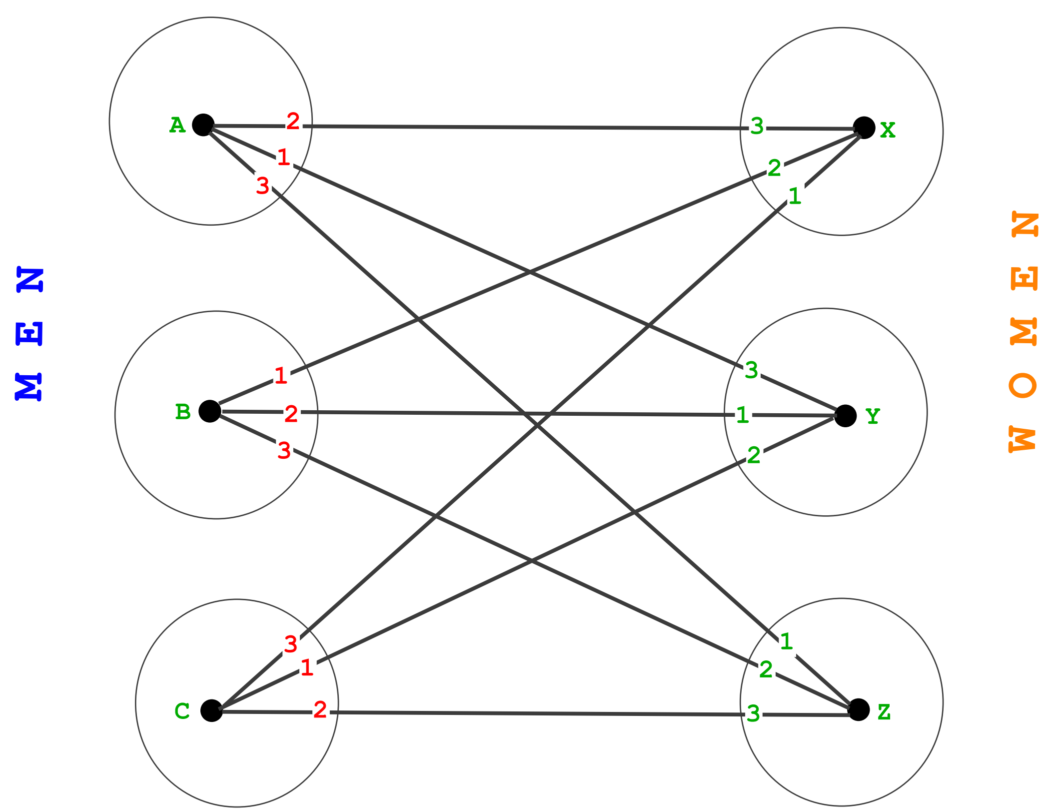
Algorithms & Data — Problem Set 1
(due 11:59pm, Friday January 29th)

Instructions:

- This assignment should be submitted via Blackboard. Late assignments will not be accepted.
- You are encouraged to solve the problems on your own. You are permitted to study with friends and discuss the problems; however, *you must write up your own solutions, in your own words.*
- If you do collaborate with any of the other students on any problem, you must list all the collaborators in your submission for each problem.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class (or the class staff) is strictly prohibited.
- We require that all homework submissions be neat, organized, and *typeset*. You may use plain text or a word processor like Microsoft Word or LaTeX for your submissions. If you need to draw any diagrams, however, you may draw them with your hand.

1. The Gale-Shapley Algorithm (25 pts = 5+5+5+5+5)

Consider the instance of the Stable Matching Problem on the figure below. Men's and women's preferences are indicated by the numerical labels; for example, the man A likes the woman Y the most, while the woman X likes the man C the most.



i. How many matchings (not necessarily stable) between men and women are there?

