EE360C: Algorithms
University of Texas at Austin — Homework #2
Dr. David Soloveichik — Due: September 20, 2018 (*in-class quiz*)

Homework #2

You do not need to turn in these problems. The goal is to be ready for the in-class quiz that will cover the same or similar problems, and to prepare you for the exams.

Problem 1: Truthfulness in Stable Matching

For this problem, we will explore the issue of truthfulness in the Stable Matching Problem and specifically in the Gale-Shapley algorithm. The basic question is: can a man or a woman end up better off by lying about his or her preferences? More concretely we suppose each participant has a true preference order. Now consider a woman w. Suppose w prefers man m to m_0, but both m and m_0 are low on her list of preferences. Can it be the case that by switching the order of m and m_0 on her list of preferences (i.e., by falsely claiming that she prefers m_0 to m) and running the algorithm with this false preference list, w will end up with a man m_{true} that she truly prefers to both m and m_0? (We can ask the same question for men but will focus on the case of women for the purposes of this question.) Resolve this question by doing one of the following two things:

Give a proof that, for any set of preference lists, switching the order of a pair on the list cannot improve a woman's partner in the Gale-Shapley algorithm; or

Give an example of a set of preference lists for which there is a switch that would improve the partner of a woman who switched preferences.

Problem 2: Understanding Stable Matching Optimality

We say a man m is a *valid partner* for woman w if there exists some stable matching in which they are matched. A *man-optimal* stable matching is the one in which every man receives his best valid partner. A *woman-pessimal* stable matching is the one in which every woman receives her worst valid partner.

Recall from class that the Gale-Shapley algorithm returns the man-optimal stable matching. Prove that this stable matching is also woman-pessimal. (You may use the man-optimality of Gale-Shapley without proof.)

Problem 3: Stable Marriage Runtime

Prove that the number of proposals made in the course of the Gale-Shapley algorithm is $\Omega(n^2)$ in the worst case.

Note that this proves that the running time of Gale-Shapley is $\Omega(n^2)$. We already saw in class that it is $O(n^2)$; thus, Gale-Shapley running time is $\Theta(n^2)$.

Problem 4: Properties of Big-Oh

(a) Prove that for any constant $d > 0$, $f(n) = O(d \cdot g(n))$ iff $f(n) = O(g(n))$. (Note that this justifies dropping multiplicative constants in O notation. We won't say something like: the algorithm runs in $O(2n^2)$ time.)

(b) Prove that if $f(n) = O(h_1(n))$ and $g(n) = O(h_2(n))$ then $f(n)+g(n) = O(\max\{h_1(n), h_2(n)\})$. (Note that this property is used all the time in algorithms (usually implicitly!): If your algorithm first does something that's $O(n)$ and then something that's $O(n^2)$, we claim that the overall algorithm is $O(n^2)$.)

Problem 5: Comparing Efficiency

For a given problem, suppose you have two algorithms: A_1 and A_2 with worst-case time complexity of $T_1(n)$ and $T_2(n)$, respectively. For each part, answer the following four questions:

- Is $T_1(n) = O(T_2(n))$?
- Is $T_1(n) = \Omega(T_2(n))$?
- Is $T_1(n) = \Theta(T_2(n))$?
- If your goal is to pick the fastest algorithm for large n, would you pick A_1 or A_2?

(a) $T_1(n) = 5\sqrt{n} + \log_2 n + 3$ and $T_2(n) = 100\log_2 n + 25$

(b) $T_1(n) = 5\sqrt{n} + \ln n$ and $T_2(n) = \sqrt{n} \cdot \log_2 n$

(c) $T_1(n) = 5\log_{10} n - 20$ and $T_2(n) = \frac{1}{2}\log_2 n$

(d) $T_1(n) = 10n^{2/4}$ and $T_2(n) = \sqrt{n} \cdot \log_2 n$

(e) $T_1(n) = 10^{(2\log_{10} n)} - 2n$ and $T_2(n) = 5n^3 + 10n - 3$

Problem 6: Asymptotics of Logs of Factorials

Prove that $\log(n!) = \Theta(n \log n)$. (This appears in a number of contexts, including for proving lower bounds on sorting algorithms.)