Section 1: Stable Matchings and Proofs Workshop

1. Gale–Shapley review

Consider the following lists of preferences:

$$p_1 : r_3 > r_1 > r_2 > r_4$$
$$p_2 : r_2 > r_1 > r_4 > r_3$$
$$p_3 : r_2 > r_3 > r_1 > r_4$$
$$p_4 : r_3 > r_4 > r_1 > r_2$$

$$r_1 : p_4 > p_1 > p_3 > p_2$$
$$r_2 : p_1 > p_3 > p_2 > p_4$$
$$r_3 : p_1 > p_3 > p_4 > p_2$$
$$r_4 : p_3 > p_1 > p_2 > p_4$$

(a) Run the Gale–Shapley algorithm on the instance above, with p_i proposing. When multiple p_i are free to propose, choose the one with the smallest index (e.g., if p_1 and p_2 are both free, have p_1 propose).

(b) Run the Gale–Shapley algorithm again on the instance above, with p_i proposing. When multiple p_i are free to propose, now choose the one with the *largest* index. Do you get the same result?

(c) Run the Gale–Shapley algorithm on the instance above, with r_i proposing. When multiple r_i are free to propose, choose the one with the smallest index. Do you get the same result?

2. The number of stable matchings

In the previous problem, we saw two distinct stable matchings for the same instance (depending on whether the p_i or r_i are the ones to propose). Is it possible to have an instance of the stable matching problem with more than 2 stable matchings? If so, give an instance with at least 3 stable matchings. If not, prove that every instance has at most 2 stable matchings.

Review of graph concepts

- **Degree:** The number of edges connected to a vertex.
- **Path[1]:** A list of vertices $v_1, v_2, \ldots, v_k$ such that each $\{v_i, v_{i+1}\}$ is an edge. ((v_i, v_{i+1}) in a directed graph)
- **Cycle[2]:** A path $v_1, v_2, \ldots, v_k$ with $v_1 = v_k$.
- **Simple path[3]:** A path with all distinct vertices
- **Simple cycle[4]:** A cycle with all distinct vertices, except the first/last.
- **Connected:** There is a path between any two vertices in the graph.
- **Tree:** A connected, acyclic (no cycles) graph.
- **Rooted tree:** A tree with a designated vertex called the *root*. (Note: Words like "parent" and "child" require a root. For non-rooted trees, say "neighbor" to refer to vertices connected by a single edge to the current one.)

[1] Also known as a *walk* by other sources.
[2] Also known as a *closed walk* by other sources.
[3] Also known as a *path* by other sources.
[4] Also known as a *cycle* by other sources.

3. Proof-writing workshop

Attached as an appendix to this handout, there are 4 sample proofs of the following statement:

Every tree with at least 2 vertices has at least 2 vertices of degree 1.

(a) Take a minute to think about the problem yourself. (It's okay if you don't have a proof.)

(b) Read each sample proof. Discuss with people around you:

  (i) Is it correct? (Are there false statements?)

  (ii) Is it complete? (Are there unjustified claims, unused hypotheses, or undefined notation?)

  (iii) Is it concise? (Are there excessive details, unnecessary notations, or irrelevant arguments?)

  (iv) Is it clear? (Are the main ideas obvious or buried? Could stylistic choices like paragraph breaks, diagrams, bullets, etc. be improved? Are there spelling, grammar, or formatting errors?)

  (v) What do you like about the proof? How would you improve this proof?

The following problems will not be covered in section, but may be useful to think about.
We recommend trying them by yourself first. Solutions will be posted in the evening.

4. Find the bug: failed induction

In this problem, you will fix an incorrect induction proof.

Problem: Suppose you have a stable matching instance with n people in P and n people in R. Of the n members of R, 5 are *popular*. That is, every person in P has those 5 members of R as their first 5 choices (in some order, not necessarily the same for each person in P). Similarly, you have 5 popular members of P, such that every person in R has those 5 as their top choices. Prove that in every stable matching of such an instance, every popular person is matched with another popular person.

Spoof. Let $P(n)$ be "In every stable matching of an instance with two groups of size n and 5 popular people per group, every popular person is matched with another popular person." We will show $P(n)$ holds for all $n \geq 5$ by induction on n.

Base case $(n = 5)$**:** With both sets having size 5, every person is popular. Since every stable matching pairs every person, every person is matched to a popular person.