Solution: A matching could be identified with a 3-element permutation. For example, the matchings
$\{(A, X), (B, Y), (C, Z)\}$, $\{(A, X), (B, Z), (C, Y)\}$, $\{(A, Y), (B, X), (C, Z)\}$
give raise to the permutations

$$(X, Y, Z), (X, Z, Y), (Y, X, Z)$$

(note that when generating all the matchings the first vertices A, B, C never change and can be ignored).
Hence there are $3! = 6$ possible matchings.

---

ii. Consider the matching $\{(A, X), (B, Y), (C, Z)\}$. Is this matching stable? Why or why not?

Solution: This matching is not stable because the edge (B, X) causes instability. We see B likes X more than his current partner (X is 1st on B's preference list, while Y 2nd). X, on the other hand, likes B more than her current partner A (B is 2nd of X's preference list, while A 3rd).

---

iii. How many stable matchings are there? List them.

Solution: There are two stable matchings:
$W = \{(A, Z), (B, Y), (C, X)\}$ and $M = \{(A, Z), (B, X), (C, Y)\}$

---

iv. What is the matching returned by Gale-Shapley algorithm?

Solution: One quick way to find which of the two stable matchings is the one returned by Gale-Shapley algorithm is to see which of them simultaneously provides the best possible partners to all men. The second matching M defined above has the property that for all the men the rankings of the women they are paired with are $\geq$ than those in the matching W. Hence M is the matching returned by G-S algorithm.

---

v. What is the matching returned by the version of Gale-Shapley algorithm in which women play the role of men (i.e., women propose).

Solution: The matching returned by this version of Gale-Shapley algorithm is the one in which women are paired with the best possible men. Clearly, the matching W defined above provides the best men to the women because all of the women get the men they prefer the most.

---

2. Single Stable Matching (15 pts = 5+10)

i. True or False? Suppose that in an instance I of the Stable Matching Problem there is a man m and a woman w who like each other the most. Then the pair (m, w) must belong to every stable matching in this instance. Explain your answer.

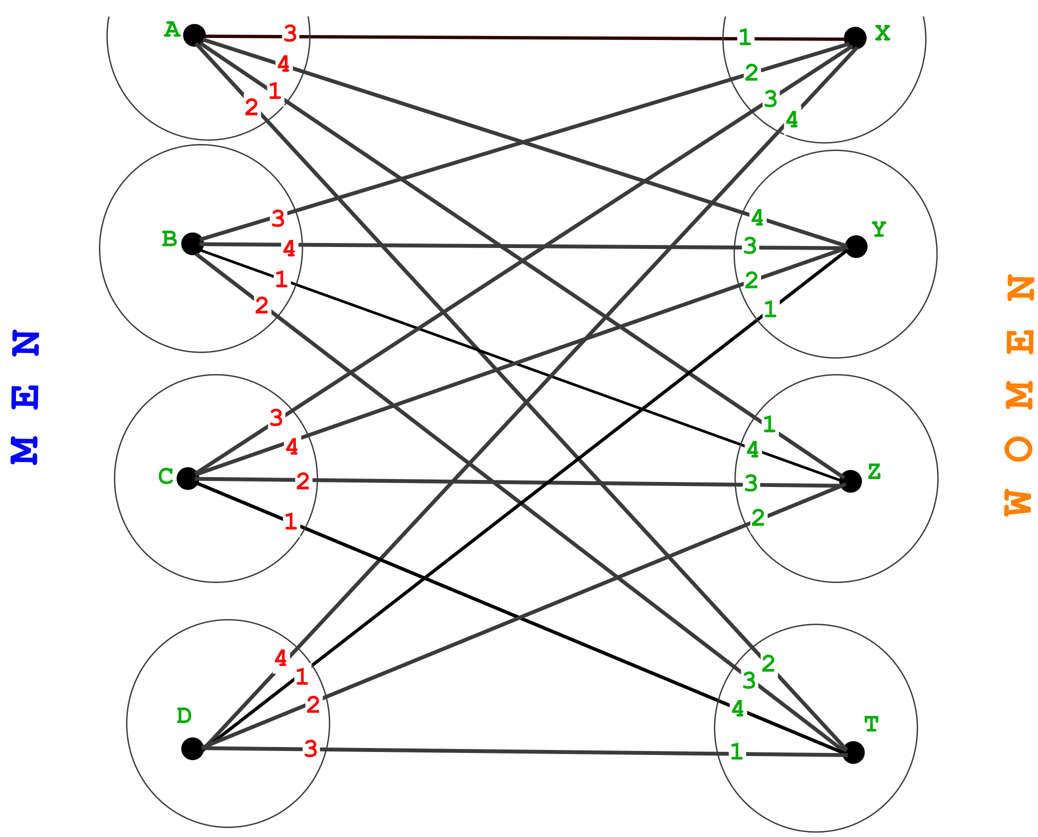
Solution: True. If m and w like each other the most and the pair (m, w) is not in a matching M, then M is guaranteed to be unstable. Indeed, the edge (m, w) is causing the instability:

