

CS 4820 HW #1 PROBLEM #2

(a) An algorithm to solve this problem that runs in $O(nm)$ is a variation of the Gale-Shapley algorithm as follows:

- (1) Pick an unmatched resident r_1 , and have them apply to the highest hospital on their ranking that they haven't applied to yet.
- (2) If the hospital does not have any residents matched to it, match it with r_1 .
- (3) If the hospital is already matched to another resident r_2 :
 - (a) If the hospital prefers r_1 to r_2 , unmatched it with r_2 and match it with r_1 .
 - (b) If the hospital prefers r_2 to r_1 , do not change the matching and cross the hospital off r_1 's list.
- (4) Repeat steps 1-3 until either all residents or all hospitals are matched.

Lemma This algorithm runs in $O(nm)$ time.

Proof. There are n applicants, and each applies to at most m hospitals. Since each application can be done in constant time, the algorithm runs in at most $O(nm)$ time. \square

Lemma This algorithm produces a weakly stable matching.

Proof. The first property is trivial: each applicant only applies to hospitals on their preference list, and a matching can only come as a result of an application, so all residents are either unmatched or matched with a hospital on their list.

The second property is harder. Pick any unmatched pair (r, h) . Either r applied to h , or they did not.

Suppose r did apply to h . Since they are unmatched, h must have either rejected r from the start, or accepted r and then proceeded to reject them in favour of another applicant. In the former case, h must have been matched with a better applicant r' by the time r applied there; in the second case, h must have received an application from a better applicant r' **after** r had applied there. Since hospitals are only unmatched with applicants if they find a better applicant, the resident that h was eventually matched to must have been better than or equal to r' on their preference list; in any case, h must prefer this resident to r . This means that the first part of the second property holds.

Now suppose that r did **not** apply to h . This could be due to one of two things. One case is that h was not on r 's preference list; in that case, the second property need not apply. The other case is that h **was** on r 's preference list, but r never applied to it. Since residents keep applying to hospitals in the order of their preference, at some point r must have matched with a better hospital than h and stayed matched. This means that the second part of the second property holds.

Both the first and second properties must hold, and therefore the matching produced by this algorithm is weakly stable. \square

(b) The simplest example is when $k = 1$. Suppose every resident prefers some hospital h_1 , and only applies there. This hospital will take its top choice, after which the algorithm will terminate, leaving the other students without a match. To improve their standing, another student could apply to their second choice, h_2 , in which case they will be matched with the latter. Assuming that any match is better than no match, this satisfies the problem criteria.