobs to unilaterally fire employees and/or employees to just walk away. A matching is stable if

- there is no matched entity who prefers being unmatched over being with their current partner;
- there is no matched/filled job and unmatched candidate that would both prefer to be matched with each other over their current status;
- there is no matched job and matched candidate that would both prefer to be matched with each other over their current partners; and
- similarly, there is no unmatched job and matched candidate that would both prefer to be matched with each other over their current status;
- there is no unmatched job and unmatched candidate that would both prefer to be with each other over being unmatched.

(a) Prove that a stable pairing still exists in the case where we allow unmatched entities.

(HINT: You can approach this by introducing imaginary/virtual entities that jobs/candidates "match" if they are unmatched. How should you adjust the preference lists of jobs/candidates, including those of the newly introduced imaginary ones for this to work?)

Solution: Introduce two new jobs, which represent imaginary jobs that candidates can "match" to if they are unmatched. These jobs will not have preference lists of their own, since it makes no sense to have an "empty" candidate reject a job, or vice versa. We place these empty jobs and candidates into the preference list of all "live" candidates in some arbitrary fashion.

We prove that there always exists a stable pairing by induction.

Base case: Suppose $n = 1$. Here, the job can choose its top choice (whether that be either chosing the live candidate or the empty candidate), and we guarantee that this matching is stable. This is because the job has chosen its top choice, so it will always prefer being unmatched than being matched.

Inductive step: Suppose that we have n jobs, and there exists a stable pairing. We now prove that when we insert another job and candidate into this matching, that the matching is still stable.

Suppose that $J_1, \ldots, J_{n-1}$ are the "real" jobs and J_n represents the option to be unmatched, and $c_1, \ldots, c_{n-1}$ and c_n represent the same things, but with candidates. When we add another job J_{n+1} and c_{n+1}, c_{n+1} can choose to either pair with J_{n+1}, or not pair.

If it pairs with J_{n+1}, then this matching is stable, since it means that c_{n+1} prefers J_{n+1} over all of $J_1, \ldots, J_n$. There can also not be any c_i in our original set of n candidates that can prefer J_{n+1} over their matched job, since the original matching was stable.

Otherwise, if it does not pair with J_{n+1}, then it means that it prefers J_i over J_{n+1}, which still results in a stable matching situation. Furthermore, the candidate c_i which was initially matched with J_i can now match with J_{n+1}, a valid pairing. Just like the previous case, since the original n jobs and candidates are stable to begin with, there does not exist a candidate which also prefers J_{n+1} over their current job. Thus, in both cases, we have shown that this results in a stable pairing, and so we are done. ■

(b) As you saw in the lecture, we may have different stable matchings. But interestingly, if an entity remains unmatched in one stable matching, it/she must remain unmatched in any other stable matching as well. Prove this fact by contradiction.

Solution: We prove this by contradiction, as required by the problem statement. Suppose there exists a candidate c_i which was unmatched in one stable matching, but is now matched with some job J_i. Since the preference lists have not changed, it means that in the initial matching, this candidate c_i preferred the empty job J_e over job J_i. And since the original matching was stable, this also means that J_i preferred c_i over all other possible candidates it could have matched with. The only way this can happen in the original matching was if there was another candidate c_j who prefers job J_i over J_e, and that J_i prefers c_i over c_j.

Now we look at the new matching, where candidate c_i has now paired with J_i and c_j is now paired with J_e. Clearly, since c_j prefers J_i over J_e and J_i prefers c_i over c_e, then we have created a rogue pair between J_i and c_i, and thus this matching is not stable. As a result, if a candidate remains unmatched in one stable matching, then it must remain unmatched in all other stable matchings. ■