a) whoever it is that m is paired with must have a lower rank than w because m likes w the most.
b) similarly, whoever it is that w is paired with must be lower on w's preference list than m because w likes m the most.

Hence if the pair (m, w) is not in M, then m and w like each more than the partners they are paired with in M $\implies$ M cannot be stable.

---

ii. Explain why there is only one stable matching in the below instance of the Stable Matching Problem. List all of that matching's edges.



Solution: Notice that the edges (A, Z) and (D, Y) form pairs in which both the man and the woman like each other the most. By part i. both of those edges must belong to every stable matching. There are only two ways to complete the partial matching $M = \{(A, Z), (D, Y)\}$; consider the first of them

$$M_1 = \{(A, Z), (D, Y), (B, X), (C, T)\}$$

The edge (B, X) causes instability in M_1 so M_1 is not stable. That means that the only remaining candidate matching

$$M_2 = \{(A, Z), (D, Y), (B, T), (C, X)\}$$

must be stable (b/c there's always a stable matching).

---

3. Stable Matching Examples (20 pts = 5+5+10)

i. How many instances of the Stable Matching Problem with 3 men and 3 women are there? Assume that we have three fixed distinguishable men and three fixed distinguishable women, say, A, B, C and X, Y, Z respectively, and that an "instance" of the SM Problem is defined as a six-touple

$$(A, L_A), (B, L_B), (C, L_C),\ (X, L_X), (Y, L_Y), (Z, L_Z)$$

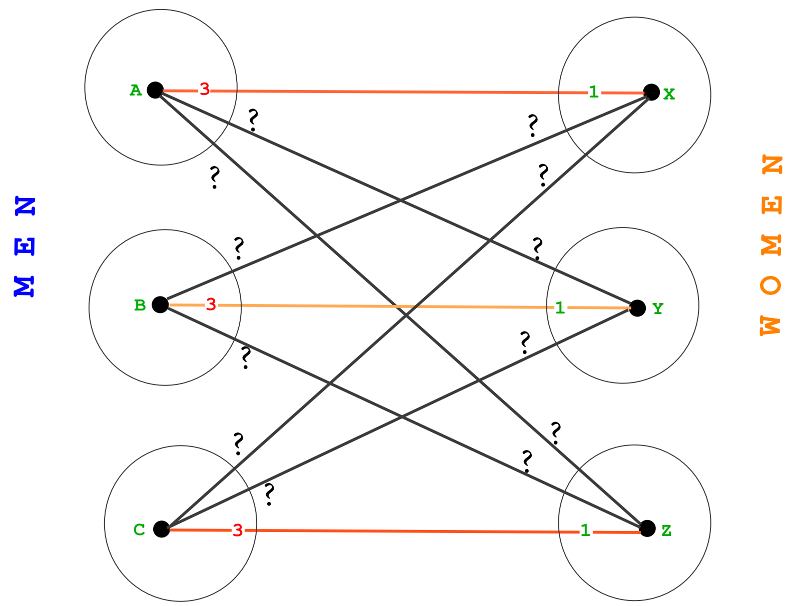
where (A, L_A) signifies the fact that the man A has the preference list L_A, (B, L_B) — that the man B has the preference list L_B and so on.

