CS 70 Discrete Mathematics and Probability Theory
Summer 2016 Dinh, Psomas, and Ye HW 2

Due Tuesday July 5 at 1:59PM

1. (8 points: 3/5) **Hit or miss**

   For each of the claims and proofs below, state whether the claim is true or not and whether the proof is correct or not. For each incorrect proof, point out what is wrong with the proof. Simply saying that the claim or the induction hypothesis is false is *not* a valid explanation of what is wrong with the proof.

   (a) **Claim:** For all nonnegative integers n, $2n = 0$.

   Proof. We will prove by strong induction on n.
   Base Case: $2 \times 0 = 0$. It is true for $n = 0$.
   Inductive Hypothesis: Assume that $2k = 0$ for all $0 \le k \le n$.
   Inductive Step: We must show that $2(n+1) = 0$. Write $n+1 = a+b$ where $0 < a,b \le n$. From the inductive hypothesis, we know $2a = 0$ and $2b = 0$, therefore,

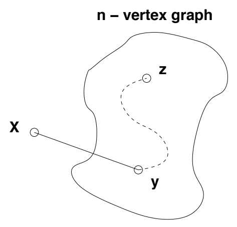
   $$2(n+1) = 2(a+b) = 2a+2b = 0+0 = 0.$$

   So the statement is true. □

   Answer: The proof is incorrect. When $n = 0$, we cannot write $n+1 = 1 = a+b$ where $0 < a,b \le n = 0$.

   (b) **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 $n \ge 1$.
   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 $n \ge 1$.
   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 $n+1$. □

Answer: The mistake is in the argument that *"every $(n+1)$-vertex graph with minimum degree 1 can be obtained from an n-vertex graph with minimum degree 1 by adding 1 more vertex."* Instead of starting by considering an arbitrary $(n+1)$-vertex graph, this proof only considers an $(n+1)$-vertex graph that you can make by starting with an n-vertex graph with minimum degree 1. As a counterexample, consider a graph on four vertices $V = \{1,2,3,4\}$ with two edges $E = \{\{1,2\},\{3,4\}\}$. Every vertex in this graph has degree 1, but there is no way to build this 4-vertex graph from a 3-vertex graph with minimum degree 1.

More generally, this is an example of *build-up error* in proof by induction. Usually this arises from a faulty assumption that every size $n+1$ graph with some property can be "built up" from a size n graph with the same property. (This assumption is correct for some properties, but incorrect for others, such as the one in the argument above.)

One way to avoid an accidental build-up error is to use a *"shrink down, grow back"* process in the inductive step: start with a size $n+1$ graph, remove a vertex (or edge), apply the inductive hypothesis $P(n)$ to the smaller graph, and then add back the vertex (or edge) and argue that $P(n+1)$ holds.

Let's see what would have happened if we'd tried to prove the claim above by this method. In the inductive step, we must show that $P(n)$ implies $P(n+1)$ for all $n \geq 1$. Consider an $(n+1)$-vertex graph G in which every vertex has degree at least 1. Remove an arbitrary vertex v, leaving an n-vertex graph G' in which every vertex has degree... uh-oh!

The reduced graph G' might contain a vertex of degree 0, making the inductive hypothesis $P(n)$ inapplicable! We are stuck —— and properly so, since the claim is false!

2. (14 points: 3/3/4/4) **Trees**

Recall that a **tree** is a connected acyclic graph (graph without cycles). In the note, we presented a few other definitions of a tree, and in this problem, we will prove two fundamental properties of a tree, and derive two definitions of a tree we learn from lecture note based on these properties. Let's start with the properties:

(a) Prove that any pair of vertices in a tree are connected by exactly one (simple) path.

(b) Prove that adding any edge to a tree creates a simple cycle.

Now you will show that if a graph satisfies either of these two properties then it must be a tree:

(c) Prove that if every pair of vertices in a graph are connected by exactly one simple path, then the graph must be a tree.

(d) Prove that if the graph has no simple cycles and has the property that the addition of any single edge (not already in the graph) will create a simple cycle, then the graph is a tree.

Answer:

(a) Pick any pair of vertices x, y. We know there is a path between them since the graph is connected. We will prove that this path is unique by contradiction:

Suppose there are two distinct paths from x to y. At some point (say at vertex a) the paths must diverge, and at some point (say at vertex b) they must reconnect. So by following the first path from a to b and the second path in reverse from b to a we get a cycle. This gives the necessary contradiction.

