Full Name:	Goutham Reddy Valagolam
Email:	valagolam.g@husky.neu.edu
Test Name:	**INFO 6205 Spring 2019 Section_5 Quiz_04**
Taken On:	5 Feb 2019 16:32:23 EST
Time Taken:	30 min 44 sec/ 40 min
Work Experience:	2 years
Invited by:	Robin
Invited on:	5 Feb 2019 16:26:12 EST
Tags Score:	

70%

35/50

scored in **INFO 6205 Spring 2019 Section_5 Quiz_04** in 30 min 44 sec on 5 Feb 2019 16:32:23 EST

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	**Weighted Quick-Union** › **Multiple Choice**	2 min 14 sec	0/ 5	⊗
Q2	**Quick Find, Quick Union** › **Multiple Choice**	1 min 26 sec	5/ 5	✓
Q3	**Quick Union** › **Multiple Choice**	57 sec	5/ 5	✓
Q4	**Stability of Selection Sort** › **Subjective**	7 min 44 sec	0/ 10	⊖
Q5	**Implementation** › **Coding**	17 min 46 sec	25/ 25	✓

Weighted Quick-Union > Multiple Choice

QUESTION DESCRIPTION

What's the worst case of find(), connected() and union() method applying weighted quick-union algorithm with N sites?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ( ) N^2
- (●) N
- ( ) NlogN
- ✅ ( ) logN

No Comments

Quick Find, Quick Union > Multiple Choice

QUESTION DESCRIPTION

Statement 1 : Quick-Find union operation is too expensive
Statement 2 : Trees formed in Quick-Union are always flat
Statement 3 : Find / connected operation can be N-array access in Quick-Union, hence it is too expensive
Statement 4 : It takes O(N) array accesses to process one union operation on N objects in Quick-Find
Which statements are true ?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ( ) All of these
- ( ) Statement 1 and 3
- ( ) Statement 1 and 2
- ( ) Statement 3 and 4
- ( ) Statement 1, 2 and 4
- ( ) Statement 1, 2 and 3
- ✅ (●) Statement 1, 3 and 4
- ( ) None of these

No Comments

Quick Union > Multiple Choice

QUESTION DESCRIPTION

If we are going to apply quick-find algorithm to solve the dynamic connectivity problem until all components are connected, how many times of array operations is necessary.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

☑ ● N^2

○ NlogN

○ N

○ logN

No Comments

Stability of Selection Sort > Subjective

QUESTION DESCRIPTION

A sorting algorithm is said to be **stable** if two objects with equal keys appear in the same order in the sorted output as they appear in the input array before sorting.
So is selection sort stable? Why?
Tips: only take learned implementation into consideration.

CANDIDATE ANSWER

In selection, the sort is unstable,

because swapping a pair of objects can change the other pair of objects with equals keys

No Comments

Implementation > Coding

QUESTION DESCRIPTION

Red envelope is Chinese tradition during the Spring Festival. This year is no exception, but it's a little different. Your parents want to know whether you are working hard abroad or not. So they decide only you pass their test can you get your lucky money. Sad to have parents knowing algorithms :(
First they will give you 3 different numbers, say, representing three denominations of banknotes. Then they

tell you how much is in your red envelope and ask you how many pieces of bills **at least** in your red envelope. If it's impossible to get that number, return -1;

Here is an example:
3 denominations : 1,2,5
your luck money is 21
The answer should be 5 : 4*5+1*1. 4+1 =5
Although 10*2 +1*1 = 21, 10+1 =11 > 5.

Happy new year. (o o)

CANDIDATE ANSWER

Language used: **Java 8**

```java
public class Solution {
    // inplement this function. feel free to write other methods if you need
    // assume x,y,z is sorted and x is the smallest
    private static int process(int x, int y, int z, int target){
        int min=9999999;
        for(int i=0;i<target;i++)
        {
            for(int j=0;j<target;j++)
            {
                for(int k=0;k<target;k++)
                {
                    if((z*i+y*j+x*k)==target)
                    {
                        if(min>(i+j+k))
                        {
                         min=i+j+k;
                        }
                        else{
                            continue;
                        }



                    }
                }
            }
        }
        if(min!=9999999)
        {
            return min;
        }
        else{
        return -1;
        }
    }
}


```

TESTCASE	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	✓ Success	5	0.16 sec	25.8 MB
Testcase 1	Easy	✓ Success	5	0.15 sec	25.9 MB
Testcase 2	Easy	✓ Success	5	0.19 sec	25.7 MB

| Testcase 3 | Easy | ✓ Success | 5 | 0.17 sec | 25.7 MB |
| Testcase 4 | Easy | ✓ Success | 5 | 0.16 sec | 25.9 MB |

No Comments