Introduction to Algorithms	Problem Set 1
CS 4820 Spring 2016	Lillyan Pan

1 Problem 2

1.1 Algorithm

1. We use a set S to hold matched pairs of hospitals and residents, let R be the set of applicants and H be the set of hospitals
2. Initially all $h \in H$ and $r \in R$ are unmatched
3. While there is a free hospital h that hasn't made offers to all residents on its preference list
   - Choose such a hospital h
   - Let r be the highest-ranked resident in hs preference list that h has not offered to
   - If r is free then
     - (r, h) become matched
   - Else r is currently matched to h'
     - If r prefers h' to h then h remains free
     - Else r prefers h to h'
       - (r, h) become matched
       - Hospital h' becomes free
4. Return set S of matched pairs

1.2 Runtime

Assuming the the generation and ordering of preference lists are already given, the runtime takes $O(mn)$. Each iteration of the while loop involves one offer. Note, because hospitals move monotonically down their preference lists, no hospital will offer to the same student twice. Note, $k \leq m$. A hospital will make n offers at most. Thus, the number of iterations of the while loop is at most mn before the algorithm halts. Each iteration contains operations that run in $O(1)$ time. Hence, total running time is $O(mn)$.

1.3 Correctness

We will show correctness using a proof by contradiction. Let S denote the match constructed by our algorithm from the set of applicants R and the set of hospitals H, and let $S \subseteq R \times H$, in which $r_i \in R$ for $i \in [1, n]$ and $h_j \in H$ for $j \in [1, m]$ denote its elements. Let us call a matching stable if it does not exhibit the following types of instability

1. There are two pairs (r, h) and (r', h') in S with the property that h prefers r' to r, and r' prefers h to h'.
2. There is a resident r, and a free hospital slot h on r's preference list that is unmatched.
3. There is a pair (r, h) in S, and a free hospital slot h' on r's preference list, so that h' is not part of any pair in the matching, and r prefers h' to h.

4. There is a pair (r, h) in S, and a student r' that ranks h, so that r' is not part of any pair in the matching, and h prefers r' to r.

Lemma 1. *Given any set of prospective residents and hospitals and any preference lists, there exists a stable matching.*

Proof of Lemma 1. To prove the algorithm will always produce a stable matching, we will prove by contradiction that the four types of instability are unattainable.

First, suppose there is an instability of type (1). It follows that h must have offered to r'; so r' rejected h and thus the student prefers her final match h' to h. This contradicts our original assumption, hence instability (1) is not possible.

Next, suppose there is an instability of type (2). For h to be unmatched, the position must have been offered to every applicant that listed the particular residency option; in particular, h must have offered to r which means r would no longer be free. This contradicts our original assumption, hence instability (2) is not possible.

Next, suppose there is an instability of type (3). Then h' must have offered to r and been rejected; it follows that r prefers her final match h to h'. This contradicts our original assumption, hence instability (3) is not possible.

Finally, suppose there is an instability of type (4). Then no hospital offered to r' at all; in particular, h never offered to r', and so h must prefer r to r'. This contradicts our original assumption, hence instability (4) is not possible.