CSE421: Design and Analysis of Algorithms October 1, 2021

Homework 1

Anup Rao Due: October 8, 2021

Read the fine print[1]. Each problem is worth 10 points:

1. If c is the first company on the applicant a's preference list and a is the first applicant on c's preference list, does it have to be the case that c and a must be matched to each other in every stable matching?

2. Consider the following preference lists.

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

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

For each of A, B, C, D, give the names of their most preferable *valid* partners. Explain how you computed your solution, and why it must be correct.

3. Arrange the following in increasing order of asymptotic growth rate. For full credit it is enough to just give the order.

(a) $f_1(n) = n^4/100$

(b) $f_2(n) = n^3/20$

(c) $f_3(n) = 2^{3\sqrt{n}}$

(d) $f_4(n) = n(\log n)^{1000}$

(e) $f_5(n) = 2^{n \log n}$

(f) $f_6(n) = 2^{(\log n)^{0.9}}$

[1] In solving the problem sets, you are allowed to collaborate with fellow students taking the class, but **each submission can have at most one author**. If you do collaborate in any way, you must acknowledge, for each problem, the people you worked with on that problem. The problems have been carefully chosen for their pedagogical value, and hence might be similar to those given in past offerings of this course at UW, or similar to other courses at other schools. Using any pre-existing solutions from these sources, for from the web, constitutes a violation of the academic integrity you are expected to exemplify, and is strictly prohibited. Most of the problems only require one or two key ideas for their solution. It will help you a lot to spell out these main ideas so that you can get most of the credit for a problem even if you err on the finer details. Please justify all answers. Some other guidelines for writing good solutions are here: `http://www.cs.washington.edu/education/courses/cse421/08wi/guidelines.pdf`.

4. Prove or disprove: In every stable matching, there must be at least one party who is matched to his/her top choice. HINT: Play with some examples with $n = 3$. Note that the problem asks about *every* stable matching, rather than the unique matching that is found by the algorithm discussed in class. Thus if you wish to prove the statement true, you need to prove it for every stable matching, whereas to show that it is false, you just need to find some stable matching that disproves it.