I collaborated with: Nevin Bernet

Problem
The National Resident Matching Program (NRMP) matches medical students with residency programs. Here is what the NRMP says on their website:

> The NRMP conducts its Matches using a mathematical algorithm that pairs the rank ordered preferences of applicants and program directors to produce a best fit for filling available training positions. Research on the NRMP algorithm was a basis for Dr, Alvin Roths receipt of the 2012 Nobel Prize in Economics.

In this matching problem, there are n students and m hospitals. Each hospital h_i has p_i available positions. Each student ranks the m hospitals and each hospital ranks the n students. Since there are more students than total positions available, we assume that $n > \sum_{i=1}^{m} p_i$. Thus, some students are never matched. As a result, we need a slightly expanded version of stability. As before, the matching is unstable if

- s is assigned to h and s' is is assigned to h' but s prefers h' to h and h' prefers s to s'.

But it is also unstable if

- s is assigned to h and s' is not assigned to a hospital but h prefers s' to s.

Give an algorithm to find a stable matching of students to hospitals where every hospital position is filled with a student. Show that your algorithm is correct and that it runs in time polynomial in n and m. Your algorithm description and analysis should be clear and concise.
Solution

Algorithm 1 Stable matching

Require: set of lists of every student's preference and another set of lists of every hospital's preference.

```
initialize each student and hospital to be free
while some student is free and hasn't "proposed" to every hospital do
    s ← such student
    h_i ← 1st hospital on s's list to whom s has not yet proposed
    if h_i still has available positions then
        match s to h_i
    else if h_i prefers s to another student already matched in the hospital then
        swap least preferable student with s
    else
        h_i rejects s
    end if
end while
```

Space: n lists of length n, m lists of length m, m heaps of size p_i: $O(n^2 + m^2 + mp_i)$
Time: The selection of students, a hospital's ranking of a given student and vice versa, are almost identical to the P-R algorithm which we discussed were all constant time operations. Heaps allow us to check the least preferred student currently in a hospital in constant type. Swapping students takes $O(\log(n))$ time as the new student added needs to percolate down to their spot in the heap based on the hospital's preference.
Overall Running Time:

- Preprocessing: $O(n^2)$ for lists + $O(n)$ for queue of students.
- Loop: $O(n^2) * O(1) * O(\log(n))$

Suppose, for the sake of contradiction, that the algorithm proposed above does not produce a stable matching. Then, the final matching has an instability $\{s, h\}$. That is, s is matched to h' but s prefers h over h'. Also, h is matched to s' but h prefers s over s'. According to the algorithm, s proposed to h before proposing to h'. Therefore, h either removed s from its heap, or s did not rank high enough to get in the heap. However, hospitals always choose the most preferred candidate, so h prefers s' over s. This contradicts the instability defined above, so our algorithm produces a stable matching.