(b) Pick any pair of vertices x, y not connected by an edge. We prove that adding the edge $\{x,y\}$ will create a simple cycle. From part (a), we know that there is a unique path between x and y. Therefore, adding the edge $\{x,y\}$ creates a simple cycle obtained by following the path from x to y, then following the edge $\{x,y\}$ from y back to x.

(c) Assume we have a graph with the property that there is a unique simple path between every pair of vertices. We will show that the graph is a tree, namely, it is connected and acyclic. First, the graph is connected because every pair of vertices is connected by a path. Moreover, the graph is acyclic because there is a unique path between every pair of vertices. More explicitly, if the graph has a cycle, then for any two vertices x, y in the cycle there are at least two simple paths between them (obtained by going from x to y through the right or left half of the cycle), contradicting the uniqueness of the path. Therefore, we conclude the graph is a tree.

(d) Assume we have a graph with no simple cycles, but adding any edge will create a simple cycle. We will show that the graph is a tree. We know the graph is acyclic because it has no simple cycles. To show the graph is connected, we prove that any pair of vertices x, y are connected by a path. We consider two cases: If $\{x,y\}$ is an edge, then clearly there is a path from x to y. Otherwise, if $\{x,y\}$ is not an edge, then by assumption, adding the edge $\{x,y\}$ will create a simple cycle. This means there is a simple path from x to y obtained by removing the edge $\{x,y\}$ from this cycle. Therefore, we conclude the graph is a tree.

3. (12 points: 6/6) **Networks and Tours**

Please prove or disprove the following claims.

(a) Suppose we have n websites such that for every pair of websites A and B, either A has a link to B or B has a link to A. Prove or disprove that there exists a website that is reachable from every other website by clicking at most 2 links. (*Hint: Induction*)

(b) We have shown in the lecture (or you have read Lecture Note 4) that a connected undirected graph has an Eulerian tour if and only if every vertex has even degree.

Prove or disprove that if a connected graph G on n vertices has exactly $2d$ vertices of odd degree, then there are d walks that *together* cover all the edges of G (i.e., each edge of G occurs in exactly one of the d walks; and each of the walks should not contain any particular edge more than once).

Answer:

(a) We prove this by induction on the number of websites n.

Base case For $n = 2$, there's always a link from one website to the other.

Induction Hypothesis When there are k websites, there exists a website w that is reachable from every other website by clicking at most 2 links.

Induction Step Let A be the set of websites with a link to w, and B be the set of websites two links away from w. The induction hypothesis states that the set of k websites $W = \{w\} \cup A \cup B$. Now suppose we add another website v. Between this website and every website in W, there must be a link from one to the other. If there is at least one link from v to $\{w\} \cup A$, w would still be reachable from v with at most 2 clicks. Otherwise, if all links from $\{w\} \cup A$ point to v, v will be reachable from every website in B with at most 2 clicks, because every website in B can click one link to go to a website in A, then click on one more link to go to v. In either case there exists a website in the new set of $k+1$ websites that is reachable from every other website by clicking at most 2 links.

(b) We split the $2d$ odd-degree vertices into d pairs, and join each pair with an edge, adding d more edges in total. 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 d added edges breaks the tour into d walks covering all the edges in the original graph, with each edge belonging to exactly one walk.

4. (12 points: 1.5/2/3.5/5) **Hypercube routing**

Recall that an n-dimensional hypercube contains 2^n vertices, each labeled with a distinct n bit string, and two vertices are adjacent if and only if their bit strings differ in exactly one position.

(a) The hypercube is a popular architecture for parallel computation. Let each vertex of the hypercube represent a processor and each edge represent a communication link. Suppose we want to send a packet for vertex x to vertex y. Consider the following "bit-fixing" algorithm:

> In each step, the current processor compares its address to the destination address of the packet. Let's say that the two addresses match up to the first k positions. The processor then forwards the packet and the destination address on to its neighboring processor whose address matches the destination address in at least the first $k+1$ positions. This process continues until the packet arrives at its destination.

Consider the following example where $n = 4$: Suppose that the source vertex is (1001) and the destination vertex is (0100). Give the sequence of processors that the packet is forwarded to using the bit-fixing algorithm.

(b) The *Hamming distance* $H(x,y)$ between two n-bit strings x and y is the number of bit positions where they differ. Show that for an arbitrary source vertex and arbitrary destination vertex, the number of edges that the packet must traverse under this algorithm is the Hamming distance between the n-bit strings labeling source and destination vertices.