Inductive hypothesis: Suppose $P(n)$ holds for $n = 5, \ldots, k$ for an arbitrary integer $k \geq 5$.

Inductive step: Let $r_1, \ldots, r_k$, $p_1, \ldots, p_k$ be k people in each group, with $r_1, \ldots, r_5, p_1, \ldots, p_5$ being the popular ones. We add r_{k+1} and p_{k+1}. By popularity, r_{k+1} has $p_1, \ldots, p_5$ (in some order) as their 5 favorite people and p_{k+1} has $r_1, \ldots, r_5$ (in some order) as their 5 favorite people. Further, let p_{k+1} and r_{k+1} be each other's 6th choices (i.e. top choice outside the popular people).

Now, consider any stable matching in the old (size k) instance. We create a stable matching for the new instance by pairing r_{k+1} with p_{k+1}. We now show that this matching is stable for the new instance.

Since it was stable for the small instance, the only possible unstable pairs must involve r_{k+1} or p_{k+1}. By IH, every popular person is matched to another popular person. Regardless of where r_{k+1} and p_{k+1} was added to the popular person's list, they fall after the popular ones, so r_{k+1} and p_{k+1} cannot form an unstable pair with the popular people. And since they have each other as their next choices, they cannot form an unstable pair with anyone else. Thus we have that there are no unstable pairs. The popular people remain matched to each other, as required. ☐

(a) There are at least two correctness errors in this proof. Describe them.

(b) Write a correct proof of this claim. Do NOT use induction. Use a proof by contradiction instead.

5. Practice a reduction

A *reduction* from problem A to problem B is a solution to A in which you can call a library function that solves B. Typically, that library function does the bulk of the work, and your solution just consists of some preprocessing of the inputs to A in order to match what B expects, and postprocessing of the output of B to match what A requires. Note that you have no control over how the library function works internally—you only know what input it takes and what output it is guaranteed to give you.

In this question, you will solve a problem by reducing it to the basic stable matching problem.

Problem: Suppose that is a set of r riders and h horses with many more riders than horses; in particular, $2h < r < 3h$. You wish to set up a set of 3 rounds of rides which will give each rider exactly one chance to ride a horse. To keep things fair among the horses, you wish for each to have exactly 2 or 3 rides.

Because it's winter, by the time the third ride starts it will be very dark, so every rider would prefer *any* horse on the first two rides over being on the third ride. Between the first two rides, each rider doesn't have a preference over time of day, and have the same preference over horses. If a rider must be on the third ride, it has the same preference list for that ride as well.

Each horse has a single list over riders, which doesn't change by ride. Since horses love their jobs, they prefer to being one of the horses on the third ride to one of the ones left home.

Design an algorithm which calls the following library **exactly once** and ensures there are no pairs r, h which would both prefer to change the matching and get a better result for themselves.

> `BasicStableMatching`
> **Input:** A set of $2k$ people in two groups of k people each. Each person has an ordered preference list of all k members of the other group.
> **Output:** A stable matching among the $2k$ agents.

(a) Give a 1–2 sentence summary of your idea.

(b) Give the algorithm you're going to run.

(c) Give a 1–2 sentence summary of the idea of your proof.

(d) Write a proof of correctness.

(e) Give the running time of your algorithm, and briefly justify (1–3 sentences).

Appendix — Problem 3 — Sample Solution? 1

Every tree with at least 2 vertices has at least 2 vertices of degree 1.

Proof. Suppose for contradiction that at most 1 vertex has degree 1, so the rest have degree at least 2. Then the sum of the degrees is at least $2n - 1$. However, recall that a tree has $n - 1$ edges, so the sum of degrees should be $2n - 2$, contradiction. □

Appendix — Problem 3 — Sample Solution? 2

Every tree with at least 2 vertices has at least 2 vertices of degree 1.

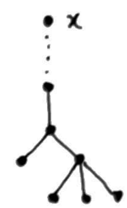
By strong induction on the number of vertices n.

BC: $n=2$. The only tree is , it satisfies the claim.

IS: Assume that every tree with n vertices has 2 vertices of degree 1, for $n = 2, \dots, k$. We prove for $k+1$.

Let T be a tree with $k+1$ vertices and remove a vertex x.

