Question - 1
Quick Union

SCORE: **5 points**

If we are going to apply the quick-find algorithm to solve the dynamic connectivity problem until all components are connected, how many array accesses are necessary?

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

Question - 2
Stable Sort

SCORE: **5 points**

Consider an employee database where people are added based on hire date and then sorted based on last name. If two people in the employee database have the same last name, which sorting method(s) will not change their order relative to each other? *(Hint: this concept is known as stability)*

- ( ) Selection sort
- (●) Insertion sort
- (●) Merge Sort
- ( ) Shell Sort

Question - 3
Knuth Shuffle

SCORE: **5 points**

When shuffling a deck of cards [2,3,4,5,6,7,8,9] with "Knuth shuffle", which of the following could possibly be the situation after the first (i.e. 0^{th}) iteration?

- (●) 2,3,4,5,6,7,8,9
- ( ) 3,2,4,5,6,7,8,9
- ( ) 9,3,4,5,6,2,8,7
- ( ) 2,4,3,5,6,7,8,9

Help

Question - 4

Decide Sort

SCORE: **5 points**

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?

- (●) Insertion Sort
- ( ) Selection Sort
- ( ) Merge Sort
- ( ) All of the above

Question - 5

Anagrams

SCORE: **30 points**

Strings Sorting

(An anagram is a word or phrase formed by rearranging the letters of a different word or phrase, 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.