CS 312: Algorithms **Fall 2018**

Homework 1

Your Name: ____________________________

Collaborators and sources: ____

You may work in groups, but you must write solutions yourself. List collaborators on your submission.

If you are asked to design an algorithm, please provide: (a) either pseudocode or a precise English description of the algorithm, (b) an explanation of the intuition for the algorithm, (c) a proof of correctness, (d) the running time of your algorithm and (e) justification for your running time analysis.

Submission instructions. This assignment is due by noon on Thursday, Sep 20 in Gradescope (as a pdf file). Please review the course policy about Gradescope submissions to learn how to submit a high-quality pdf.

1. **(5 points) Gradescope submission**
   - The solutions are either typed or written neatly (with ample white-space and no scratching out, etc.).
   - The submission is a pdf.
   - The **Gradescope scanning recommendations** (see their website, which will recommend specific scanning apps) are followed to ensure the scan is high quality.
   - The pages are marked correctly during the gradescope submission.

2. **(10 points) Stable Matching.** *(work independently)* Use the Propose-And-Reject algorithm (p. 6 of the text) to find a stable matching for the following set of four colleges, four students, and their preference lists.

College	Preference list
A	1, 3, 4, 2
B	1, 2, 3, 4
C	3, 1, 2, 4
D	2, 1, 4, 3

Student	Preference list
1	B, C, D, A
2	A, B, C, D
3	C, D, A, B
4	B, D, A, C

3. **(10 points) Stable Matchings: K&T Ch 1, Ex 2.** *(work independently)* Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

   True or false? Consider an instance of the Stable Matching Problem in which there exists a college c and a student s such that c is ranked first on the preference list of s and s is ranked first on the preference list of c. Then in every stable matching M for this instance, the pair (c, s) belongs to M.

4. **(20 points) Stable Matchings: K&T Ch 1, Ex 5.** Consider a version of the stable matching problem where there are n students and n colleges as before. Assume each student ranks the colleges (and vice versa), but now we allow ties in the ranking. In other words, we could have a school that is indifferent two students s_1 and s_2, but prefers either of them over some other student s_3 (and vice versa). We say a student s *prefers* college c_1 to c_2 if c_1 is ranked higher on the s's preference list and c_1 and c_2 are not tied.

(a) **Strong Instability.** A strong instability in a matching is a student-college pair, each of which prefer each other to their current pairing. In other words, neither is indifferent about the switch. Does there always exist a matching with no strong instability? Either give an example of a set of colleges and students with preference lists for which every perfect matchings has a strong instability; or give an algorithm that is guaranteed to find a matching with no strong instability and prove that it is correct.

(b) **Weak Instability.** A weak instability in a matching is a student-college pair where one party prefers the other, and the other may be indifferent. Formally, a student s and a college c with pairs c' and s' form a weak instability if either

- s prefers c to c' and c either prefers s to s' or is indifferent between s and s'.
- c prefers s to s' and s either prefers c to c' or is indifferent between c and c'.

In other words, the pairing between c and s is either preferred by both, or preferred by one while the other is indifferent. Does there always exist a perfect matching with no weak instability? Either give an example of a set of colleges and students with preference lists for which every perfect matching has a weak instability; or give an algorithm that is guaranteed to find a perfect matching with no weak instability.

5. **(20 points) Stable Matching Running Time.** In class, we saw that the Propose-and-reject algorithm terminates in at most n^2 rounds when there are n students and n colleges.

(a) It seems possible that the algorithm may complete in fewer rounds if the preference lists have a certain structure. Describe a family of preference lists, one for each value of n, such that the propose-and-reject algorithm will complete in only $O(n)$ rounds when run on these preference lists.

More specifically, your solution should do the following: for an arbitrary positive integer n, give a precise description of preference lists for n colleges and n students. Let $T(n)$ be the number of rounds that the propose-and-reject algorithm takes for these preference lists. Prove that $T(n)$ is $O(n)$.

(b) Could it be the case that the running time is actually $O(n)$ for all preference lists? Show that this is not true by designing preference lists so that the number of rounds of the algorithm is $\Omega(n^2)$. (If the definition of $\Omega(\cdot)$ has not yet been covered, it will be on Monday. Wait until then to do this part of the problem.)

6. **(0 points).** How long did it take you to complete this assignment?