Name: _____ **PID:** _____

Definition. Suppose there are n students $x_1, , x_n$ that want to do internship at n companies $y_1, , y_n$. The students send out the applications to all companies, and each student gets to rank the companies in terms of preference. In return, each company also reviews and ranks all applicants.
We want to pair up the students with the company (one student per company) such that **there is no student and company that would both prefer to work with each other rather than their assigned pairs**. This is called a **stable matching**.

Example 1. 3 students: Peter Parker, Scott Lang, and Clark Kent, apply to three companies: Stark Ind., Pym Lab, H.Y.D.R.A. Here are there preferences:

	1*st choice*	2*nd choice*	3*rd choice*
Peter	*Stark Ind.*	*Pym Lab*	*H.Y.D.R.A.*
Scott	*Pym Lab*	*Stark Ind.*	*H.Y.D.R.A.*
Clark	*Stark Ind.*	*Pym Lab*	*H.Y.D.R.A.*

	1*st choice*	2*nd choice*	3*rd choice*
Stark Ind.	*Scott*	*Peter*	*Clark*
Pym Lab	*Peter*	*Scott*	*Clark*
H.Y.D.R.A	*Scott*	*Peter*	*Clark*

Decide whether the following matchings are stable. Explain.

(A) Clark - Stark Ind.; Peter - Pym Lab; Scott - H.Y.D.R.A

(B) Clark - H.Y.D.R.A; Peter - Pym Lab; Scott - Stark Ind.

Gale-Shapley Algorithm for finding stable matchings:

1. Every student applies to the top company choice in their preference list.
2. Each company that receives at least one application picks the top candidate in their list and rejects the rest. Companies with no applicant do nothing.
3. If no student is rejected, stop. We have found a stable matching. Otherwise, the rejected students cross the name of the company who rejected them off their list and then propose to their next favorite among those remaining. Students who were not rejected do nothing.
4. Return to step 2.

Example 2. Use Gale-Shapley's algorithm to find a stable matching in the following preference lists:

A	2	3	1
B	1	3	2
C	1	2	3

1	A	B	C
2	B	C	A
3	B	A	C

Suppose we want to build a train system to connect a group of cities so that the citizens can travel between cities either directly or indirectly. Our goal is to do this for as low a cost as possible.

Greedy solution:

1. Look at all the potential track segments in non-decreasing order of cost (you decide tie-breaker).
2. Start with the segment with the smallest cost and begin to build the railroad system from this segment.
3. Add the next available choice that does not form a cycle to the construction of the railroad system. If the addition of the next lowest cost segment forms a cycle, do not use it.
4. Continue with step 3 until no more segment can be added

Example 3. Build the lowest cost railroad system for the following maps: