Adit Jain
aj27923
EE360C
Programming Assignment #1
Dr. David Soloveichik

Report

Part (a) : Pseudocode

Initially all students are free and haven't exhausted all their proposals. Place them into a queue. Also keep track of what internship a student has to propose to next, given he has proposed to all earlier internships. I did this by keeping an ArrayList where index values represent students and the values at those indices represent the index at which that particular student has to propose to next.

```
1: while (size of queue != 0)
2:      remove student s from head of queue
3:      let I be the internship from s' preference list that he must propose to next
4:      if (I has open slots)
5:           match s with I
6:      else
7:           if (I prefers s over its lowest preferred current match, say x)
8:                match s with I
9:                add x to the queue since he/she is free
10:          else
11:               add s back onto the queue
12:
13: return the set of students and each of their internship assignments
```

Part (b) : Runtime Complexity

The Big O of this algorithm is mn^2, where m is the number of internships and n is the number of students. There are certain difficulties involved with trying to implement an O(mn) solution algorithm for several reasons. For one, keeping a running track of every internship's current lowest preferred match involves looping through up to n students in an internship's list of current filled slots to find a least preferred current match. This is an O(n) runtime for this alone. On the other condition inside the loop, finding the size of the array list holding an internship's current student matches is an O(n) algorithm. Both of these algorithms are taking place inside the outer loop, which has a runtime of O(mn). Therefore the overall runtime is $O(mn^2)$ since O(mn) is multiplied by either one of the two conditionals that take place inside the outer loop, which both have a runtime of O(n) independently.