



# **JAVA** **Basic Sorting**

## **Assignment Questions**

**Q1.** Which of the following(s) is/are true about bubble sort:

1. It is stable sort
2. It has a worst case space complexity of  $O(n)$
3. It involves swapping of adjacent elements
4. After each iteration, the greatest element is placed at the end of the array.

**Q2.** What will the following array look like after one iteration of bubble sort [1,6,2,5,4,3].

1. [1,3,2,4,5,6]
2. [1,2,3,4,5,6]
3. [1,2,5,4,3,6]
4. [1,2,4,5,3,6]

**Q3.** In which case does bubble sort works in the most efficient way:

1. When the array is sorted in increasing order
2. When the array is sorted partially
3. When the array is sorted in decreasing order.
4. When the array is nearly sorted.

**Q4.** Sort the array in descending order using Bubble Sort.

**Q5.** Check if the given array is almost sorted. (elements are at-most one position away)

**Q6.** Which of the following is true about selection sort:

1. In each iteration we find the minimum element in the unsorted part of the array.
2. In each iteration we find the index of the minimum element in the unsorted part of the array.
3. We swap the index of the minimum element with the first element of the array.
4. It takes  $O(n^2)$  swaps.

**Q7.** Which of the following examples represent the worst case input for an insertion sort?

1. array in sorted order
2. large array
3. normal unsorted array
4. array sorted in reverse order

**Q8.** How many passes would be required during insertion sort to sort an array of 5 elements?

1. 1
2. Depends on order of elements
3. 4
4. 5

**Q9.** Given an array of digits (values are from 0 to 9), the task is to find the minimum possible sum of two numbers formed from digits of the array. Please note that all digits of the given array must be used to form the two numbers.

**Q10.** Given an array of strings `arr[]` with all strings in lowercase. Sort given strings using Bubble Sort and display the sorted array.

**Q11.** What is an in-place sorting algorithm?

1. It needs  $O(1)$  or  $O(\log n)$  memory to create auxiliary locations
2. The input is already sorted and in-place
3. It requires additional storage
4. It requires additional space

**Q12.** In the following scenarios, when will you use selection sort?

1. The input is already sorted
2. A large file has to be sorted
3. Large values need to be sorted with small keys
4. Small values need to be sorted with large keys

**Q13.** Given an integer array and an integer  $k$  where  $k \leq \text{size of array}$ , We need to return the  $k$ th smallest element of the array.

**Q14.** Find the minimum operations required to sort the array in increasing order. In one operation, you can set each occurrence of one element to 0.

**Q15.** Given an array, `arr[]` containing  $n$  integers, the task is to find an integer (say  $K$ ) such that after replacing each and every index of the array by  $|a_i - K|$  where  $(i \in [1, n])$ , results in a sorted array. If no such integer exists that satisfies the above condition then return -1.



**THANK  
YOU !**