(c) Consider the following example where $n = 3$: Suppose that x is (110) and y is (011). What is the length of the shortest path between x and y? What is the set of all vertices and the set of all edges that lie on shortest paths between x and y? Do you see a pattern? You do not need to prove your answer here – you'll provide a general proof in part (d).

(d) Answer the last question for an arbitrary pair of vertices x and y in the hypercube. Can you describe the set of vertices and the set of edges that lie on shortest paths between x and y? Prove that your answers are correct. (*Hint:* consider the bits where x and y differ.)

Answer:

(a) The source $x = (1001)$ and $y = (0100)$ differ in three bits, and the bit-fixing algorithm sequentially flips the differing bits from left to right, so the sequence of processors from x to y is:

$$1001 \to 0001 \to 0101 \to 0100.$$

(b) This is an easy induction, since after the first step, the bit-fixing algorithm sends the packet from x to a neighboring vertex x' which is one step closer to y. i.e. $H(x',y) = H(x,y) - 1$. Now by the induction hypothesis (you have to spell out what it is!), the packet must traverse $H(x',y) = H(x,y) - 1$ edges to go from x' to y, for a grand total of $H(x',y) + 1 = H(x,y)$.

(c) The length of the shortest path is 2, and we see there are two paths:

$$110 \to 010 \to 011 \quad \text{and} \quad 110 \to 111 \to 011.$$

Note that the first path is the same path obtained by the bit-fixing algorithm from part (a), where we flip the bits from left to right, while the second path is flipping the bits from right to left. Therefore, the set of all vertices in the shortest paths is

$$\{110, 010, 111, 011\}$$

Here the pattern is that we look at the starting and ending vertices $x = (110)$ and to $y = (011)$, and hold the middle bit where they agree constant, and try all possible patterns for the first and last bits where they disagree. Another way of saying this is that the set of vertices is the 2 dimensional sub cube with middle bit equal to 1.

(d) By part (b), the length of the shortest path between x and y is the Hamming distance $k = H(x,y)$. To achieve this shortest path, we must leave alone all bit positions where x and y agree, and in each step change one of the bit positions where x and y disagree (to look like y). This means that the set of all vertices in shortest paths corresponds exactly to the 2^k n-bit strings which agree with x and y where the two are the same, and have an arbitrary set of k bits in those positions where x and y disagree. In other words, this set of vertices form a k-dimensional subcube of the n-dimensional hypercube, and all edges in this subcube appear in the shortest paths.

5. (7 points: 1/2/2/2) **Bipartite graphs**

An undirected graph is called *bipartite* if its vertices can be partitioned into two disjoint sets L, R such that each edge connects a vertex in L with a vertex in R (i.e., there is no edge connecting two vertices in L or two vertices in R).

(a) Prove that a bipartite graph has no cycles of odd length.

(b) Prove that $\sum_{v \in L} \text{degree}(v) = \sum_{v \in R} \text{degree}(v)$.

(c) Let s denote the average degree of vertices in L and t the average degree of vertices in R, i.e., $s = \frac{1}{|L|} \sum_{v \in L} \text{degree}(v)$ and $t = \frac{1}{|R|} \sum_{v \in R} \text{degree}(v)$. Prove that $s/t = |R|/|L|$.

(d) In 1992, the University of Chicago interviewed a random sample of 2500 people in the U.S. about the number of opposite-gender sex partners they had had. They reported that on average men have 74% more opposite-gender partners than women. At around the same time, the U.S. Census Bureau reported that the female population of the U.S. was about 140 million and the male population was about 134 million. With reference to part c, explain why the University of Chicago and the U.S. Census Bureau can't both be right.

Answer:

(a) Let us start traveling the cycle from a node n_0 in L. Since each edge in the graph connects a vertex in L to one in R, the 1^{st} edge in the set connects our start node n_0 to the a node n_1 in R. The 2^{nd} edge in the cycle must connect n_1 to a node n_2 in L. Continuing on, the $2k+1^{\text{th}}$ edge connects node n_{2k} in L to node n_{2k+1} in R, and the $2k^{\text{th}}$ edge connects node n_{2k-1} in R to node n_{2k} in L. Since only even numbered edges connect to vertices in L, and we started our cycle in L, the cycle must edge with an even number of edges.

(b) Since each edge connects a node in L to one in R, each edge contributes 1 to the total degree in L and 1 to the total degree in R. Since all edges contribute equally to the total degree of all vertices in L and R, the total degree of all vertices in L must be equal to the total degree of all vertices in R.

