CS 70 Discrete Mathematics and Probability Theory
Fall 2017 Satish Rao and Kannan Ramchandran

HW 2

Sundry

Before you start your homework, write down your team. Who else did you work with on this homework? List names and email addresses. (In case of homework party, you can also just describe the group.) How did you work on this homework? Working in groups of 3-5 will earn credit for your "Sundry" grade.

Please copy the following statement and sign next to it:

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up.

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up. (*signature here*)

1 Induction

Prove the following identity for all positive integers n and k.

$$\sum_{j=1}^{n} j(j+1)(j+2)\cdots(j+k-1) = \frac{n(n+1)(n+2)\cdots(n+k)}{k+1}$$

Solution:

We will prove the statement $P(n) = \forall k \in \mathbb{Z}_+ \; \sum_{j=1}^{n} j(j+1)\cdots(j+k-1) = n(n+1)\cdots(n+k)/(k+1)$ by induction on n.

Base Case ($n = 1$): Evaluating the left side, we get $1\cdot 2\cdots k$ and on the right side, we get $\dfrac{1\cdot 2\cdots k\cdot(k+1)}{k+1} = 1\cdot 2\cdots k$. Hence, the base case holds.

Induction Hypothesis: Assume that the equality holds for a fixed $n \geq 1$ (and all $k \in \mathbb{Z}_+$). That is:

$$\sum_{j=1}^{n} j(j+1)(j+2)\cdots(j+k-1) = \frac{n(n+1)(n+2)\cdots(n+k)}{k+1}$$

$$\text{Or: } \sum_{j=1}^{n}\prod_{i=0}^{k-1}(j+i) = \frac{1}{k+1}\prod_{i=0}^{k}(n+i)$$

We must prove that the equality holds true for $n+1$. That is:

$$\sum_{j=1}^{n+1}\prod_{i=0}^{k-1}(j+i) \quad = \quad \frac{1}{k+1}\prod_{i=0}^{k}(n+1+i)$$

Inductive Step: In order to obtain the left hand side of the equation above from the equation in the inductive hypothesis, we add $(n+1)\cdot(n+2)\cdots(n+k)$ to both sides of the equation in the inductive hypothesis. This is because $(n+1)\cdot(n+2)\cdots(n+k)$ is the difference between the two terms.

$$\begin{aligned}
\sum_{j=1}^{n}\prod_{i=0}^{k-1}(j+i)+(n+1)\cdot(n+2)\cdots(n+k) &= \frac{1}{k+1}\prod_{i=0}^{k}(n+i)+(n+1)\cdot(n+2)\cdots(n+k) \\
\Longrightarrow \sum_{j=1}^{n+1}\prod_{i=0}^{k-1}(j+i) &= \frac{1}{k+1}\left(\prod_{i=0}^{k}(n+i)+(k+1)\prod_{i=1}^{k}(n+i)\right) \\
\Longrightarrow \sum_{j=1}^{n+1}\prod_{i=0}^{k-1}(j+i) &= \frac{1}{k+1}\left(n\cdot\prod_{i=1}^{k}(n+i)+(k+1)\prod_{i=1}^{k}(n+i)\right)
\end{aligned}$$

Factoring $\prod_{i=1}^{k}(n+i)$ on the right hand side:

$$\begin{aligned}
\Longrightarrow \sum_{j=1}^{n+1}\prod_{i=0}^{k-1}(j+i) &= \frac{1}{k+1}(n+k+1)\prod_{i=1}^{k}(n+i) \\
\Longrightarrow \sum_{j=1}^{n+1}\prod_{i=0}^{k-1}(j+i) &= \frac{1}{k+1}\prod_{i=1}^{k+1}(n+i)
\end{aligned}$$

Rewriting the right hand side of the equation to exactly match the statement we aimed to prove by shifting the index backward by 1:

$$\sum_{j=1}^{n+1}\prod_{i=0}^{k-1}(j+i) \quad = \quad \frac{1}{k+1}\prod_{i=0}^{k}(n+1+i)$$

This completes the induction.

2 Finite Number of Solutions

Prove that for every positive integer k, the following is true:

> For every real number $r>0$, there are only finitely many solutions in positive integers to
>
> $$\frac{1}{n_1}+\cdots+\frac{1}{n_k}=r.$$

In other words, there exists some number m (that depends on k and r) such that there are at most m ways of choosing a positive integer n_1, and a (possibly different) positive integer n_2, etc., that satisfy the equation.

