I have read and agree to the collaboration policy. Lynne Diep.

Lynne Diep
Homework Heavy Grading
Collaborators: None

CMPS 102: HW 1-1
Due: April 21, 2017

Problem 1:
Show that there is always a stable assignment of students to Google teams, and give an algorithm to find one.

- The first type of instability that can occur is that there is a team t, and there are students s and s', so that
  - s is matched with t, and
  - s' is assigned to no team, and
  - t favors s' over s.

- The second type of instability that can occur is that there are teams t and t0 and students s and s0 so that
  - s is matched with t, and
  - s' is matched with t', and
  - t favors s' over s, and
  - s' favors t over t'.

So we basically have the Stable Matching Problem, except that, one, teams generally want more than one intern, and, two, there is a surplus of students who want internships at Google.

Algorithm:
At any time, a team either has available or unavailable positions. A student is either committed to a particular team or free.

Let there be t teams and s students, and are initially free.

The algorithm shows stable assignment of students to teams –

```
while some team t has available positions {
        t offers a position to student s on its preference list
        if s is free then
                s accepts the offer
        else s already accepted to team t'
        if s prefers t' to t {
                s remains committed to t'
        }
        else s remains committed to t {
                the number of positions at t’ increase by 1
                the number of positions at t decrease by 1
        }
}
return matches between students and teams, and free students who did not receive an internship
```

Information about algorithm –

a. student “s” is matched to team where they get their first offer, s will only change matches if the team is higher on their preference list.
b. The team offers get worse since they give to the students at the top of their preference list first, and leave the “worser” teams to those at the bottom on their list.
c. Algorithm terminates after (total # of teams) * (total # of students) iterations

Running Time - $O(n^2)$ where n is the number of teams and students

How the algorithm is stable –
First instability: s is assigned to t, s' is assigned to no team, and t prefers s' to s:
t makes offers in order of preference, and s takes the offer if they are free. s’ is assigned to no team. If t prefers s’ to s, then t would have made an offer to s’ first. So this is a contradiction to the assumption, thus the algorithm is stable.

Second instability: s is matched with t, and s’ is matched with t’, and t favors s’ over s, and s’ favors t over t’:
t made offers until all positions were filled. If t preferred s’ to s, then t would offer the position to s’ first, if not, then t would have preferred s to s’; this contradicts the assumption. Thus, the algorithm is stable.