Solution: Each instance of the SM Problem can be identified with an (ordered) sequence of 6 preference lists. Each preference list is a permutation of numbers $\{1, 2, 3\}$. Since we have 6 independent permutations, there are altogether $(3!)^6 = 6^6 = 46{,}656$ instances of the Stable Matching Problem for 3 women and 3 men.

---

ii. How many instances of the Stable Matching Problem with 3 men and 3 women are there having a stable matching in which every man is paired with his worst woman and every woman with their best man (assuming the instance definition from part i.)?

Solution: Looking at the below figure, we see that the question is the following: given an initial matching M (one of the possible 6 is marked in orange color) in which every man is matched with his worst woman and every woman with her best man, in how many ways can we assign numerical labels to the remaining edges so that we obtain 6 valid preference lists and the matching M is stable?



Fortunately, since the women are paired with their best men, any valid completion of preference lists is not going to spoil the stability of M. This is because any edge leaving vertices X, Y, Z will have its associated preference greater than 1, and cannot be, therefore, unstable. Hence for a given matching M (as above) we have $(2!)^6$ completions, and the 6 initial matchings result in the answer

$$6 \cdot 2^6 = 384$$

---

iii. Let I be an instance of the Stable Matching Problem with men set M, women set W ($|M| = |W| = n$), and their preference lists. Let m_1, m_2 be two different men $\in M$, w_1 the woman of highest rank on m_1's preference list which is matched with m_1 in some stable matching $\in I$, w_2 the woman of highest rank on m_2's preference list which is matched with m_2 in some stable (perhaps different) matching $\in I$. Show that

$w_1 \neq w_2$.

Solution: We will use the following fact which is proved in the textbook:

> The Gale-Shapley algorithm returns a matching S with the following property: every man m is paired with a woman w such that if $\tilde{S}$ is another stable matching and m is paired with a woman w' in $\tilde{S}$, then $rank(w) \geq rank(w')$.

In other words, Gale-Shapley algorithm pairs every man with the best possible woman available in all possible stable matchings.

Returning to our problem, we are told that in the pair (m_1, w_1), w_1 is the best possible woman for m_1 over all stable matchings. Similarly, in the pair (m_2, w_2), w_2 is the best possible woman for m_2 over all stable matchings. By the fact we quoted, both of those pairs are included in the matching S returned by the Gale-Shapley algorithm. Hence w_1 must be different w_2 as otherwise S would not be a matching.

---

4. Linear Algorithm (15 pts =5+10)

Suppose that a number S and a sorted array $a_1 < a_2 < \cdots < a_{n-1} < a_n$ of distinct numbers are given.

i. What is the running time of the "brute force" algorithm to determine whether $a_i + a_j = S$ for some $1 \leq i < j \leq n$?

Solution: The "brute force" algorithm examines all possible $\binom{n}{2}$ pairs of numbers and checks whether any of them sums up to S. The running time is proportional to $\binom{n}{2} = O(n^2)$

---

ii. Give an algorithm for the above problem which runs in $O(n)$ time.

Solution: The trick is to create a new array $B = (S - a_1, S - a_2, \ldots, S - a_n)$ and then check whether B and the original array $A = (a_1, a_2, \ldots, a_n)$ contain any common elements. A common element in the list B is of the form $S - a_i$, while in the list A simply a_j. Those elements being the same means

$$S - a_i = a_j \implies a_i + a_j = S$$

How do we find common elements in arrays A and B? We simply merge the two arrays taking note whenever we encounter the same numbers (we can use the merge because both lists are sorted).

Since creating the array B takes $O(n)$ time, and so does the merging process, the running time of this algorithm is $O(n)$

---

5. Growth of functions (25 pts = 5+5+5+10)

For each of these parts, indicate whether $f = O(g)$, $f = \Omega(g)$, or both (i.e., $f = \Theta(g)$). In each case, give a brief justification for your answer. (*Hint:* It may help to plot the functions and obtain an estimate of their relative growth rates. In some cases, it may also help to express each function as a power of 2 and then compare.)

