Written Assignment #2

Tanner J. Evans

CS 361 – Algorithms and Data Structures

September 8, 2020

1.4: Gale and Shapley published their paper on the Stable Matching Problem in 1962; but a version of their algorithm had already been in use for ten years by the National Resident Matching Program, for the problem of assigning medical residents to hospitals.

Basically, the situation was the following. There were *m* hospitals, each with a certain number of available positions for hiring residents. There were *n* medical students graduating in a given year, each interested in joining one of the hospitals. Each hospital had a ranking of the students in order of preference, and each student had a ranking of the hospitals in order of preference. We will assume that there were more students graduating than there were slots available in the m hospitals.

The interest, naturally, was in finding a way of assigning each student to at most one hospital, in such a way that all available positions in all hospitals were filled. (Since we are assuming a surplus of students, there would be some students who do not get assigned to any hospital.)

We say that an assignment of students to hospitals is stable if neither of the following situations arises:

- First type of instability: There are students *s* and *s'*, and a hospital *h*, so that
  - *s* is assigned to *h*, and
  - *s'* is assigned to no hospital, and
  - *h* prefers *s'* to *s*.
- Second type of instability: There are students s and s, and hospitals *h* and *h'*, so that
  - *s* is assigned to *h*, and
  - *s'* is assigned to *h'*, and
  - *h* prefers *s'* to *s*, and
  - *s'* prefers *h* to *h'*.

So we basically have the Stable Matching Problem, except that (i) hospitals generally want more than one resident, and (ii) there is a surplus of medical students.

Show that there is always a stable assignment of students to hospitals, and give an algorithm to find one.

```
h = a hospital, H = the set of hospitals, s = a student, and S = the set of students
      Initially all h ∊ H have no students assigned, all s ∊ S are unassigned, and
the number of s exceeds the number of assignments for all h ∊ H.
While there is a hospital h who has an assignment unfilled.
      Select such a hospital h from among those with the fewest assignments filled
      Let s be the highest-ranked student in h’s preference list whom h has not yet
            offered an assignment
      h offers s an assignment
      If s is free, then
            s is assigned to one of h’s assignment slots
      Else s is already assigned to h’
            If s prefers h’ to h then
                  s remains assigned to h’
            Else s prefers h to h’ and
                  s is assigned to h and
                  h’ frees an assignment slot.
            Endif
      Endif
Endwhile
Return the set H of hospitals with filled assignment slots.
```

Given the two definitions of instability we are given, we must show that neither can arise given the application of the Gale-Shapley Algorithm.

The first instability is where s is assigned to h, s' is unassigned, and h prefers s' to s.

Let us assume that this has occurred. But h offers assignment by order of their preference list. They must therefore have already offered s' an assignment, and therefore s' can not be unassigned. Therefore we can not experience this instability.

The second instability is where s is assigned to h, s' is assigned to h', h prefers s' to s, and s prefers h to h'.

Let us assume that this has occurred. Since h ranks s' higher than s, h must have offered an assignment to s' first. The only way for s' to change assignments once assigned is by being offered an assignment at a hospital they prefer. The only way this instability could then arise is if s' changed from a more preferable hospital to a less preferable one, which is not possible given a Gale-Shapley Algorithm. The same is true of s. Therefore we can not experience this instability.

Since we have shown that the two forms of instability can not occur given the application of a Gale-Shapley Algorithm, the Algorithm must then always provide a stable set of assignments.