Solution:

We will first transfer the problem to mathematical notation:

Claim: $\forall k \in \mathbb{Z}\ \forall r \in \mathbb{R}\ \big((k > 0 \wedge r > 0) \implies$ (there are finitely many solutions to $n_1^{-1} + \cdots + n_k^{-1} = r, n_i \in \mathbb{Z}, n_i > 0)\big)$

Proof: We will prove this by induction on k.

Base Case ($k = 1$): In the base case, iff r can be written as $1/n_1$ when n_1 is a positive integer, then there is exactly one solution, $n_1 = 1/r$. If r cannot be written in that form, then there are exactly zero solutions.

Inductive Hypothesis: In all cases, there is a finite number of solutions. Assume that there are finitely many solutions for some $k \geq 1$ for all r.

Inductive Step: Each real number r_1 either can or cannot be written as the sum of $k+1$ integers' inverses. If r_1 cannot be written in that form, then there are exactly zero solutions. If r_1 can be written in that form, then the integers' inverses can be ordered. Since r_1 is the sum of $k+1$ integers' inverses, the largest $1/n_i$ must be at least $r_1/(k+1)$. This means that the smallest n_i must be at most $(k+1)/r_1$, which means that the smallest n_i has finitely many possible values. For each of the possible smallest n_i values, there is a real number $r_1 - 1/n_i$ that can be written as the sum of k integers' inverses in finitely many ways (using the induction hypothesis). This means that there are only finitely many possible solutions for $k+1$ (combining all solutions (finitely many) for each possible smallest n_i values (finitely many)). By the principle of induction, there are finitely many solutions for all k for all r.

3 Induction with Two Directions

Pacman is walking on an infinite 2D grid. He starts at some location $(i, j) \in \mathbb{N}^2$ in the first quadrant, and is constrained to stay in the first quadrant (say, by walls along the x and y axes). Every second he does one of the following (if possible):

(i) Walk one inch down, to $(i, j-1)$.

(ii) Walk one inch left, to $(i-1, j)$.

For example, if he is at $(5,0)$, his only option is to walk left to $(4,0)$.

Prove that no matter how he walks, he will always reach $(0,0)$ in finite time.

Solution:

First, consider proving the statement for all starting locations $(x,0)$, by simple induction. Then prove $(0,1)$ directly (only one possible move). Then consider $(1,1)$: Depending on the move, this reduces to either $(0,1)$ or $(1,0)$. Use this idea to prove by strong induction for all starting locations $(x,1)$. Consider doing the same thing for all $(x,2)$, then generalize by induction for (x,y).

One proof method is to define a potential function and show that it decreases a finite number of times until a certain point, which is used to show termination. This function could be $\Phi = x + y$, where x, y are the coordinates on the grid. The function is always nonnegative on the coordinates, so it only decreases finitely many times as the Pacman moves down.

Here is an example proof:

We can strengthen the Induction Hypothesis to show that the number of steps required from (i, j) is exactly $i + j$ for all naturals $i, j \in \mathbb{N}$. The base case is obvious. For the induction step, Pacman either moves to $(i, j-1)$ or $(i-1, j)$ in one step, and by the induction hypothesis a further $i + j - 1$ steps are required, for total of $i + j$.

Next: What if the Pacman could move left either 1 or 2 inches?

4 Stable Marriage

Consider a set of four men and four women with the following preferences:

men	preferences	women	preferences
A	1>2>3>4	1	D>A>B>C
B	1>3>2>4	2	A>B>C>D
C	1>3>2>4	3	A>B>C>D
D	3>1>2>4	4	A>B>D>C

(a) Run on this instance the stable matching algorithm presented in class. Show each stage of the algorithm, and give the resulting matching, expressed as $\{(M,W),\ldots\}$.

(b) We know that there can be no more than n^2 stages of the algorithm, because at least one woman is deleted from at least one list at each stage. Can you construct an instance with n men and n women so that at least $n^2/2$ stages are required?

(c) Suppose we relax the rules for the men, so that each unpaired man proposes to the next woman on his list at a time of his choice (some men might procrastinate for several days, while others might propose and get rejected several times in a single day). Can the order of the proposals change the resulting pairing? Give an example of such a change or prove that the pairing that results is the same.

