CSE421: Design and Analysis of Algorithms	March 27th, 2024
Homework 1	
Shayan Oveis Gharan	Due: April 3rd, 2024 at 11:59 PM

Please see https://courses.cs.washington.edu/courses/cse421/24sp/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) Prove or disprove:

a) For any instance of the stable matching problem the following holds:. In **every** stable matching, there is an **agent** (a company or an applicant) who gets their first (most favorite) choice?

Hint: If correct, you need to give a proof (be sure to show the claim for every instance and for every stable matching!). If incorrect, give a counter-example, i.e., an instance and a stable matching (be sure to justify that the matching is stable!). In this case just one counter example is enough.

b) For every instance of the stable matching problem the following holds: There **exists** a stable matching in which a company gets their first (most favorite) choice.

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 in the Gale-Shapley algorithm.

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

P3) Use induction to solve this problem: Given 2^k real numbers $x_1, \ldots, x_{2^k}$ such that $\sum_{i=1}^{2^k} x_i = 1$. Show that

$$\sum_i x_i^2 \geq \frac{1}{2^k}.$$

Hint: You can use the following inequality: For any two real numbers a, b

$$(a+b)^2 \leq 2a^2 + 2b^2$$

We prove by induction.

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

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

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

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

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

(g) $f_7(n) = \log(n!)$

(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) **Extra Credit:** Design a polynomial time algorithm that outputs a 100 stable matchings of a given instance with n companies and n applicants. If the instance does not have 100 stable matchings output "Impossible".