(a) $f(n) = n^{1.01}$; $g(n) = n(\log n)^2$.

Solution: According to a theorem in the textbook

$$\log_b n = O(n^x)$$

for all $b > 1$ and $x > 0$. In particular, for sufficiently large n we can write

$$\log n \leq n^{0.005} \implies (\log n)^2 \leq n^{0.01} \implies n(\log n)^2 \leq n^{1.01}$$

This means that $n^{1.01} = \Omega(n(\log n)^2)$.
We could perform this argument and show that $n(\log n)^2 \leq n^{1.001}$. Hence we cannot have $n^{1.01} = O(n(\log n)^2)$.

(b) $f(n) = n^2/\log n$; $g(n) = n(\log n)^2$.

Solution: Let $\epsilon > 0$ (in the first part of the argument take $\epsilon = 0$). For $n > 1$, the inequality

$$n^2/\log n \geq n(\log n)^{2+\epsilon}$$

is equivalent to

$$n^2 \geq n(\log n)^{3+\epsilon} \iff n \geq (\log n)^{3+\epsilon}$$

For sufficiently large n the last inequality is true for every $\epsilon \geq 0$ because of the theorem mentioned in part a). This means that the first inequality is true as well and we have $n^2/\log n = \Omega(n(\log n)^{2+\epsilon})$.

For $\epsilon = 0$ this last bound gives $f(n) = \Omega(g(n))$. Since the bound is true for $\epsilon > 0$ we cannot have $f(n) = O(g(n))$

(c) $f(n) = (\log n)^{\log n}$; $g(n) = 2^{(\log n)^2}$.

Solution: Since we have $\log n = 2^{\log(\log n)}$, we are asked to compare

$$(2^{\log(\log n)})^{\log n} = 2^{\log(\log n)\log n} \quad \text{vs.} \quad 2^{(\log n)^2}$$

which is equivalent to comparing the exponents

$$\log(\log n)\log n \quad \text{vs.} \quad (\log n)^2$$

which, in turn, is equivalent to comparing

$$\log(\log n) \quad \text{vs.} \quad \log n$$

Since we obviously have $\log(\log n) = O(\log n)$ and do not have $\log(\log n) = \Omega(\log n)$ we can only write

$$(\log n)^{\log n} = O(2^{(\log n)^2}) \text{ i.e., } f(n) = O(g(n))$$

(d) $f(n) = \sum_{i=1}^{n} i^k$; $g(n) = n^{k+1}$.

Solution: The answer is a consequence of a fact that you should know: for $k, n \geq 1$

$$\sum_{i=1}^{n} i^k = P(n)$$

where P is a polynomial of degree k+1. For example,

$$\begin{aligned} \textstyle\sum_{i=1}^{n} i &= \tfrac{n(n+1)}{2} \\ \textstyle\sum_{i=1}^{n} i^2 &= \tfrac{n(n+1)(2n+1)}{6} \\ \textstyle\sum_{i=1}^{n} i^3 &= (\tfrac{n(n+1)}{2})^2 \end{aligned}$$

One intuitive argument explaining this fact is that the sequence

$$S(n) = 1^k + 2^k + \ldots + n^k$$

has the property that for $n \geq 2$ its first difference

$$\Delta(S(n)) = S(n) - S(n-1) = n^k$$

is a polynomial of k-th degree. The difference operator Δ behaves similarly to the derivative in calculus, and knowing that $\Delta(S(n))$ is a polynomial of k-th degree implies that $S(n)$ is a polynomial of $k+1$-st degree (compare that to $f'(x) = x^k \implies f(x) = 1/(k+1)x^{k+1}$). For detailed proofs of this theorem (called the Faulhaber Formula) go to Wikipedia. Hence $\sum_{i=1}^{n} i^k = \Theta(n^{k+1})$.