

CS 330, Fall 2020, Homework 2

Due Wednesday, September 16, 2020, 11:59 pm Eastern Time

Homework Guidelines

Collaboration policy Collaboration on homework problems, with the exception of programming assignments and reading quizzes, is permitted, but not encouraged. If you choose to collaborate on some problems, you are allowed to discuss each problem with at most 5 other students currently enrolled in the class. Before working with others on a problem, you should think about it yourself for at least 45 minutes. Finding answers to problems on the Web or from other outside sources (these include anyone not enrolled in the class) is strictly forbidden.

You must write up each problem solution by yourself without assistance, even if you collaborate with others to solve the problem. You must also identify your collaborators. If you did not work with anyone, you should write "Collaborators: none." It is a violation of this policy to submit a problem solution that you cannot orally explain to an instructor or TA.

Solution guidelines For problems that require you to provide an algorithm, you must give the following:

1. a precise description of the algorithm in English and, if helpful, pseudocode,
2. a proof of correctness,
3. an analysis of running time and space.

You may use algorithms from class as subroutines. You may also use any facts that we proved in class.

You should be as clear and concise as possible in your write-up of solutions.

A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error and because it is easier to read and understand. Points might be subtracted for illegible handwriting and for solutions that are too long. Incorrect solutions will get from 0 to 30% of the grade, depending on how far they are from a working solution. Correct solutions with possibly minor flaws will get 70 to 100%, depending on the flaws and clarity of the write up.

1. **Stable Matching** (2-page limit, 15 points)

- (a) (**Unique Matching**) Give an algorithm that takes an instance of the stable matching problem as input and decides if there is *exactly one* stable matching for this instance. That is, the algorithm takes as input the preference lists for two groups of the same size, n (say, hospitals and residents), and outputs either “unique”, or “more than one”. Your algorithm should run in polynomial time (for example, $O(n^2)$ or $O(n^{4.2})$; it shouldn't require exponential time).

Notice that proving your algorithm correct requires proving three statements:

- i. The algorithm always terminates;
- ii. If there is a unique stable matching, then the algorithm will output “Unique”;
- iii. If the algorithm outputs “Unique”, then there really is only one stable matching.
(This is the most challenging statement of the three)

Hint: It is helpful to read the section on “Extensions” at the end of KT Section 1.1; specifically, read and understand Claims 1.7 and 1.8.

- (b) (**Lower bounds**) Prove that it is possible for the Gale-Shapley algorithm to perform $\Omega(n^2)$ offers before termination. [*Hint:* One way to do this is to describe, for each n , both a suitable input and a sequence of $\Omega(n^2)$ valid offers.]

2. **Asymptotics Review** (1-page limit, 10 points)

Rank the following functions by their asymptotic growth rate (big-O relationships).

List functions on separate lines, where higher lines correspond to asymptotically larger functions (so $f(n)$ goes above $g(n)$ if $g(n) = O(f(n))$ and $f(n) \neq O(g(n))$). If two functions are asymptotically equivalent (that is, $f(n) = O(g(n))$ and $g(n) = O(f(n))$), then they should go on the same line.

n^{100}	$n^3 - 5n^2 - 10n + 12$	n^2	\sqrt{n}	2^n
n^3	$\log_2 n$	$\log_3 n$	$\sum_{i=1}^n i$	2.1^n
$10 + \frac{1}{n}$	$n \log_2 n$	$n \cdot 2^n$	$\binom{n}{2} - n + 2$	$\log_2(n!)$

Hint: You may want to review Big-Oh rules and logarithm properties.