Full Name:	Pu Wentao
Email:	pu.w@husky.neu.edu
Test Name:	**INFO 6205 Spring 2019 Section_3 FINAL**
Taken On:	25 Apr 2019 13:37:12 EDT
Time Taken:	30 min 37 sec/ 60 min
NUID:	001407516
Invited by:	Robin
Invited on:	25 Apr 2019 13:31:45 EDT
Tags Score:	

100%

50/50

scored in **INFO 6205 Spring 2019 Section_3 FINAL** in 30 min 37 sec on 25 Apr 2019 13:37:12 EDT

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Gift giving > **Coding**	30 min 18 sec	50/ 50	✓

QUESTION 1

Correct Answer

Score 50

Gift giving > Coding

QUESTION DESCRIPTION

You are a team leader and want to give your team members gifts as rewards for their hard work. You are careful to ensure that no team member gets more than one gift. Each member *m* has a greed factor G_m, which is the minimum price of a gift that the team member will be happy with; and each gift *x* has a price P_x. If P_x >= Gm, the member *m* will be happy. Your goal is to write a method to maximize the number of your happy team members and return that number.

Note: you may assume the greed factor and price are always positive Integers. You cannot assign a gift to more than one member and no member may have more than one gift.
Hint: think greedily, how to maximize the value of each gift and who should be the most appropriate person to get the gift.

input format:
N (the number of team members) G_0 G_1 ... G_N
M (the number of gifts) P_0 P_1 ... P_M

Example 1 input:
3 1 2 3

3 1 2 3
3 1 1 1
expected result: 1
explanation: at most one team member will be happy.

Example 2 input:
3 1 2 3
3 3 2 1
expected result: 3
explanation: at most 3 team members will be happy.

CANDIDATE ANSWER

Language used: **Java 8**

```
class Result {

/*
     * Complete the 'assignGift' function below.
     *
     * The function is expected to return an int.
     * The function accepts the following parameters:
     *  1. List<Integer> greedFactors (greed factors of each member)
     *  2. List<Integer> prices (the prices of the gifts you have)
     * @return max number of happy members
     */

    public static List<Integer> sortList(List<Integer> list){
       for(int i =0;i<list.size();i++){
           for(int j =i+1;j<list.size();j++){
               if(list.get(i)< list.get(j)){
                   Integer tmp = list.get(i);
                   list.set(i,list.get(j));
                   list.set(j,tmp);
               }
           }
       }
       return list;
    }
    public static int assignGift(List<Integer> greedFactors, List<Integer>
prices) {
       greedFactors= sortList(greedFactors);
       prices= sortList(prices);

        int count = 0;
        //int num = Math.min(greedFactors.size(),prices.size());
        int num = greedFactors.size();
        int index =0;
        //boolean[] flag = new boolean[greedFactors.size()];
        for(int i =0;i<num;i++){
            if(index>=prices.size())
                break;
            if(greedFactors.get(i)<=prices.get(index)){
                index++;
                count++;
            }
        }
        return count;
    }

}
```

```
47
48
49
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
eg1	Easy	Sample case	Success	8	0.302 sec	30.7 MB
eg2	Easy	Sample case	Success	8	0.298 sec	30.9 MB
different size	Medium	Sample case	Success	5	0.299 sec	30.6 MB
size4	Easy	Sample case	Success	8	0.3 sec	31.1 MB
no match	Easy	Sample case	Success	8	0.3 sec	30.7 MB
one match	Easy	Sample case	Success	8	0.3 sec	31 MB
different size 2	Medium	Sample case	Success	5	0.297 sec	30.3 MB

No Comments