Lab 5

Steve Boucher

December 15, 2022

Bags and Graphs

1 Introduction

The goal of the final was to come up with our own solution to the hospital and residents situation in which an algorithm is designed to pair each resident to a hospital with X spaces. Each pair must be in a stable relationship, meaning that no hospital and no resident would prefer to make a pairing over their current. Two versions of this issue were required for this lab, One where the Hospitals rank the residents and the second scenario they do not.

2 File Reading

I employed a similar strategy to read the data on a text file as I did in lab 5. This was done by abusing the function.

```
String.charAt(int);
```

to where I could check the first line of a command for a specific letter and assume what the first letter means to the rest of the command. If it was an r, then I was adding a resident and if it was an H, I was adding a hospital. I ended up running through the text file a total of two times for each part. The first run-through was to set up residents and hospitals, and the second pass-through was to fill in the preferences and hospital capacity.

```
25        while (sc1.hasNextLine()) {
26            textCommand = sc1.nextLine();
27            int charCounter = 1;
28            int id = 0;
29            String stringCounter = " ";
30            if (textCommand.charAt(0) == 'r') {
31
32                while (textCommand.charAt(charCounter) != ':') {
31                    stringCounter = stringCounter
                    + textCommand.charAt(charCounter);
32                    charCounter++;
33                }
34
35                id = Integer.parseInt(stringCounter.trim());
36                // itest(id);
37                rList.insert(id);
38            }
{
```

In the code above. I have displayed a snippet of code used for adding a resident to the list. I find the first character is an r or resident, then find the first digit and tell it that that will be its id.

3 Part 1

The solution I used to the problem was based on the pseudo-code given in the slide deck.

```
332             while (rList.getRes(resItteration).match[0] == -1) {
333                 while (hList.checkSpace(rList.getRes(resItteration).prefs.get(prefCounter)) == 0)
334                     prefCounter++;
335                 }
336                 if (hList.checkSpace(rList.getRes(resItteration).prefs.get(prefCounter)) >= 0
                         && rList.getRes(resItteration).match[0] == -1) {
338                     rList.getRes(resItteration).match[0] =           rList.getRes(resItteration).p
339                     hList.match(rList.getRes(resItteration).prefs
                         .get(prefCounter), resItteration);
340                 }
{
```

4 Conclusion

I'm very proud of the work I did on this final project. I would have liked to figure out a more concrete/reliable solution but was very pleased with how far I was able to get on my own. The problem was definitely challenging, but I don't think it has been as bad as some of our labs. Thank you so much professor for an amazing semester