(c) We just showed that $\sum_{v \in L} \text{degree}(v) = \sum_{v \in R} \text{degree}(v)$. Substituting in s and t, we get $|L| \cdot s = |R| \cdot t$. Manipulating this ratio, we get $s/t = |R|/|L|$.

(d) A graph of opposite-gendered partners is bipartite (that is, each node (person) is either male or female, and each edge (partnership) connects one male to one female). As we showed, the ratio of average degree of male nodes to average degree of female nodes must be equal to the total number of females divided by the total number of males. Clearly, $1.74 \neq \frac{140\text{mil}}{134\text{mil}}$, so it is not possible that both of these statistics are accurate.

6. (12 points: 2/2/3/5) **TA Assignment**

You have been asked to assign TAs for the summer sessions. Each class has its own method for ranking candidates, and each candidate has their own preferences. An assignment is **unstable** if a class and a candidate prefer each other to their current assignments. Otherwise, it is **stable**.

Candidate information:

Candidate	CS61C Grade	CS70 Grade	CS61A Grade	Teaching Experience	Overall GPA	Preferences
A	A+	A	A	Yes	3.80	CS61C > CS70 > CS61A
B	A	A	A	No	3.61	CS61C > CS61A > CS70
C	A	A+	A-	Yes	3.60	CS61C > CS70 > CS61A

Ranking method:

- CS61C: Rank by CS61C grade. Break ties using teaching experience, then overall GPA.
- CS70: Rank by teaching experience. Break ties using CS70 grade, then overall GPA.
- CS61A: Rank by CS61A grade. Break ties using overall GPA, then teaching experience.

(a) Find a stable assignment.

(b) Can you find another, or is there only one stable assignment (if there is only one, why)?

(c) CS61C is overenrolled and needs two TAs. There is another candidate.

Candidate	CS61C Grade	CS70 Grade	CS61A Grade	Teaching Experience	Overall GPA	Preference
D	A+	A	A+	No	3.90	CS70 > CS61A > CS61C

Find a stable assignment.

(d) Prove your assignment in Part (c) is stable.

Answer:

(a) Use SMA with the following class rankings and students proposing.

CS61C	A > C > B
70	C > A > B
61A	A > B > C

Day	CS61C	CS70	CS61A
1	A, B, C		
2	A	C	B

Assignment: (CS61C, A), (CS70, C), (CS61A, B).

(b) Run SMA with classes proposing and get the same assignment.

Day	A	B	C
1	61C,61A		70
2	61C	61A	70

For any stable assignment, if a student is paired with a class C', the student must prefer his/her optimal class to C' and prefer C' to his/her pessimal class. When students proposed, SMA outputs the student optimal assignment. When classes proposed, SMA outputs the class optimal assignment, which is the student pessimal assignment. Because the student optimal assignment is the same as the student pessimal assignment, C' can only be the class in the assignment for any student, and thus there can only be one stable assignment.

(c) Use SMA with students proposing and rule that CS61C can hold 2 proposals, with the following class rankings:

CS61C	A > D > C > B
CS70	C > A > D > B
CS61A	D > A > B > C

Day	CS61C	CS70	CS61A
1	A, B, C	D	
2	A, C	D	B

Assignment: (CS61C, A and C), (CS70, D), (CS61A, B).

(d) Follow the stability proof in the lecture note to prove that the assignment is stable. Suppose some TA T in the assignment prefers some class C^* to their assigned class C. We will argue that C^* prefers their TA(s) to T, so there cannot be a rogue couple (a class and a TA prefer each other to their current assignments). Since C^* occurs before C in T's list, he must have proposed to it before he proposed to C. Therefore, according to the algorithm, C^* must have rejected him for somebody it prefers. If C^* is not CS61C, the Improvement Lemma shows C^* likes its final TA at least as much as T. If C^* is CS61C, then we must now prove an alternate Improvement Lemma to show this.

Prove: If T is rejected by CS61C on the k-th day, then every subsequent day $j \geq k$, CS61C has 2 TAs whom it likes at least as much as T.

- *Base case*: On day k, CS61C rejects T, so it must prefer the two TAs it holds.
- *Induction hypothesis*: Suppose claim is true for $j \geq k$
- *Induction step*: On day $j+1$, by induction hypothesis, CS61C has 2 TAs T' and T'' it prefers to T. Either nobody proposes to CS61C, or T''' proposes. If T''' is accepted, then it must be preferred over T' or T'', which are both at least as good as T, so T''' is preferred over T.

