CSE 421 Intro to Algorithms Winter 2020

Homework 1

Due Friday January 15, 6:00 pm PST

Problem 0: Read the Grading & Academic Integrity Guidelines on the course website. Note in particular that, though you are allowed and encouraged to discuss the problems with others in the class, you are not allowed to consult outside resources (including prior offerings of this course) for solutions; each write-up must be done individually and must not be shared; all others with whom you discussed the solution must be noted on your solution.

Problem 1: In the Stable Matching problem with the following two preference lists

	1st	**2nd**	**3rd**	**4th**
W	**A**	**C**	**B**	**D**
X	**B**	**A**	**D**	**C**
Y	**D**	**A**	**C**	**B**
Z	**B**	**C**	**A**	**D**

and

	1st	**2nd**	**3rd**	**4th**
A	**Y**	**W**	**X**	**Z**
B	**W**	**Y**	**Z**	**X**
C	**W**	**Y**	**Z**	**X**
D	**Z**	**W**	**X**	**Y**

a. What is the best valid partner for each member of both groups?
b. What is the worst valid partner for each member of both groups?
c. Explain how you got your answers to parts (a) and (b) and why they are correct.

Problem 2: In the *Unique Stable Matching Problem*, one is given two equal-sized groups with ordered preference lists for the other group as in the ordinary stable matching problem and the goal is to determine whether or not the stable matching (which we know must exist) is unique. This requires a simple YES/NO answer. Give an efficient algorithm for the Unique Stable Matching Problem.

(Recall from the grading guidelines that, as always when you are asked for an efficient solution, you need to prove that your solution is correct and justify that claim of efficiency with a runtime analysis.)

Problem 3: In the Stable Matching problem, we assumed that everyone from each group could be matched with anyone from the other group. Suppose that, instead, each participant has a ranked ordering of the members of the other group whom they find acceptable, and that they designate the remainder as "unacceptable". (They would rather be unmatched than paired with an unacceptable partner.) The requirements are now that:

- Nobody is matched with a partner whom they find unacceptable
- There are no *unstable** pairs, where an unstable* pair is one in which both sides of the pair prefer being paired with each other than being in their current state (matched with someone they like less, as in ordinary unstable pairs, or simply unmatched).

Let's call the problem of finding a solution satisfying both these properties the *Stable Matching with Unacceptable Partners* problem. Show that there is a solution to this new problem and give an efficient algorithm to find it. (Recall from the grading guidelines that, as always when you are asked for an efficient solution, you need to prove that your solution is correct and justify that claim of efficiency with a runtime analysis.)

Problem 4 (Extra Credit): You're working with a consumer testing company doing some stress-testing on various models of a consumer product to determine the height from which they can be dropped and still not break. The setup for this experiment, for a particular manufacturer's model, is as follows:

You have a ladder with n rungs, and you want to find the highest rung from which you can drop a particular model and not have it break. We call it the safety-height for this model.

If you wanted to reduce the number of trials to a minimum, it might be natural to try binary search: drop the model from the middle rung, see if it breaks, and then recursively try from rung $n/4$ or $3n/4$ depending on the outcome. This would require $\log n$ drops. But this has the drawback that your testing company has to pay for each copy of the model in order to continue to perform the tests after each one breaks.

If your primary goal were to keep the number of copies of the model you need to a minimum, you could instead simply start at the 1st rung, then the 2nd, 3rd, and so on, until the model breaks. This only requires 1 copy but could take up to n drops, each of which is tedious to perform and record.

The general question here is what the optimal trade-off is between the best choices for the # of drops and for a given fixed # k of copies of the model.

a. Describe the best strategy that you can think of that achieves $o(n)$ drops given a budget for 2 copies of the model.
b. Give the exact number of drops your solution requires for $n = 21$ and $n = 22$.
c. Give the asymptotic rate of growth of the worst-case number of drops for your solution from part (a) as a function of n.

Problem 5 (Extra Credit): For the previous problem, give the precisely best possible number of drops as a function of n, and prove that it is impossible to do better.