Case 1: $\deg(x) = 1$

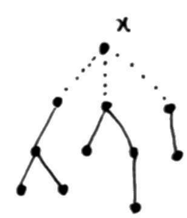


Then removing x cannot disconnect the graph, and removing any vertex will not introduce cycles, so the remaining graph is a tree.

By IH, it has at least 2 vertices of degree 1.

Attaching x may increase the degree of one of them, but $\deg(x) = 1$, so there are still at least 2

Case 2: $\deg(x) \geq 2$



Removing x disconnects the graph into a forest of ≥ 2 trees.

By IH, each tree has at least 2 vertices of degree 1.

Attaching x may increase the degree of one vertex per tree, but there are ≥ 2 trees and hence ≥ 2 remaining vertices of degree 1.

Appendix — Problem 3 — Sample Solution? 3

Every tree with at least 2 vertices has at least 2 vertices of degree 1.

Let P(n) be the statement, “Every tree on n vertices has at least 2 vertices of degree 1.” We will prove P(n) by induction for n >= 2.

Base Case: n=2. There is only one undirected tree with exactly 2 nodes, and it has 2 vertices that are both degree 1.

Inductive Hypothesis: Suppose P(n) is true for n = 2, …, k for an arbitrary k >= 2.

Inductive Step: Let T be an arbitrary tree with k nodes. By inductive hypothesis, T has at least two nodes of degree one. Call them u and v, and create a new node w. Since we are interested in connected trees, we must attach w; we break into cases depending on what it is adjacent to.

Case 1: w is attached to neither u nor v. If w is adjacent to a node other than u, v then u and v still have degree one, so the claim holds on T’.

Case 2: w is attached to one of u, v but not the other. If w is adjacent to u or v, then the other of u, v, and w will both be degree one.

Case 3: w is attached to both u, v. In this case, the graph would be left with no vertices of degree 1, but luckily this case is impossible! If w were connected to both u and v, then the path in T between u and v (which exists because T was connected) along with (u, w) and (v, w) form a cycle, which is not allowed in a tree.

In all (allowed) cases, T’ has the required degree one vertices. Since we constructed T’ to have k + 1 vertices, we have shown P(k+1).

Appendix — Problem 3 — Sample Solution? 4

Every tree with at least 2 vertices has at least 2 vertices of degree 1.

Proof. Let T = (V, E) be an arbitrary tree. Let P be a simple path of maximal length in the tree, so P cannot be extended any longer by definition of maximal. Let $x_1, ..., x_n$ be the vertices in the path, so $\{x_1, x_2\}, \{x2, x_3\}, ..., \{x_n - 1, x_n\} \in E$. Suppose that $deg(x_n) >= 2$. So $\exists y \in V$ such that $y \neq x_n - 1$ and $x_n, y \in E$. If $\exists i = 1, ..., n-2$ such that $y = x_i$, then $x_i, x_i + 1, ..., x_n, x_i = y$ is a cycle, which is a contradiction because trees are always acyclic. If $\forall i = 1, ..., n-2$ we have $y \neq x_i$ then $x_1, \ldots, x_n, y$ is a longer path, which is a contradiction because we said P had maximal length. So now we've covered all the cases and we can conclude that $deg(x_n) < 2$. And $deg(x_n) \neq 0$ because $\{x_n - 1, x_n\}$ is an edge, according to P. So $deg(x_n) = 1$.

Next, suppose that $deg(x_1) >= 2$. So $\exists z \in V$ such that $z \neq x_2$ and $x_1, z \in E$. If $\exists i = 3, ..., n$ such that $z = x_i$, then $x_i = z, x_1, ...x_i$ is a cycle, which is a contradiction because trees are always acyclic. If $\forall i = 3, ..., n$ we have $z \neq x_i$ then $z, x_1, \ldots, x_n$ is a longer path, which is a contradiction because we said P had maximal length. So now we've covered all the cases again and we can conclude that $deg(x_1) < 2$. And $deg(x_1) \neq 0$ because $\{x_1, x_2\}$ is an edge, according to P. So $deg(x_1) = 1$.

Lastly, considering that every tree with at least two vertices contains at least one edge, and the longest simple path P contains at least two distinct vertices, it follows that x1 != xn. So x1 and xn are our two vertices that satisfy the claim, and we conclude that the claim holds. Q.E.D. □