After proving the alternate Improvement Lemma, we can claim C^* likes its final TA at least as much as T. Therefore, no TA T can be involved in a rogue couple, and thus the assignment is stable.

7. (8 points: 3/5) **Long Courtship**

(a) Run the traditional propose-and-reject algorithm on the following example:

Man	Preference List
1	$A > B > C > D$
2	$B > C > A > D$
3	$C > A > B > D$
4	$A > B > C > D$

Woman	Preference List
A	$2 > 3 > 4 > 1$
B	$3 > 4 > 1 > 2$
C	$4 > 1 > 2 > 3$
D	$1 > 2 > 3 > 4$

Answer: The stable pairing reached by male propose-and-reject algorithm is $\{(1,D),(2,A),(3,B),(4,C)\}$.

Woman	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
A	1,④	④	④	4,③	③	③	3,②	②	②	②
B	②	2,①	①	①	1,④	④	④	4,③	③	③
C	③	③	3,②	②	②	2,①	①	①	1,④	④
D										①

(b) We know from the notes that the propose-and-reject algorithm must terminate after at most n^2 proposals. Prove a sharper bound showing that the algorithm must terminate after at most $n(n-1)+1$ proposals. Is this instance a worst-case instance for $n = 4$? How many days does the algorithm take on this instance?

Answer:

If we consider a scenario where every man has proposed to $n-1$ women, then either every woman has received a proposal and thus every woman has accepted a proposal and the algorithm terminates, or there is one woman who has never received a proposal. Since there are n men, the maximum number of possible proposals must be $n(n-1)+1$, where 1 is added to account for the last woman to whom a proposal might be extended on the last day.

In the example above there were $13 = 4(4-1)+1$ proposals in 10 days.

8. (16 points: 8/8) **Better Off Alone**

In the stable marriage problem, suppose that some men and women have standards and would not just settle for anyone. In other words, in addition to the preference orderings they have, they prefer being alone to being with some of the lower-ranked individuals (in their own preference list). A pairing could ultimately have to be partial, i.e., some individuals would remain single.

The notion of stability here should be adjusted a little bit. A pairing is stable if

- there is no paired individual who prefers being single over being with his/her current partner,
- there is no paired man and paired woman that would both prefer to be with each other over their current partners, and
- there is no single man and single woman that would both prefer to be with each other over being single.
- there is no paired man and single woman (or single man and paired woman) that would both prefer to be with each other over the current choice (the current partner or being alone).

(a) Prove that a stable pairing still exists in the case where we allow single individuals. You can approach this by introducing imaginary mates that people "marry" if they are single. How should you adjust the preference lists of people, including those of the newly introduced imaginary ones for this to work?

Answer: Following the hint, we introduce an imaginary mate (let's call it a robot) for each person. Note that we introduce one robot for each individual person, i.e. there are as many robots as there are people. For simplicity let us say each robot is owned by the person we introduce it for.

Each robot is in love with its owner, i.e. it puts its owner at the top of its preference list. The rest of its preference list can be arbitrary. The owner of a robot puts it in his/her preference list exactly after the last person he/she is willing to marry. i.e. owners like their robots more than people they are not willing to marry, but less than people they like to marry. The ordering of people who someone does not like to marry as well as robots he/she does not own is irrelevant as long as they all come after their robot.

To illustrate, consider this simple example: there are three men $1,2,3$ and three women A,B,C. The preference lists for men is given below:

Man	Preference List
1	$A > B$
2	$B > A > C$
3	C

and the following depicts the preference lists for women:

Woman	Preference List
A	1
B	$3 > 2 > 1$
C	$2 > 3 > 1$

In this example, 1 is willing to marry A and B and he likes A better than B, but he'd rather be single than to be with C. On the other side B has a low standard and does not like being single at all. She likes 3 first, then 2, then 1 and if there is no option left she is willing to be forced into singleness. On the other hand, A has pretty high standards. She either marries 1 or remains single.

According to our explanation we should introduce a robot for each person. Let's name the robot owned by person X as R_X. So we introduce male robots R_A, R_B, R_C and female robots R_1, R_2, R_3. Now we should modify the existing preference lists and also introduce the preference lists for robots.

According to our method, 1's preference list should begin with his original preference list, i.e. $A > B$. Then comes the robot owned by 1, i.e. R_1. The rest of the ordering, which should include C and R_2, R_3 does not matter, and can be arbitrary.

