CS41 Homework 2

This homework is due at 11:59PM on Sunday, September 22. Write your solution using LaTeX. Submit this homework in a file named `hw2.tex` using **github**. This is an individual homework. It's ok to discuss approaches at a high level. In fact, I encourage you to discuss general strategies. However, you should not reveal specific details of a solution, nor should you show your written solution to anyone else. The only exception to this rule is work you've done with a lab partner/group *while in lab*. In this case, note (in your post-homework survey) who you've worked with and what parts were solved during lab.

The main **learning goals** of this homework are to work with stable matching and the Gale-Shapley algorithm, to get comfortable analyzing it and applying algorithmic thinking, and to practice writing thorough arguments.

1. **Hipster Coffee Tours.** A group of n Portland hipsters $H = \{h_1, \ldots, h_n\}$ are touring a set of local coffee shops $C = \{c_1, \ldots, c_n\}$ over the course of $m \geq n$ days. Each hipster h_j has an itinerary where he/she decides to visit one coffee shop per day (or maybe take a day off if $m > n$). However, hipsters are fiercely independent and prefer not to share coffee shops with other hipsters. Furthermore, each hipster is looking for a favorite coffee shop to call his or her own. Each hipster h would like to choose a particular day d_h and stay at his/her current coffee shop c_h for the remaining $m - d_h$ days of the tour. Of course, this means that no other hipsters can visit c_h after day d_h, since hipsters don't like sharing coffee shops.

   Show that no matter what the hipsters' itineraries are, it is possible to assign each hipster h a unique coffee shop c_h, such that when h arrives at c_h according to the itinerary for h, all other hipsters h' have either stopped touring coffee shops themselves, or h' will not visit c_h after h arrives at c_h. Describe an algorithm to find this *matching*.

   Hint: The input is somewhat like the input to stable matching, but at least one piece is missing. Find a clever way to construct the missing piece(s), run stable matching, and show that the final result solves the hipster problem.

   It may be necessary to break ties; i.e., two hipsters might choose to visit the same coffee shop on the same day. You may assume that the tie can be broken by having hipsters arrive at different times of the day such that if h and h' both want to visit c on the same day, that there is some timestamp on their visits such that it is easy to determine who arrived at c first. Thus, for any given day, at any given coffee shop, there is a well-defined ordering to the planned arrival time of the hipsters.

2. **Asymptotic Analysis.** Let $f(n) = n^2 \log(n) + 17n - 4$ and $g(n) = \frac{n^{2.5}}{2} - n$. Prove that $f(n) = O(g(n))$. You may use techniques and facts from class and the textbook. Your proof should be complete and formal.

3. **Asymptotic analysis.** Arrange the following functions in ascending order of growth rate. That is, if g follows f in your list, then it should be the case that $f = O(g)$.

   - $f_1(n) = \frac{\sqrt{n}}{6}$
   - $f_2(n) = 12n \log(n)$

- $f_3(n) = 5\log(n)^4$
- $f_4(n) = \pi \cdot 2^n$
- $f_5(n) = 7n^3$
- $f_6(n) = 16n^2 + 22n$

No proofs are necessary.

4. **(extra credit problem) Homework Partner Matching.** In class, we discussed a version of the stable matching problem where we want to match n doctors to n hospitals. In this problem, we discuss the homogeneous version. The input is a set of students $A = \{s_1, \ldots, s_{2n}\}$ of size $2n$. Each student ranks the others in order of preference. A homework partner assignment of students into partners $M = \{(i, j)\}$ is a matching; it is unstable if there exists $(i, j), (i', j') \in M$ such that i prefers j' to j and that j' prefers i to i'. It is stable if it is a perfect matching and there are no instabilities.

   (a) Does a stable homework partner assignment always exist? Prove that such an assigment must always exist, or give an example where no stable assignment occurs. (Remember, you must have $2n$ students.)

   (b) Design and analyze an efficient algorithm that either returns a stable matching for homework partners or outputs that no such matching exists.