HackerRank For Work

INFO6205_Fall2018_Section... 30 minutes

## Question - 1

SCORE: **5 points**

**Question 1**

Which of the following is an external sorting?

- Bubble Sort
- Selection Sort
- Merge Sort
- Insertion Sort

## Question - 2

SCORE: **5 points**

**Question 2**

Which of the following are stable sorting algorithm?

- Selection Sort
- Insertion Sort
- Shell Sort
- Merge Sort

## Question - 3

SCORE: **5 points**

**Question 3**

The time complexity required to merge two sorted arrays of size m and n is

- O(m)
- O(n)
- O(logm + logn)
- O(m+n)

## Question - 4

SCORE: **5 points**

**Question 4**

Which of the following shows the correct order of algorithms in increasing order of worst case time complexity.

- Merge Sort, Shell Sort, Insertion sort, Selection Sort

- Merge Sort, Insertion sort, Shell Sort, Selection Sort
- Merge Sort, Shell Sort, Selection Sort, Insertion sort
- Insertion Sort, Merge Sort, Shell Sort, Selection Sort

## Question - 5

SCORE: **30 points**

**Find the median**

All of you know the concept of a median. If not, When a distribution is sorted, the middle value(or mean of the middle two values) is known as the median.

```
Example:
When the number of elements is odd :[1,2,4,5,7]
The median = 4


When the number of elements is even :
[1,2,4,5,7,8]
The median = 4.5
```

This program finds the median of an array. The logic for finding the median is provided. All you have to do is implement the **mergeSort() and merge()** methods.

needless to say, the mergeSort() implements the logic for the sort and merge() merges the two arrays. The array **a[] contains the actual list of elements while tempA[] is an auxiliary array for the merge sort.**

```
Sample Input :
3
8 6 7

The first line is size of the array.
The second line is the value of the integers in
the array, ie. The elements in the array.


Sample output :
7.0

The output is the median of the array [8,6,7]
(from the example input above)
```

**NOTE: you are NOT allowed to use the Arrays.sort() method.**