CSE421: Design and Analysis of Algorithms March 30th, 2021

Homework 1

Shayan Oveis Gharan Due: April 8th, 2021 at 11:59 PM

Please see https://courses.cs.washington.edu/courses/cse421/21sp/grading.html for general guidelines about Homework problems.

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.

P1) Consider the following instance of the stable matching problem. For every company $c_1, \ldots, c_4$ and every applicant $a_1, \ldots, a_4$ what is their best valid partner? Justify your claim.

	1st	2nd	3rd	4th
c_1	a_3	a_4	a_2	a_1
c_2	a_2	a_1	a_3	a_4
c_3	a_3	a_4	a_1	a_2
c_4	a_2	a_1	a_4	a_3

	1st	2nd	3rd	4th
a_1	c_2	c_4	c_1	c_3
a_2	c_1	c_3	c_4	c_2
a_3	c_2	c_4	c_3	c_1
a_4	c_4	c_3	c_1	c_2

P2) Prove that in every instance of stable matching problem with n companies and n applicants, companies make at most $n(n-1)+1$ proposals.

Hint: Prove that in the Gale-Shapley algorithm (when companies propose) at most one company gets its last choice.

P3) Suppose the preference lists of all companies in the stable matching problem is equal to $a_1 > a_2 > \cdots > a_n$. Use induction to prove that in this case there is a unique stable matching.

P4) Arrange the following in increasing order of asymptotic growth rate. For full credit it is enough to just give the order. All logs are in base 2.

(a) $f_1(n) = 2^{2\sqrt{\log n}}$

(b) $f_2(n) = \frac{n}{2^{\sqrt{\log \log n}}}$

(c) $f_3(n) = \frac{n(\log \log n)^{99}}{(\log n)^{99}}$

(d) $f_4(n) = n!^2$

(e) $f_5(n) = 2^{\left(4^{\log n}\right)}$

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

(g) $f_7(n) = 2^{\log^{1/3} n}$

(h) $f_8(n) = 2^{\frac{\log n}{\log \log n}}$

(i) $f_9(n) = 2^{\log n - \log \log n}$

(j) $f_{10}(n) = \left(4^2\right)^{\log n}$

P5) **Optional (0 points)** Gale and Shapley published their paper on the stable marriage problem in 1962; but a version of their algorithm had already been in use for ten years by the National Resident Matching Program, for the problem of assigning medical residents to hospitals.

Basically, the situation was the following. There were m hospitals, each with a certain number of available positions for hiring residents. There were n medical students graduating in a given year, each interested in joining one of the hospitals. Each hospital had a ranking of the students in order of preference, and each student had a ranking of the hospitals in order of preference. We will assume that there were more students graduating than there were slots available in the m hospitals.

The interest, naturally, was in finding a way of assigning each student to at most one hospital, in such a way that all available positions in all hospitals were filled. (Since we are assuming a surplus of students, there would be some students who do not get assigned to any hospital.) We say that an assignment of students to hospitals is stable if neither of the following situations arises.

- **First type of instability:** There are students s and s', and a hospital h, so that s is assigned to h, and s' is assigned to no hospital, and h prefers s' to s.
- **Second type of instability:** There are students s and s', and hospitals h and h', so that
  - s is assigned to h, and
  - s' is assigned to h', and
  - h prefers s' to s, and s' prefers h to h'.

So we basically have the stable marriage problem from class, except that (i) hospitals generally want more than one resident, and (ii) there is a surplus of medical students.

Show that there is always a stable assignment of students to hospitals, and give an polynomial-time algorithm to find one.

P6) **Extra Credit:** Design an algorithm that outputs all stable matchings of a given instance with n men and n women such that your algorithm runs in time polynomial in n and the number of stable matchings of the input instance.