For B, the preference list should begin with $3 > 2 > 1$ and continue with R_B, but the ordering between the remaining robots (R_A and R_C) does not matter.

What about robots' preference lists? They should begin with their owners and the rest does not matter. So for example R_A's list should begin with A, but the rest of the humans/robots (B, C, R_1, R_2, and R_3) can come in any arbitrary order.

So the following is a list of preference lists that adhere to our method. There are arbitrary choices which are shown in bold (everything in bold can be reordered within the bold elements).

Man	Preference List
1	$A > B > R_1 > \mathbf{3} > \mathbf{R_3} > \mathbf{R_2}$
2	$B > A > C > R_2 > \mathbf{R_1} > \mathbf{R_3}$
3	$C > R_3 > \mathbf{R_1} > \mathbf{R_3} > \mathbf{A} > \mathbf{B}$
R_A	$A > \mathbf{B} > \mathbf{C} > \mathbf{R_1} > \mathbf{R_2} > \mathbf{R_3}$
R_B	$B > \mathbf{R_1} > \mathbf{R_2} > \mathbf{R_3} > \mathbf{A} > \mathbf{C}$
R_C	$C > \mathbf{A} > \mathbf{R_2} > \mathbf{B} > \mathbf{R_1} > \mathbf{R_3}$

and the following depicts the preference lists for women and female robots:

Woman	Preference List
A	$1 > R_A > \mathbf{3} > \mathbf{R_B} > \mathbf{2} > \mathbf{R_C}$
B	$3 > 2 > 1 > R_B > \mathbf{R_C} > \mathbf{R_A}$
C	$2 > 3 > 1 > R_C > \mathbf{R_A} > \mathbf{R_B}$
R_1	$1 > \mathbf{R_B} > \mathbf{2} > \mathbf{R_C} > \mathbf{3} > \mathbf{R_A}$
R_2	$2 > \mathbf{R_A} > \mathbf{R_C} > \mathbf{1} > \mathbf{3} > \mathbf{R_B}$
R_3	$3 > \mathbf{2} > \mathbf{1} > \mathbf{R_A} > \mathbf{R_C} > \mathbf{R_B}$

Now let us prove that a stable pairing between robots and owners actually corresponds to a stable pairing (with singleness as an option). This will finish the proof, since we know that in the robots and owners case, the propose and reject algorithm will give us a stable matching.

It is obvious that to extract a pairing without robots, we should simply remove all pairs in which there is at least one robot (two robots can marry each other, yes). Then each human who is not matched is declared to be single. It remains to check that this is a stable matching (in the new, modified sense). Before we do that, notice that a person will never be matched with another person's robot, because if that were so he/she and his/her robot would form a rogue couple (the robot's love is there, and the owner actually likes his/her robot more than other robots).

i. No one who is paired would rather break out of his/her pairing and be single. This is because if that were so, that person along with its robot would have formed a rogue couple in the original pairing. Remember, the robot loves its owner more than anything, so if the owner likes it more than his/her mate too, they would be a rogue couple.

ii. There is no rogue couple. If a rogue couple m and w existed, they would also be a rogue couple in the pairing which includes robots. If neither m nor w is single, this is fairly obvious. If one or both of them are single, they prefer the other person over being single, which in the robots scenario means they prefer being with each other over being with their robot(s) which is their actual match.

This shows that each stable pairing in the robots and humans setup gives us a stable pairing in the humans-only setup. It is noteworthy that the reverse direction also works. If there is a stable pairing in the humans-only setup, one can extend it to a pairing for robots and humans setup by first creating pairs of owners who are single and their robots, and then finding an arbitrary stable matching between the unmatched robots (i.e. we exclude everything other than the unmatched robots and find a stable pairing between them). To show why this works, we have to refute the possibility of a rogue pair. There are three cases:

i. A human-human rogue pair. This would also be rogue pair in the humans-only setup. The humans prefer each other over their current matches. If their matches are robots, that translates to them preferring each other over being single in the humans-only setup.

ii. A human-robot rogue pair. If the human is matched to his/her robot, our pair won't be a rogue pair since a human likes his/her robot more than any other robot. On the other hand if the human is matched to another human, he/she prefers being with that human over being single which places that human higher than any robot. Again this refutes the human-robot pair being rogue.

iii. A robot-robot rogue pair. If both robots are matched to other robots, then by our construction, this won't be a rogue couple (we explicitly selected a stable matching between left-alone robots). On the other hand, if either robot is matched to a human, that human is its owner, and obviously a robot loves its owner more than anything, including other robots. So again this cannot be a rogue pair.

