MATH 416, PROBLEM SET 1

Comments about homework.

- Solutions to homework should be written clearly, with justification, in complete sentences. Your solution should resemble something you'd write to teach another student in the class how to solve the problem.
- You are encouraged to work with other 416 students on the homework, but solutions must be written independently. Include a list of your collaborators at the top of your homework.
- You should submit your homework on Gradescope, indicating to Gradescope where the various pieces of your solutions are. The easiest (and recommended) way to do this is to start a new page for each problem.
- Attempting and struggling with problems is **critical** to learning mathematics. Do not search for published solutions to problems. I don't have to tell you that doing so constitutes academic dishonesty; it's also a terrible way to get better at math.

  If you get stuck, ask someone else for a hint. Better yet, go for a walk.

Problem 1. Arrange the following list of functions in ascending order of growth rate. That is, if function $g(n)$ immediately follows function $f(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$. Give a brief explanation for each pair of consecutive functions.

$$g_1(n) = 2^{\sqrt{\log n}}$$
$$g_2(n) = 2^n$$
$$g_3(n) = n(\log n)^3$$
$$g_4(n) = n^{4/3}$$
$$g_5(n) = n^{\log n}$$
$$g_6(n) = 2^{2^n}$$
$$g_7(n) = 2^{n^2}$$

Problem 2. Assume that $f(n)$ and $g(n)$ are functions such that $f(n)$ is $O(g(n))$. For each of the following statements, decide whether it is true or false and provide either a proof or counterexample:

(a) $\log_2 f(n)$ is $O(\log_2 g(n))$.
(b) $2^{f(n)}$ is $O(2^{g(n)})$.
(c) $f(n)^2$ is $O(g(n)^2)$.

Problem 3. Prove that $o(f(n)) \cap \omega(f(n))$ is the empty set. (See WS3 for the definition of little o and little ω.)

Problem 4. Suppose that H and S together with preference arrays are given. For $h \in H$ and $s, t \in S$, define $\max_h(s, t)$ to be the higher of s and t in h's preference list. Show that if M and P are two stable matchings for this instance of the stable-matching problem, then

$$M \vee P = \{(h, \max_h(s, t)) | (h, s) \in M, (h, t) \in P\}$$

is also a stable matching. (In particular, you must show that it is a matching.)

Problem 5 (Examples of stable matchings).

(a) Give preference lists (for a fixed small n, say $2 \leq n \leq 10$) and a stable matching in which no hospital or student is matched with their first choice.

(b) Now for any $n \geq 2$, give preference lists and a stable matching in which every student is matched with their first choice but every hospital is matched with its last choice.

(c) Show that for every $k \geq 1$ there are preference lists for $n = 2k$ hospitals and $2k$ students that admit at least 2^k stable matchings. (So the number of stable matchings can grow exponentially in the number of parties.)

(*Hint:* First solve the problem for $k = 1$. Then for the general case try to paste together many 'independent' copies of your solution for $k = 1$.)

Problem 6.

(a) Prove that in a stable matching produced by the GS Algorithm, at most one hospital is paired with its last choice. (Contrast with Problem 5b.)

(b) For every n, give an example of preference lists such that in the stable matching produced by the GS Algorithm, there is exactly one hospital matched with its last choice and each of the other $n-1$ hospitals is matched with its second-to-last choice.

(*Hint:* One approach is to make one student the last choice of every hospital.)

(c) Prove that, for every $n \geq 1$, there are preference lists for $|H| = |S| = n$ for which the `while` loop in the GS algorithm must iterate $n^2 - n + 1$ times, so that the running time of the algorithm is $\Theta(n^2)$.

(*Hint:* This is not unrelated to the previous part.)

Problem 7. Recall that the Gale–Shapley Algorithm produces the unique hospital-optimal, student-pessimal stable matching. By falsifying their preference list, can a student end up (after running the algorithm) with a better position than they would with their true preference list? Give a proof or an example.

Problem 8. Suppose that we are given preference lists for $|H| = |S| = n$. Prove that there is no perfect matching, stable or otherwise (!), in which *every* hospital is matched with a student it strictly prefers to the student matched with it by the Gale–Shapley Algorithm.

(*Hint:* Consider the final step of the algorithm before it stops.)

Problem 9. A past MATH 416 student proposed the following algorithm for the stable-matching problem.

> Start with an arbitrary perfect matching M of hospitals and students. (For instance, $\{(h_1, s_1), (h_2, s_2), \ldots, (h_n, s_n)\}$.)
>
> Search for pairs of pairs $(h, s), (h', s') \in M$ for which (h, s') is a blocking pair; if you find one, then rematch these four parties: i.e., replace the pairs $(h, s), (h', s')$ with $(h, s'), (h', s)$.
>
> Repeat until there are no more blocking pairs.

This algorithm will certainly succeed if it halts. Prove that, unfortunately, it need not halt.

(*Hint:* You should be able to produce an example with $|H| = |S| = 3$.)