Assignment 7

Name : Hrishikesh Rajan
Email : hrishikeshrajan3@gmail.com

---

4. An almost sorted array is given to us and the task is to sort that array completely. Then,which sorting algorithm would you prefer and why?[Salesforce]

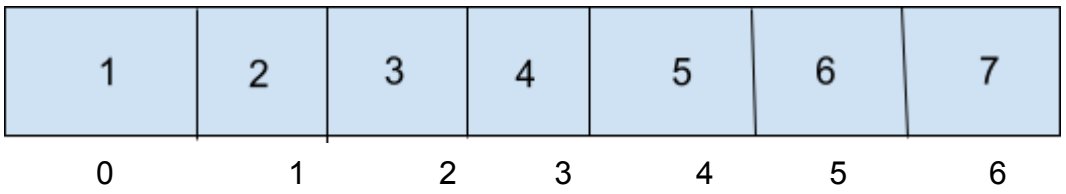
CODE

```
function insertionSort(arr,n){


    for(let current = 1; current < n; current++){
       let temp = arr[current];
       let previous ;
       for(previous  = current - 1;  previous >= 0 &&  arr [ previous ] > temp; previous–){
                arr[ previous +1] = arr [ previous ];
        }

        arr[ previous + 1 ] = temp

    }
    return arr;

}
```

EXPLANATION:

Insertion sort will be the best options

The time complexity for insertion in the worst case is O (n ^ 2)
In the worst case scenario we have to compare and swap with previous elements O (n) times.



In this case

Current = 1, value = 2

Parent = Current - 1 , value = 1

When we compare 2 < 1 ? false, So a this stage no swap , so we skipped swap that can cause O(n^ 2) comparison and swap to it's previous elements in worst case Each iteration follows the same pattern as shown below,

2) 3 <2
3) 4<3
4) 5<4
5) 6<5
6) 7<6

We didn't swap or compare the previous elements for these 6 iterations. It is already a small value. The comparison till the base index will not work if swap does not occur. As a result, the nested loop program works like a normal loop program

Hence total time complexity is O (n) for **best case**