Full Name:	Wendi Yu
Email:	yu.wend@husky.neu.edu
Test Name:	**Spring_2018_INFO6205_Sec05_Quiz_5**
Taken On:	6 Feb 2018 16:45:52 EST
Time Taken:	22 min 6 sec/ 30 min
Invited by:	Robin
Invited on:	6 Feb 2018 16:33:13 EST
Tags Score:	

100%
100/100
scored in **Spring_2018_INFO6205_Sec05_** in 22 min 6 sec on 6 Feb 2018 16:45:52 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**	6 sec	15/ 15	✓
Q2	Sorting Method Selection (2) › **Multiple Choice**	2 sec	15/ 15	✓
Q3	Sorting Method Selection (3) › **Subjective**	58 sec	15/ 15	⊖
Q4	Merge Sort › **Subjective**	1 min 22 sec	15/ 15	⊖
Q5	Anagram › **Coding**	18 min 10 sec	40/ 40	!

QUESTION 1
✓
Correct Answer

Score 15

Sorting Method Selection (1) › Multiple Choice

QUESTION DESCRIPTION

As a developer of your company, you are asked to implement sorting method for processing business data.
If the input data are already sorted in ***most cases***, which one would you choose?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ✓ (●) Insertion Sort
- ( ) Selection Sort
- ( ) Merge Sort

No Comments

QUESTION 2

Correct Answer

Score 15

Sorting Method Selection (2) › Multiple Choice

QUESTION DESCRIPTION

As a developer of your company, you are asked to implement sorting method for processing business data.
If the input data are ***random***, which one would you choose?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ( ) Insertion Sort
- ( ) Selection Sort
- (✓) Merge Sort

No Comments

QUESTION 3

Self Evaluation

Score 15

Sorting Method Selection (3) › Subjective

QUESTION DESCRIPTION

One of the methods (Insertion / Selection / Merge) was not selected in the former 2 questions.. Please describe the reason.

INTERNAL NOTES

Selection Sort is always O(n^2).

CANDIDATE ANSWER

Selection Sort is not selected because when in mostly sorted array, insert sort will not need much compare and swap operation and when in random array, merge sort will have O(N log N) time complexity while Selection sort and Insertion Sort will have O(N2) time complexity.

No Comments

QUESTION 4

Self Evaluation

Score 15

Merge Sort › Subjective

QUESTION DESCRIPTION

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

Because at each iteration you split the array into two sublists, and recursively invoke the algorithm.

At best case you split it exactly to half, and thus you reduce the problem (of each recursive call) to half of the original problem. You need log_2(n) iterations, and each iteration takes exactly O(n) (each iteration is on all sublists, total size is still n), so at total O(n log n).

No Comments

QUESTION 5

Needs Review

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**

```
1  public static boolean isAnagram(String s, String t) {
2        // 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     }
10    for(int i=0; i<t.length();i++){
11       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
16      return true;
17    }
18
```

TESTCASE	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Success	8	0.17 sec	34.3 MB
Testcase 1	Easy	Success	8	0.17 sec	34.7 MB
Testcase 2	Easy	Success	8	0.17 sec	35 MB
Testcase 3	Easy	Success	8	0.17 sec	37.3 MB
Testcase 4	Easy	Success	4	0.17 sec	35.4 MB
Testcase 5	Easy	Success	4	0.16 sec	34.7 MB

No Comments