work allocate

Problem	Submissions	Leaderboard	Discussions

Given an array of jobs where every job has a deadline and associated profit if the job is finished before the deadline. It is also given that every job takes a single unit of time, so the minimum possible deadline for any job is 1. Maximize the total profit if only one job can be scheduled at a time.

Input Format

a 4 20

b 1 10

c 1 40

d 1 30

Output Format

c a

Sample Input 0

```
a 2 100
b 1 19
c 2 27
d 1 25
e 3 15
```

Sample Output 0

```
c a e
```

Contest ends in a month

Submissions: 18
Max Score: 10
Difficulty: Medium

Rate This Challenge:
☆ ☆ ☆ ☆ ☆

More

| Java 15 | ⌄ |

```java
1  import java.io.*;
2  import java.util.*;
3
4  public class Solution {
5
6      public static void main(String[] args) {
7          Scanner sc = new Scanner(System.in);
8
9
```

```java
        ArrayList<Character> ch=new ArrayList<>();
        ArrayList<Integer> dl=new ArrayList<>();
        ArrayList<Integer> pr=new ArrayList<>();


        while(sc.hasNext()){
            String parts[]=sc.nextLine().split(" ");
            ch.add(parts[0].charAt(0));
            dl.add(Integer.parseInt(parts[1]));
            pr.add(Integer.parseInt(parts[2]));

        }
        int n=dl.size();
        char charr[]=new char[n];
        int day[]=new int[n];
        int profit[]=new int[n];

        for(int i=0;i<n;i++){
            charr[i]=ch.get(i);
            day[i]=dl.get(i);
            profit[i]=pr.get(i);
        }


        int max=Integer.MIN_VALUE;
        for(int i=0;i<n;i++){
            for(int j=i+1;j<n;j++){
                if(profit[i]<profit[j]){
                    int temp=profit[i];
                    profit[i]=profit[j];
                    profit[j]=temp;

                    temp=day[i];
                    day[i]=day[j];
                    day[j]=temp;

                    char chr=charr[i];
                    charr[i]=charr[j];
```

```java
                    charr[j]=chr;

                }
            }
        }
         for (int i = 0; i < n; i++) {
            max = Math.max(day[i], max);
        }

        char arr[]=new char[max+1];

        Arrays.fill(arr, '\0');

        for(int i=0;i<n;i++){
            for(int j=day[i];j>=1;j--){
                if(arr[j]=='\0'){
                    arr[j]=charr[i];
                    break;
                }
            }
        }
        for(int i=0;i<arr.length;i++){
            if(arr[i] != '\0'){
                System.out.print(arr[i] + " ");
            }
        }
    }
}
```

Line: 1 Col: 1

Run Code   Submit Code