This completes the proof.

(b) As you saw in the lecture, we may have different stable pairings. But interestingly, if a person remains single in one stable pairing, s/he must remain single in any other stable pairing as well (there really is no hope for some people!). Prove this fact by contradiction.

Answer: We will perform proof by contradiction. Assume that there exists some man m_1 who is paired with a woman w_1 in stable pairing S and unpaired in stable pairing T. Since S is a stable pairing and m_1 is unpaired, w_1 must be paired in T with a man m_2 whom she prefers over m_1. (If w_1 were unpaired or paired with a man she does not prefer over m_1, then (m_1, w_1) would be a rouge couple, which is a contradiction.)

Since m_2 is paired with w_1 in T, he must be paired in S with some woman w_2 whom m_2 prefers over w_1. This process continues (w_2 must be paired with some m_3 in T, m_3 must be paired with some w_3 in S, etc.) until all persons are paired. Since this requires m_1 to be paired in T, where he is known to be unpaired, we have reached a contradiction. Therefore, our assumption must be false, and there cannot exist some man who is paired in a stable pairing S and unpaired in a stable pairing T. A similar argument can be used for women.
Since no man or woman can be paired in one stable pairing and unpaired in another, every man or woman must be either paired in all stable pairings or unpaired in all stable pairings.

Here is another possible proof:
We know that some male-optimal stable pairing exists. Call this pairing M. We first establish two lemmas.

Lemma 1. If a man is single in male-optimal pairing M, then he is single in all other stabling pairings.
Proof. Assume there exists a man that is single in M but not single in some other stable pairing M'. Then M would not be a male-optimal pairing, so this is a contradiction.

Lemma 2. If a woman is paired in male-optimal pairing M, she is paired in all other stable pairings.
Proof. Assume there exists a woman that is paired in M but single in some other stable pairing M'. Then M would not be female-pessimal, so this is a contradiction.

Let there be k single men in M. Let M' be some other stable pairing. Then by Lemma 1, we know single men in M' will be greater than or equal to k. We also know that there are $n-k$ paired men and women in M. Then by Lemma 2, we know that the number of paired women in M' will be greater than or equal to $n-k$.

Now, we want to prove that if a man is paired in M, then he is paired in every other stable pairing. We prove this by contradiction. Assume that there exists a man m that is paired in M

but is single in some other stable pairing M'. Then there must be strictly greater than k single men in M', and thus strictly greater than k single women in M'. Since there are strictly greater than k single women in M', there must be strictly less than $n-k$ paired women in M'. But this contradicts that the number of paired women in M' will be greater than or equal to $n-k$.

We also have to prove that if a woman is single in M, then she must be single every other stable pairing. We again prove this by contradiction. Assume that there exists a woman w that is single in M and paired in some other stable pairing M'. Then there are strictly greater than $n-k$ paired women in M', which means there are strictly greater than $n-k$ paired men in M'. This means there must be strictly less than k single men in M'. But this contradicts that the number of single men in M' will be greater than or equal to k.

Since we have proved both 1) If a man is single in M then he is single in every other stable pairing and 2) If a man is paired in M then he is paired in every other stable pairing (note that the contrapositive of this is if a man is single in any other stable pairing, then this man is single in M), we know that a man is single in M if and only if he is single in every other stable pairing. Similarly, since we have proved both 1) If a woman is single in M then she is single in every other stable pairing and 2) If a woman is paired in M then she is paired in every other stable pairing, we know that a woman is single in M if and only if she is single in every stable pairing. Thus we have proved that if a person is single in one stable pairing, s/he is single in every stable pairing.

9. (14 points: 7/7) **Karl and Emma fight!**

   (a) Karl and Emma are having a disagreement regarding the traditional propose-and-reject algorithm. They both agree that it favors men over women. But they disagree about what, if anything, can be done without changing the ritual form of men proposing, women rejecting, and people getting married when there are no more rejections.

   Karl mansplains: "It's hopeless. Men are obviously going to propose in the order of their preferences. It's male optimal so why would they do anything else? As far as the women are concerned, given that they face a specific choice of proposals at any given time, they are obviously going to select the suitor they like the most. So unless we smash the system entirely, it is going to keep all women down."

