CS 70 Spring 2026
Discrete Mathematics and Probability Theory
Sinclair, Song
HW 02

1 Airport

Note 3

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, for all $n > 0$, there is an airport which has no airplanes destined for it. (Hint: In the inductive step, it is not sufficient to start at a $2n+1$ airport example and add two airports to create a $2(n+1)+1$ airport example. Think of why this doesn't generate **all** $2(n+1)+1$ airport examples, and what you can do to fix this. Consider taking a look at **#2 on Discussion 2A**!)

Solution: We proceed by induction on n. For $n = 1$, let the 3 airports be A, B, C and 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 holds 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 (say X and Y) must be destined for each other's starting airports. Removing these two airports yields a smaller instance with $2k+1$ airports.

By the inductive hypothesis, we know that for the instance restricted to the remaining $2k+1$ airports, there is an airport with no incoming flights which we call airport Z. When we add back the two airports that we removed, there are two possible scenarios:

- Some of the flights get rerouted to X or Y (because X or Y is the closest airport to some of the $2k+1$ airports).
- None of the flights get rerouted.

In either scenario, 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 $n \geq 1$.

2 Universal Preference

Note 4

Suppose that preferences in a stable matching instance are universal, i.e., 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 propose-and-reject algorithm with jobs proposing? (Hint: Start with small examples and go through the algorithm. Do you see a pattern?) Prove that this happens for all n.

(b) What pairing do we get from running the propose-and-reject algorithm with candidates proposing? Explain.

(c) What does this tell us about the number of stable pairings? Justify your answer.

Solution:

(a) The pairing results in (C_i, J_i) for each $i \in \{1, 2, \ldots, n\}$. This result can be proved by induction:

Our base case is when $n = 1$, so the only pairing is (C_1, J_1), and thus the base case is trivially true.

Now assume this is true for some $n \in \mathbb{N}$. On the first day with $n+1$ jobs and $n+1$ candidates, all $n+1$ jobs will propose to C_1. C_1 prefers J_1 the most, and the rest of the jobs will be rejected. This leaves a set of n unpaired jobs and n unpaired candidates who all have the same preferences (after the pairing of (C_1, J_1)). By the process of induction, this means that every i^{th} preferred candidate will be paired with the i^{th} preferred job.

(b) The pairings will again result in (J_i, C_i) for each $i \in \{1, 2, \ldots, n\}$. This can be proved by induction in the same as above, but replacing "job" with "candidate" and vice-versa.

(c) We know that job-proposing produces a candidate-pessimal stable pairing. We also know that candidate-proposing produces a candidate-optimal stable pairing. We found that candidate-optimal and candidate-pessimal pairings are the same. This means that there is only one stable pairing, since both the best and worst pairings (for candidates) are the same pairings.

3 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.

(*Hint*: It can help to start with some small examples; find an instance for $n = 2$, and think about how you can use these preference lists to construct an instance for $n = 4$. After this, you should be in a good position to generalize the construction for all even n. Additionally, $2^{n/2}$ is a very specific number; try to think about how your construction would build such a number as n increases.)

Solution:

To prove that there exists such a stable matching instance for any even $n \geq 2$, it suffices to construct such an instance. But first, we look at the $n = 2$ case to generate some intuition. We can recognize that for the following preferences:

J_1	$C_1 > C_2$
J_2	$C_2 > C_1$

C_1	$J_2 > J_1$
C_2	$J_1 > J_2$

both $S = \{(J_1, C_1), (J_2, C_2)\}$ and $T = \{(C_1, J_2), (C_2, J_1)\}$ are stable pairings.

The $n/2$ in the exponent motivates us to consider pairing the n jobs into $n/2$ groups of 2 and likewise for the candidates. We pair up job $2k-1$ and $2k$ into a pair and candidate $2k-1$ and $2k$ into a pair, for $1 \leq k \leq n/2$.

From here, we recognize that for each pair (J_{2k-1}, J_{2k}) and (C_{2k-1}, C_{2k}), mirroring the preferences above would yield 2 stable matchings from the perspective of just these pairs. If we can extend this perspective to all $n/2$ pairs, this would be a total of $2^{n/2}$ stable matchings.

