Due: Saturday, 2/4, 4:00 PM
Grace period until Saturday, 2/4, 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.)

1 Universal Preference

Note 4

Suppose that preferences in a stable matching instance are universal: all n jobs share the preferences $C_1 > C_2 > \cdots > C_n$ and all candidates share the preferences $J_1 > J_2 > \cdots > J_n$.

(a) What pairing do we get from running the algorithm with jobs proposing? Can you prove this happens for all n?

(b) What pairing do we get from running the algorithm with candidates proposing?

(c) What does this tell us about the number of stable pairings?

2 Nothing Can Be Better Than Something

Note 4

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?)

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

3 A Better Stable Pairing

Note 4

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

> For every job j, j's partner in $R \wedge R'$ is whichever is better (according to j's preference list) of their partners in R and R'.

Also, we will say that a job/candidate *prefers* a pairing R to a pairing R' if they prefers their partner in R to their partner in R'.

(a) For this part only, consider the following example:

jobs	preferences	candidates	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	$A > B > D > C$

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

(b) Prove that, for any pairings R and R', no job prefers R or R' to $R \wedge R'$.

(c) Prove that, for any stable pairings R and R' where j and c are partners in R but not in R', one of the following holds:

- j prefers R to R' and c prefers R' to R; or
- j prefers R' to R and c prefers R to R'.

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

(You may assume this result in the next part even if you don't prove it here.)

(d) Prove an interesting result: for any stable pairings R and R', (i) $R \wedge R'$ is a pairing, and (ii) it is also stable.

[*Hint*: for (i), use the results from part (c).]

4 Build-Up Error?

Note 5

What is wrong with the following "proof"? In addition to finding a counterexample, you should explain what is fundamentally wrong with this approach, and why it demonstrates the danger of build-up error.

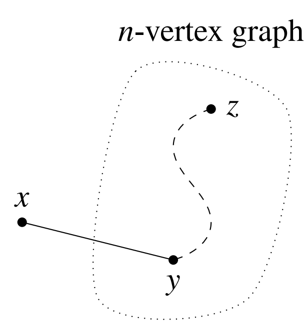
False Claim: If every vertex in an undirected graph has degree at least 1, then the graph is connected.

Proof? We use induction on the number of vertices $n \geq 1$.

Base case: There is only one graph with a single vertex and it has degree 0. Therefore, the base case is vacuously true, since the if-part is false.

Inductive hypothesis: Assume the claim is true for some $n \geq 1$.

Inductive step: We prove the claim is also true for $n+1$. Consider an undirected graph on n vertices in which every vertex has degree at least 1. By the inductive hypothesis, this graph is connected. Now add one more vertex x to obtain a graph on $(n+1)$ vertices, as shown below.



All that remains is to check that there is a path from x to every other vertex z. Since x has degree at least 1, there is an edge from x to some other vertex; call it y. Thus, we can obtain a path from x to z by adjoining the edge $\{x,y\}$ to the path from y to z. This proves the claim for $n+1$. □

5 Proofs in Graphs

Note 5

(a) On the axis from San Francisco traffic habits to Los Angeles traffic habits, Old California is more towards San Francisco: that is, civilized. In Old California, all roads were one way streets. Suppose Old California had n cities ($n \geq 2$) such that for every pair of cities X and Y, either X had a road to Y or Y had a road to X.

Prove that there existed a city which was reachable from every other city by traveling through at most 2 roads.

[*Hint:* Induction]

(b) Consider a connected graph G with n vertices which has exactly $2m$ vertices of odd degree, where $m > 0$. Prove that there are m walks that *together* cover all the edges of G (i.e., each edge of G occurs in exactly one of the m walks, and each of the walks should not contain any particular edge more than once).

[*Hint:* In lecture, we have shown that a connected undirected graph has an Eulerian tour if and only if every vertex has even degree. This fact may be useful in the proof.]

(c) Prove that any graph G is bipartite if and only if it has no tours of odd length.

[*Hint:* In one of the directions, consider the lengths of paths starting from a given vertex.]

6 Bipartite Graphs

Note 5

An undirected graph is bipartite if its vertices can be partitioned into two disjoint sets L, R such that each edge connects a vertex in L to a vertex in R (so there does not exist an edge that connects two vertices in L or two vertices in R).

(a) Suppose that a graph G is bipartite, with L and R being a bipartite partition of the vertices. Prove that $\sum_{v \in L} \deg(v) = \sum_{v \in R} \deg(v)$.

(b) Suppose that a graph G is bipartite, with L and R being a bipartite partition of the vertices. Let s and t denote the average degree of vertices in L and R respectively. Prove that $s/t = |R|/|L|$.

(c) Prove that a graph is bipartite if and only if it can be 2-colored. (A graph can be 2-colored if every vertex can be assigned one of two colors such that no two adjacent vertices have the same color).