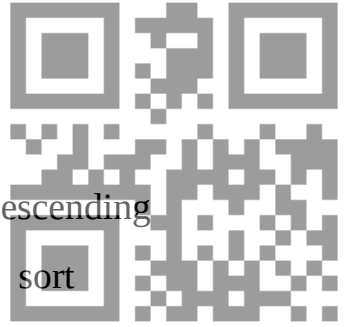




## **Sorting Quiz**



- 1- Write the bubbleSort() method code in descending order? How many comparisons does a bubble sort perform in sorting N items?
- 2- Write bubble sort algorithm that contains both bubbleSort1 and bubbleSort2 methods. Where after the first pass of bubbleSort1, the largest element is guaranteed to be at the end of the array, after the second pass, the second largest element is at second end of the array, and so on. Also, after the first pass of bubbleSort2, the smallest element is guaranteed to be at index 0, after the second pass, the second smallest is at index 1, and so on.
- 3- A bubble sort of 10,000 elements takes about 1 second on a certain computer. About how long will it take on 50,000 elements?
- 4- We need sort the following array [2 1 5 4 3 6] of integers into ascending order:
  - a- If bubble sort is chosen to sort this array, write the contents of the array each time that the sort algorithm changes it. How many comparison operations and how many swaps are performed in the sorting?

2021/2022

2021/2022

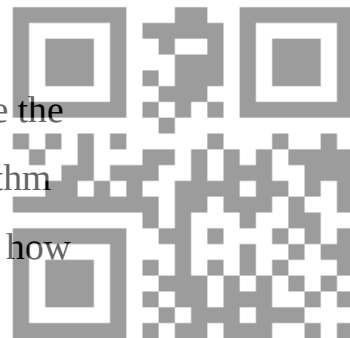
2021/2022

30206021602134

30206021602134

30206021602134





- b- If selection sort is chosen to sort this array, write the contents of the array each time that the sort algorithm changes it. How many comparison operations and how many swaps are performed in the sorting?
- c- If insertion sort is chosen to sort this array, write the contents of the array each time that the sort algorithm changes it. How many comparison operations and how many swaps are performed in the sorting?
- d- Use merge sort to sort the provided array. Be sure to show your work. For example, show the substeps of merge sort using recursion trees.

38	81	22	48	13	69	93	14	45	58	79	72
----	----	----	----	----	----	----	----	----	----	----	----

- e- Consider an array has 6 elements: [5-2-6-1-3-4] Arrange the elements in ascending order by using quick sort algorithm

2021/2022

2021/2022

2021/2022

30206021602134

30206021602134

30206021602134



# Sorting Quiz

## 1. Bubble Sort method - descending .. Show the number of comparisons in a bubble sort.

$$\# \text{comparisons} = N * (N - 1) / 2$$

```
void bubbleSort_Descending(int [] array) {
    int nElements = array.length;
    for(int i = nElements - 1; i > 0; i-- ) {
        for (int j = 0; j < i; j++) {

            // Descending --> check if array[j] is SMALLER THAN array[ j + 1 ]
            if (array[j] < array[j + 1]) {
                swap(array, j, j + 1);
            }

        }
    }
}
```

## 2. Modify the Bubble Sort method so that after

→ the first pass ( the largest element is at the end & the smallest element is at the start “index 0” )

→ the second pass ( the second largest element is at the index before the end & the second smallest element is at the “index 1” )

and so on ...

```
void bubbleSort_Descending(int [] array) {
    int nElements = array.length;
    for(int i = nElements - 1; i > 0; i-- ) {

        // makes sure the largest is at the end
        for (int j = 0; j < i; j++)
            if (array[j] > array[j + 1]) {
                swap(array, j, j + 1);
            }

        // makes sure the smallest is at the start
        for (int j = i; j > nElements - 1 - i; j--) {
            if (array[j] < array[j - 1]) {
                swap(array, j, j - 1);
            }
        }

    }
}
```

3. A bubble sort takes 1 seconds on 10,000 elements, how long will it take on 50,000 elements?

The running time in big O notation  
 $O(n^2)$

10,000 items → 1 second  
50,000 items → ?

$(10,000^2) \rightarrow 1$   
 $(50,000^2) \rightarrow ?$

$? = (50,000^2) \cdot 10 / (10,000^2) = 25 \text{ seconds}$

4. Modify bubble sort and Insertion Sort to count #comparisons & #swaps & #copies?

Can be found at the “code samples” section on the web-page of the course.

6. If bubble sort is used to sort the following array [ 2 1 5 4 3 6 ] in an ascending order, show the array contents each time the algorithm changes it.

How many comparisons and swaps are performed here ... in this case .. not in general

1	2	5	4	3	6
1	2	4	5	3	6
1	2	4	3	5	6
1	2	3	4	5	6

Number of COMPARISONS is: 15

Number of SWAPS is: 4

6. If Insertion sort is used to sort the following array [ 2 1 5 4 3 6 ] in an ascending order, show the array contents each time the algorithm changes it.

How many comparisons and swaps are performed here ... in this case .. not in general

2	2	5	4	3	6
1	2	5	4	3	6
1	2	5	5	3	6
1	2	4	5	3	6
1	2	4	5	5	6
1	2	4	4	5	6
1	2	3	4	5	6