Solution:

(a) The situations on the successive days are:
Day 1: Proposals: $\{(A,1),(B,1),(C,1),(D,3)\}$, B and C are rejected.
Day 2: Proposals: $\{(A,1),(B,3),(C,3),(D,3)\}$, C and D are rejected.
Day 3: Proposals: $\{(A,1),(B,3),(C,2),(D,1)\}$, A is rejected.
Day 4: Proposals: $\{(A,2),(B,3),(C,2),(D,1)\}$, C is rejected.
Day 5: Proposals: $\{(A,2),(B,3),(C,4),(D,1)\}$, no one is rejected.
Final matching: $(A,2),(B,3),(C,4),(D,1)$.

(b) Consider the case where the preference lists have the following structure:

men	preferences	women	preferences
1	$1 > 2 > \cdots > n-1 > n$	1	$2 > 3 > \cdots > n > 1$
2	$2 > 3 > \cdots > 1 > n$	2	$3 > 4 > \cdots > 1 > 2$
3	$3 > 4 > \cdots > 2 > n$	3	$4 > 5 > \cdots > 2 > 3$
	$\cdots$		$\cdots$
$n-1$	$n-1 > 1 > \cdots > n-2 > n$	$n-1$	$n > 1 > \cdots > n-2 > n-1$
n	$1 > 2 > \cdots > n-1 > n$	n	$1 > 2 > \cdots > n-1 > n$

In this case, man 1 and n go to woman 1 on the first day (while any other man i goes to woman i), and woman 1 rejects man 1. He then goes to woman 2 the next day, who rejects man 2, and so on. It can be shown that exactly one man is rejected every day until the algorithm terminates with a pairing. For the first n days, it can be seen that on day i, the i-th man is rejected by the i-th woman (except for man n who is rejected by woman 1). Similarly, in the next n days, man i is rejected by woman $i+1$ (except for man $n-1$ who is rejected by woman 1, and man n who is rejected by woman 2). This pattern continues until man 1 proposes to woman n, at which point all the men except for man 1 are on the string of the woman who is second last on their preference lists. Since there is exactly 1 rejection each day until the algorithm terminates, counting the number of days simply involves counting the number of rejections across all men. As man 1 ends up with the woman last on his preference list, man 1 was rejected $n-1$ times. Each of the other men end up with the woman second last on their respective preference lists, so each of the other $n-1$ men was rejected $n-2$ times. This brings the total number of rejections to $(n-1)(n-2)+n-1$. On the last day of the algorithm, there is no rejection. Hence, the total number of days required to terminate is $(n-1)(n-2)+n-1+1$, which is equal to $n^2-2n+2 \geq n^2/2$.

(c) Assume, that when a proposal is made and an answer is received, we write down it on a list L and enumerate them. Now the proof is similar to the proof in class that the algorithms finds male optimal pairing.

We should prove that the pairing P' that results is the same as pairing P found by the regular algorithm. Assume the opposite, so either there is a woman that rejects a man who was her pair in P or there is a woman who do not receive a proposal from a man who was her pair in P. By Well Ordering Principle, there is the first entry in the list L for one of that happens, let W denote the woman for whom it happens and let M denote her pair in P.

First, if W rejected M then it is because she get proposal from other man M' who is higher in her preference list. But M' proposed her only because he was rejected by other woman W' who is higher in his preference list and did not rejects him in P. In other words, even before W rejected M, W' rejected her previous pair M'. Contradiction.

Second, if W did not get a proposal from M then it is because M was not rejected by some other woman W' who is higher in his preference list and who rejects him in P. Contradiction.

Therefore, the resulting pairing P' is the same as P.

5 The Better Stable Matching

In this problem we examine a simple way to *merge* two different solutions to a stable marriage problem. Let R, R' be two distinct stable matchings. Define the new matching $R \wedge R'$ as follows:

> For every man m, m's date in $R \wedge R'$ is whichever is better (according to m's preference list) of his dates in R and R'.

Also, we will say that a man/woman *prefers* a matching R to a matching R' if he/she prefers his/her date in R to his/her date in R'. We will use the following example:

men	preferences	women	preferences
A	1>2>3>4	1	D>C>B>A
B	2>1>4>3	2	C>D>A>B
C	3>4>1>2	3	B>A>D>C
D	4>3>2>1	4	A>B>D>C

(a) $R = \{(A,4),(B,3),(C,1),(D,2)\}$ and $R' = \{(A,3),(B,4),(C,2),(D,1)\}$ are stable matchings for the example given above. Calculate $R \wedge R'$ and show that it is also stable.

(b) Prove that, for any matchings R, R', no man prefers R or R' to $R \wedge R'$.