   Emma says: "People are more perceptive and forward-looking that you think. Women talk to each other and know each other's preferences regarding men. They can also figure out the preferences of the men they might be interested in. A smart and confident woman should be able to do better for herself in the long run by not trying to cling to the best man she can get at the moment. By rejecting more strategically, she can simultaneously help out both herself and her friends."

   Is Emma ever right? If it is impossible, prove it. If it is possible, construct and analyze an example (a complete set of people and their preference lists) in which a particular woman acting on her own (by not following the ordering of her preference list when deciding whether to accept or reject among multiple proposals) can get a better match for herself without hurting any other woman. Show how she can do so. The resulting pairing should also be stable.

   Answer: Emma is right. Here is an example of six people, three men $(1,2,3)$ and three women (A,B,C), together with their respective preference lists.

Man	Preference List
1	$A > C > B$
2	$A > B > C$
3	$C > A > B$

Woman	Preference List
A	$3 > 1 > 2$
B	$2 > 3 > 1$
C	$1 > 2 > 3$

We know what happens when we run the traditional propose-and-reject algorithm with these preference lists. We get the pairing $\{(1,A),(2,B),(3,C)\}$. But here, we can see that both A and C can do better.

Now, woman A looks at the preference lists of all the men and women and notices something. If she doesn't cling to the best she can get at the moment, she can end up with someone better! Further, she can do so without hurting B and C. Let's see how this plays out.

On day 1 of the traditional algorithm, we have the following proposals. The only change here is that A now rejects 1 instead of 2 even though she likes 1 more among them.

Day	1
A	1, $\textcircled{2}$
B	
C	$\textcircled{3}$

Now, we continue from as normal, using the traditional propose-and-reject algorithm. Here are the remaining days.

Day	1	2	3	4
A	1, $\textcircled{2}$	$\textcircled{2}$	2, $\textcircled{3}$	$\textcircled{3}$
B				$\textcircled{2}$
C	$\textcircled{3}$	$\textcircled{1}$, 3	$\textcircled{1}$	$\textcircled{1}$

The pairing produced from the run above is $\{(3,A),(2,B),(1,C)\}$. We see that woman A acting on her own on day one changed the face of the game for the women.

Is this pairing stable? Well, of course it is! Each woman ends up with the man she likes the most.

In conclusion, we can say with conviction that Emma was indeed right! A forward-thinking woman can potentially improve the ostensibly bleak outcome of the traditional propose-and-reject algorithm by strategically rejecting in the early stages.

(b) Karl and Emma have another disagreement! Karl claims that if a central authority was running the propose-and-reject algorithm then cheating the system might improve the cheater's chances of getting the more desirable candidate. The cheater need not care about what happens to the others.

Karl says: "Lets say there exists a true preference list. A prefers 1 to 2 but both are low on her preference list. By switching the reported preference order among 1 and 2, she can end up with 3 whom she prefers over 1 and 2 which wasn't possible if she did not lie. Isn't that cool?"

Emma responds: "That's impossible! In the traditional propose-and-reject algorithm switching the preference order 1 and 2 cannot improve A's chance to end up with 3."

Either prove that Emma is right or give an example of set of preference list for which a switch would improve A's husband (that is, she gets matched with 3), and hence proving Karl is right.

Answer: Assume we have three men 1, 2 and 3 and three women A, B, and C with preferences as given in the table below. Row A shows true preferences of A, while in row A' she pretends she prefers 2 over 1.

Man	Preference List
1	$A > C > B$
2	$A > B > C$
3	$C > A > B$

Woman	Preference List
A	$3 > 1 > 2$
A'	$3 > 2 > 1$
B	$1 > 3 > 2$
C	$1 > 3 > 2$

First let us consider one possible execution of the traditional propose-and-reject algorithm with true preference list of A. Then we get the following pairing: $(1,A)$, $(2,B)$ and $(3,C)$.

Woman	Day 1	Day 2
A	$\textcircled{1}$, 2	$\textcircled{1}$
B		$\textcircled{2}$
C	$\textcircled{3}$	$\textcircled{3}$

If the same algorithm is run with the false preference list A' we get $(1,C)$, $(2,B)$ and $(3,A)$.

Woman	Day 1	Day 2	Day 3	Day 4
A	1, $\textcircled{2}$	$\textcircled{2}$	2, $\textcircled{3}$	$\textcircled{3}$
B				$\textcircled{2}$
C	$\textcircled{3}$	3, $\textcircled{1}$	$\textcircled{1}$	$\textcircled{1}$

This way A got the man she wanted most (3) and also helped C get man 1, who is highest on C's preference list without actually intending to help.