CS 70 Discrete Mathematics and Probability Theory
Spring 2024 Seshia, Sinclair

HW 02

Due: Saturday, 2/3, 4:00 PM
Grace period until Saturday, 2/3, 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 Pairing Up

Note 4

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.

3 Upper Bound

Note 4

(a) In the notes, we show that the stable matching algorithm terminates in at most n^2 days. Prove the following stronger result: the stable matching algorithm will always terminate in at most $(n-1)^2 + 1 = n^2 - 2n + 2$ days.

(b) Provide a set of preference lists for 4 jobs and 4 candidates that will result in the upper bound from part (a) when running the Propose-and-Reject algorithm. Verify this by running the Propose-and-Reject algorithm on your preference lists.

(a) (C_1, J_1) (C_2, J_2) ... (C_n, J_n)

Induction

$C_1 > C_2 \cdots > C_k$ $J_1 > J_2 \cdots > J_k$ happens.

$> C_{k+1}$ $> J_{k+1}$

(C_1, J_1) will match, then the rest will be the Induction hypothesis

(b) (C_1, J_1), (C_2, J_2) ... (C_n, J_n)

(c) only one Theorem 4.2 and 4.3

2. Induction

stable pair.

J_{2k-1} $C_{2k-1} > C_{2k}$ C_{2k-1} $J_{2k} > J_{2k-1}$

J_{2k} $C_{2k} > C_{2k-1}$ C_{2k} $J_{2k-1} > J_{2k}$

3. (a) $(n-1)^2+1$, $n=1$, right

$n \le k$, Induction hypothesis. only reject 1 time

$n=k+1$ there is always a candidate who receives

$(k-1)^2+1$ only one proposal (on the last day)

(DISC 1B 2)

J_1 $C_1 > C_2 > C_3 > C_4$

J_2 $C_1 > C_2 > C_3 > C_4$

J_3 $C_2 > C_3 > C_1 > C_4$

J_4 $C_3 > C_1 > C_2 > C_4$

$(n-1)^2+1$

$=(4-1)^2+1=10.$

C_1 $J_3 > J_4 > J_2 > J_1$

C_2 $J_4 > J_2 > J_1 > J_3$

C_3 $J_2 > J_1 > J_3 > J_4$

C_4 $J_1 > J_2 > J_3 > J_4$

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.

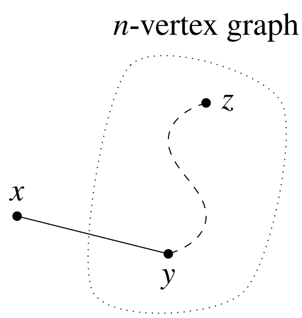
False Claim: If every vertex in an undirected graph has degree at least 1, then the graph is connected.

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 $n+1$. $\square$

5 Proofs in Graphs

Note 5

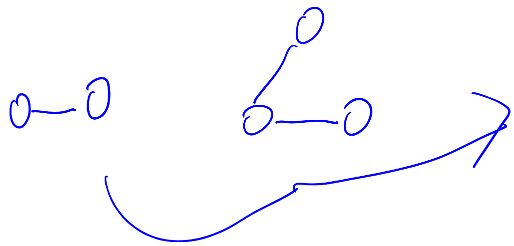
(a) On the axis from San Francisco traffic habits to Los Angeles traffic habits, Old California is more towards San Francisco: that is, civilized. In Old California, all roads were one way streets. Suppose Old California had n cities ($n \geq 2$) such that for every pair of cities X and Y, either X had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

4. there may be other graphs don't contain n vertices connected



like this

5. (a) $\underline{n=2}$ $\quad A \longrightarrow B$ $\quad$ B is the result.

assume claim is true for some $n \geq 2$

for $\underline{n+1}$, remove a vertex A and all it's edge.

the other's is also satisfy the condition

and vertex is $\underline{n}$.

So exisit a $\underline{\text{vertx}}$ B, satisfy the recommend.

add back the removed vertex A.

either

$A \to B$ $\quad$ A had a path to B, then B is still the result

or

$B \to A \to$

1. all verts have a path to A, then A is the result.
2. A must have a path to other vertx,

$B \to A$, $A \to C$, $C \to B$ $\to$ 1. C have a road to B. then A is the result.

$\downarrow$

at least have one ?

use set to describe.

remove a point x, then exist a point y satisfy the recommand.

the remain points $S = y \cup A \cup B$

A is the set of points have a road to y

B is the others.

B must have a road to A

add x back. if x have a road to $y \cup A$

then y is the result.

else if all points in $y \cup A$ have a road to x,

then x is the result

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 (Optional) 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.

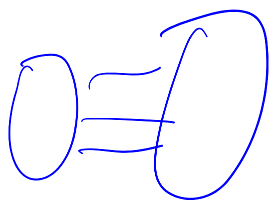
(a) Prove that a stable pairing still exists in the case where we allow unmatched entities.

(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.

5. (b) split $2m$ odd-degree into m pairs, join each pair with an edge.

(c) only if



It is bipartie, it has no tours of odd length

if have no tours of odd length.

⇓ it have no tours,

only tours of even length

shortest path

6. (a) virtual job, virtual people.

$J_1 : C_1 > C_2 > C_3 > C_1'$

$C_1 : J_1 > J_2 > J_3 > J_1'$

$J_1' : C_1 > \ldots\ldots$

$C_1' : J_1 > \ldots\ldots$

(b) A: C_1 unmatched (C_1, J)

B: but matched in (C_1, J_1) in another.

(C_1, J_2)

J_1 is also matched in A:

otherwise matching in unstach.

$(C_1, J) \to$ for J : $C_1 > C$

C_1 must be matched in B, because it reject J

say (C_1, J_1) for C_1 : $J_1 > J$

循环.

$(J_1, C_1) \to S$

$J_1 \to$ unmatched in T

(C_1, J_2)