YOUR NAME HERE

Homework 1

This homework is due by 4pm, September 15, 2015. Your homework must be turned in as a PDF file and uploaded to the class blackboard page by 4pm.

1. (2 points) Let A, B be two magicians and 1, 2 two wands. Come up with preference lists such that both possible pairings are stable.

2. (5 points) Assume that among n magicians and n wands, there is a magician and wand that mutually rank each other first. Prove that the Gale Shapely algorithm will always match them.

3. (5 points + 5 points for 541 problem) In this problem, we will see the effect of changes in preferences in the outcome of the Gale-Shapley algorithm (for this problem you can assume the version of the Gale-Shapley algorithm that we did in class where the magicians do all the requesting).

   Given an instance of the stable marriage problem (i.e. set of magicians M and the set of wands W along with their preference lists: \( L_m \) and \( L_w \) for every \( m \in M \) and \( w \in W \) respectively), call a wand \( w \in W \) potentially evil if the following property holds:

   There exists a new preference list \( L'_w \) for \( w \) such that if \( w \) changes their preference from \( L_w \) to \( L' \) then the Gale Shapley algorithm matches all magicians to a different wand.

   Prove there is an instance of the stable matching problem with 4 magicians and 4 wands so that there is a wand who is potentially evil.

   **541 problem, bonus for 441 students:** Generalize your previous answer to show that for any number \( n \) that is 4 or bigger there is an instance of the stable matching problem with \( n \) magicians and \( n \) wands where there is a wand that is potentially evil.

4. (9 points) Greedy Fresh Baker (GFB) has the best oven in all the land. Each day the baker gets orders for \( n \) cakes. For each cake the baker knows exactly how long it will take to bake the cake — for cake \( i \) it will take \( t_i \) time in the oven, only one cake can be in the oven at a time, and the oven is always being used. When the last cake is finished, the baker instantly delivers all the cakes to the clients.

   This problem asks you to help the baker design an \( O(n \log n) \) time algorithm that finds the order to bake the cakes which provably minimizes the total staleness of all the cakes. The staleness is defined to be the time that a cake is waiting, after it is cooked, before it is delivered. Here is a simple example. Suppose \( n = 3 \), and the cooking times of those three cakes are: \( t_1 = 5, t_2 = 10, t_3 = 4 \).
Now consider the schedule: 1, 2, 3.
In this case, cake 1 is baked first for 5 minutes, then cake 2 is in the oven for 10 minutes, then cake 3 is in the oven for 4 minutes. The total total staleness is then:

\[(10 + 4), \text{ [the time that cake 1 waited, after it was finished, before it was delivered]}\]
\[+ 4 \text{ (the time that cake 2 waited, after it was finished, before it was delivered)},\]
\[+ 0 \text{ (because the third cake had no wait time before it was delivered)}\]
\[... \text{for a total staleness of 18}.\]

This schedule, however, is not optimal. Derive and explain a correct algorithm, prove that it is correct, and give its running time.

Correct algorithm description: 2 points
Runtime analysis: 2 points
Proof of correctness: 5 points