

$Spring_2018_INFO6205_Sec05_Quiz_5 \ > \ shi.xiny@husky.neu.edu$

Xinying Shi Full Name:

Email: shi.xiny@husky.neu.edu

Spring_2018_INFO6205_Sec05_Quiz_5 **Test Name:**

6 Feb 2018 16:43:08 EST Taken On: Time Taken: 23 min 16 sec/ 30 min

Invited by: Robin

Invited on: 6 Feb 2018 16:33:06 EST

Tags Score:

100% 100/100

scored in Spring_2018_INFO6205_Sec05

in 23 min 16 sec on 6 Feb 2018 16:43:08 EST

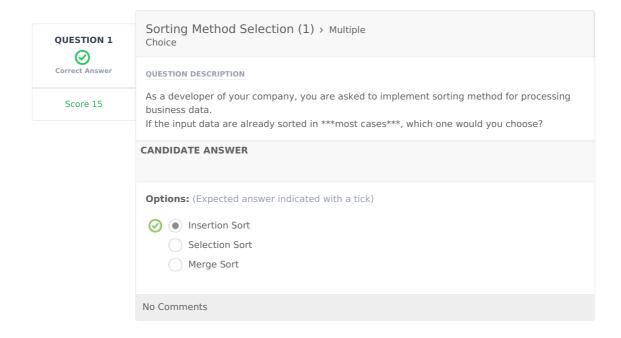
Recruiter/Team Comments:

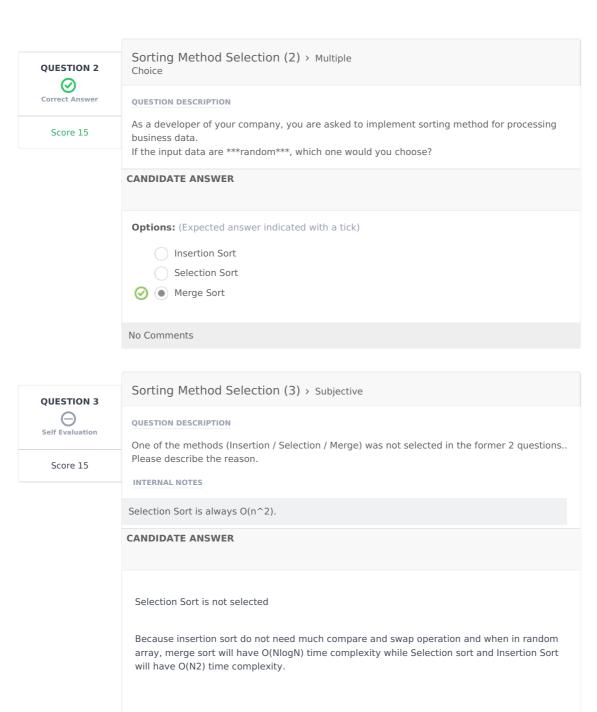
No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review.

	Question Description	Time Taken	Score	Status
Q1	Sorting Method Selection (1) > Multiple Choice	14 sec	15/ 15	\odot
Q2	Sorting Method Selection (2) > Multiple Choice	13 sec	15/ 15	\odot
Q3	Sorting Method Selection (3) > Subjective	3 min 1 sec	15/ 15	\ominus
Q4	Merge Sort > Subjective	2 min 45 sec	15/ 15	Θ
Q5	Anagram > Coding	16 min 52 sec	40/40	(!)





No Comments

QUESTION 4 Self Evaluation

Merge Sort > Subjective

QUESTION DESCRIPTION

Score 15

Please briefly describe why the complexity of Merge Sort is O(n log n).

INTERNAL NOTES

log n layers of division and n times of compare for each layer.

CANDIDATE ANSWER

On the traditional merge sort, every pass through the data will double the size of the sorted subsections.

After the first pass, the data will be sorted into sections of length two. After the second pass, length four. Then eight, etc. Finally, it will up to the size of the data.

It's necessary to keep doubling the size of the sorted sections until there's one section comprising the whole data. It will take Ig(N) doublings of the section size to reach the file size, and each pass of the data will take time proportional to the number of records.

No Comments



Score 40

Anagram > Coding

QUESTION DESCRIPTION

(An anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.)

Given two strings s and t, write a function to determine if t is an anagram of s.

```
For example,
```

```
s = "anagram", t = "nagaram", return true. 
 s = "rat", t = "car", return false.
```

Note:

You may assume the string contains only lowercase alphabets.

Hint:

- 1. There is O(n) solution for this question but your algorithm doesn't have to be O(n) as long as you can pass the test cases.
- 2. You may find toCharArray() and charAt() methods in String Class useful.
- 3. You may sort the characters in the given Strings to solve this problem.

INTERNAL NOTES

Sort the characters (or put them into hash map) first and then compare the 2 character arrays (hash maps).

CANDIDATE ANSWER

Language used: Java 8

```
public static boolean isAnagram(String s, String t) {
         // put your implementation here
 3
         HashMap<Character,Integer> map=new HashMap<>();
 4
         char[] sc=s.toCharArray();
 5
         for (char i:sc){
 6
            if (map.get(i) = = null)
 7
              map.put(i,0);
 8
            map.put(i,map.get(i)+1);
9
         \quad \text{for (int } i = 0; i < t.length(); i + +)\{
            if (map.get(t.charAt(i)) = = null||map.get(t.charAt(i)) < = 0)
12
               return false;
13
            map.put(t.charAt(i), map.get(t.charAt(i))-1);\\
14
15
         return true:
16
17
```

TESTCASE	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy 🥝	Success	8	0.17 sec	34 MB
Testcase 1	Easy 🥝	Success	8	0.17 sec	35.4 MB
Testcase 2	Easy 🥝	Success	8	0.17 sec	33.1 MB
Testcase 3	Easy 🥝	Success	8	0.16 sec	34.8 MB
Testcase 4	Easy 🥝	Success	4	0.16 sec	34.3 MB
Testcase 5	Easy 🥝	Success	4	0.16 sec	34.4 MB

No Comments