(c) Prove that, for any stable matchings R, R' where m and w are dates in R but not in R', one of the following holds:

- m prefers R to R' and w prefers R' to R; or
- m prefers R' to R and w prefers R to R'.

[*Hint*: Let M and W denote the sets of mens and women respectively that prefer R to R', and M' and W' the sets of men and women that prefer R' to R. Note that $|M|+|M'| = |W|+|W'|$. (Why is this?) Show that $|M| \leq |W'|$ and that $|M'| \leq |W|$. Deduce that $|M'| = |W|$ and $|M| = |W'|$. The claim should now follow quite easily.]

(You may assume this result in subsequent parts even if you don't prove it here.)

(d) Prove an interesting result: for any stable matchings R, R', (i) $R \wedge R'$ is a matching [*Hint*: use the results from (c)], and (ii) it is also stable.

Solution:

(a) $R \wedge R' = \{(A,3),(B,4),(C,1),(D,2)\}$.

(b) Let m be a man, and let his dates in R and R' be w and w' respectively, and without loss of generality, let $w > w'$ in m's list. Then his date in $R \wedge R'$ is w, whom he prefers over w'. However, for m to prefer R or R' over $R \wedge R'$, he must prefer w or w' over w, which is not possible (since $w > w'$ in his list).

(c) Let M and W denote the sets of men and women respectively that prefer R to R', and M' and W' the sets of men and women that prefer R' to R. Note that $|M| + |M'| = |W| + |W'|$, since the left-hand side is the number of men who have different partners in the two matchings, and the right-hand side is the number of women who have different partners.

Now, in R there cannot be a pair (m, w) such that $m \in M$ and $w \in W$, since this will be a rogue couple in R'. Hence the partner in R of every man in M must lie in W', and hence $|M| \le |W'|$. A similar argument shows that every man in M' must have a partner in R' who lies in W, and hence $|M'| \le |W|$.

Since $|M| + |M'| = |W| + |W'|$, both these inequalities must actually be tight, and hence we have $|M'| = |W|$ and $|M| = |W'|$. The result is now immediate: if the man m does not date the woman w in one but not both matchings, then

- either $m \in M$ and $w \in W'$, i.e., m prefers R to R' and w prefers R' to R,
- or $m \in M'$ and $w \in W$, i.e., m prefers R' to R and w prefers R to R'.

(d) (i) If $R \wedge R'$ is not a matching, then it is because two men get the same woman, or two women get the same man. Without loss of generality, assume it is the former case, with $(m, w) \in R$ and $(m', w) \in R'$ causing the problem. Hence m prefers R to R', and m' prefers R' to R. Using the results of the previous part would imply that w would prefer R' over R, and R over R' respectively, which is a contradiction.

(ii) Now suppose $R \wedge R'$ has a rogue couple (m, w). Then m strictly prefers w to his partners in both R and R'. Further, w prefers m to her partner in $R \wedge R'$. But w is matched to the better of her partners in R and R'. Let w's partners in R and R' be m_1 and m_2. If she is finally matched to m_1, then (m, w) is a rogue couple in R; on the other hand, if she is matched to m_2, then (m, w) is a rogue couple in R'. Since these are the only two choices for w's partner, we have a contradiction in either case.

6 Stable Matching for Classes!

Let's consider the system for getting into classes. We are given n students and m discussion sections. Each discussion section u has some number, q_u of seats, and we assume that the total number of students is larger than the total number of seats (i.e. $\sum_{u=1}^{m} q_u < n$). Each student ranks the m discussion sections in order of preference, and the instructor for each discussion 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-section pair (s, u) such that s prefers u to her allocated discussion section and the instructor for u prefers s to one of the students assigned to u. (This is like the stability criterion for Stable Marriage: it says there is no student-section pair that would like to change the assignment.)
- There is no discussion section u for which the instructor prefers some unassigned student s to one of the students assigned to u. (This extends the stability criterion to take account of the fact that some students are not assigned to discussions.)

Note that this problem is almost the same as the Stable Marriage Problem, with two differences: (i) there are more students than seats; and (ii) each discussion section can have more than one seat.

(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 "women have men on a string" mean in this setting?]

(b) State a version of the Improvement Lemma (see Lecture Note 4) 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.

Solution:

(a) We will extend the propose-and-reject algorithm given in the lecture notes. Students will play the role of men and discussion section instructors will play the role of women. 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 men and the students playing the role of women, with the difference that now we have men proposing to up to q_u women 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.