Our construction thus results in preference lists as follows:

J_1	$C_1 > C_2 > \dots$
J_2	$C_2 > C_1 > \dots$
$\vdots$	$\vdots$
J_{2k-1}	$C_{2k-1} > C_{2k} > \dots$
J_{2k}	$C_{2k} > C_{2k-1} > \dots$
$\vdots$	$\vdots$
J_{n-1}	$C_{n-1} > C_n > \dots$
J_n	$C_n > C_{n-1} > \dots$

C_1	$J_2 > J_1 > \dots$
C_2	$J_1 > J_2 > \dots$
$\vdots$	$\vdots$
C_{2k-1}	$J_{2k} > J_{2k-1} > \dots$
C_{2k}	$J_{2k-1} > J_{2k} > \dots$
$\vdots$	$\vdots$
C_{n-1}	$J_n > J_{n-1} > \dots$
C_n	$J_{n-1} > J_n > \dots$

Each matching will have jobs in the kth pair matched 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 $m \neq k$ since it always prefers the candidates in this pair over all candidates across other pairs. 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 $2^{n/2}$.

4 Short Tree Proofs

Note 5

Let $G = (V, E)$ be an undirected graph with $|V| \geq 1$.

(a) Prove that every connected component in an acyclic graph is a tree.

(b) Suppose G has k connected components. Prove that if G is acyclic, then $|E| = |V| - k$.

(c) Prove that a graph with $|V|$ edges contains a cycle.

Solution:

(a) Every connected component is connected, and acyclic because the graph is acyclic; by definition, this is a tree.

(b) Because each connected component is a tree, each connected component has $|V_i| - 1$ edges. The total number of edges is thus $\sum_i (|V_i| - 1) = |V| - k$.

(c) An acyclic graph has $|V| - k$ edges which cannot equal $|V|$, thus if a graph has $|V|$ edges it has a cycle.

5 Proofs in Graphs

Note 5

(a) 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. Also, note that Euler's Theorem holds for both simple graphs and multigraphs (graphs where multiple edges between one pair of vertices is permitted).]

(b) 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 the shortest paths from some fixed vertex to all the other vertices.]

Solution:

(a) We split the $2m$ odd-degree vertices into m pairs, and join each pair with an edge, adding m more edges in total. (Here, we allow for the possibility of multi-edges, that is, pairs of vertices with more than one edge between them.) Notice that now all vertices in this graph are of even degree. Now by Euler's theorem the resulting graph has an Eulerian tour. Removing the m added edges breaks the tour into m walks covering all the edges in the original graph, with each edge belonging to exactly one walk.

(b) To prove the claim, we need to prove two directions: if G is bipartite, it contains no tours of odd length, and if G contains no tours of odd length, it must be bipartite.

Suppose G is bipartite, and let L and R be the two disjoint sets of vertices such that there does not exist any edge between two vertices in L or two vertices in R. Further, suppose there is some tour in G, and we start traversing this tour at $v_0 \in L$.

Since each edge in G connects a vertex in L to a vertex in R, the first edge in the tour connects the start vertex v_0 to a vertex $v_1 \in R$. Similarly, the second edge connects $v_1 \in R$ to $v_2 \in L$. In general, it must be the case that the $2k$th edge connects vertex $v_{2k-1} \in R$ to $v_{2k} \in L$, and the $2k+1$th edge connects vertex $v_{2k} \in L$ to $v_{2k+1} \in R$.

Since only even numbered edges connect to vertices in L, and we started our tour in L, the tour must end with an even number of edges.

For the opposite direction, suppose G contains no tours of odd length. Without loss of generality, let us consider one connected component of G; the following reasoning can be applied to all of the connected components of G.

Let v be an arbitrary vertex in G; we can divide all of the vertices in G into two disjoint sets:

$$R = \{u \mid \text{the shortest path from } u \text{ to } v \text{ is even}\}$$
$$L = \{u \mid \text{the shortest path from } u \text{ to } v \text{ is odd}\}$$

We claim that no two vertices in L are adjacent. For contradiction, suppose there do exist adjacent vertices $u_1, u_2 \in L$. Consider the tour consisting of:

- the shortest path from v to u_1 (odd length)
- the edge (u_1, u_2) (length 1)
- the shortest path from u_2 to v (odd length)

This tour has odd length, and contradicts our assumption that G has no tours of odd length. This means that no two vertices in L are adjacent.

Similarly, we claim that no two vertices in R are adjacent. For contradiction, suppose there do exist adjacent vertices $u_1, u_2 \in R$. Consider the tour consisting of:

- the shortest path from v to u_1 (even length)
- the edge (u_1, u_2) (length 1)
- the shortest path from u_2 to v (even length)

This tour has odd length, and contradicts our assumption that G has no tours of odd length. This means that no two vertices in R are adjacent.

We've just shown that there are no edges between two vertices in L, and no edges between two vertices in R. If there are multiple connected components in G, the same partition can be applied to all of the components. Together, this means that G is bipartite.

6 Self-Grades

Make sure to review the self grades post on Edstem and submit your selfgrades for the previous homework assignment on Gradescope! This is just a reminder to do so, no need to submit anything for this question.

Solution: Submitted the previous homework selfgrades on Gradescope!