

LAB 04: Sorting Algorithms

CS211 – Data Structures and Algorithms

Usman Institute of Technology

Fall 2019

- **How to submit:**

- Create an account on <http://www.turnitin.com/> as a Student (if you don't have already)
- Use following information at time of sign-up

- CS Section A**

- Class ID: 22664649
 - Enrollment Key: DSFALL19CSA

- CS Section B**

- Class ID: 22664651
 - Enrollment Key: DSFALL19CSB

A. Create a class Sorting and write the functions in the following order.

1. Add a class member *data* of list type. You can do this by using the following code:

```
class Sorting:  
    data = []
```

2. Write a function **Print()** that prints the values stored in data variable.

```
def Print(self, n):  
    print(self.data)
```

3. Write a function **GenerateRandom()** that generates a list of 'n' random numbers and store in data.

```
def GenerateRandom(self, n):  
    // your code goes here
```

4. Write a function ***BubbleSort()*** that sorts numbers stores in the data variable using Bubble Sort algorithms.

```
def BubbleSort(self):  
    // your code goes here
```

Example: RandomNumbers = [4,5,77,2,80,3]
SortedList = [2,3,4,5,77,80]

5. Write a function ***InsertionSort()*** that sorts the numbers stores in the data variable using Insertion Sort algorithm.

```
def InsertionSort(self):  
    // your code goes here
```

Example: RandomNumbers = [4,5,77,2,80,3]
SortedList = [2,3,4,5,77,80]

```
procedure Insertion sort(A)  
  
for i = 1 to A.length  
    key = A[i]  
    //Insert A[i] into the sorted sequence A[1.. i-1]  
    j = i - 1  
    while j>=0 and A[j]>key  
        A[j+1] = A[j]  
        j = j - 1  
  
    A[j+1] = key
```

Source: Introduction to Algorithms by Thomas H.Cormen

6. Write a function *SelectionSort()* that sorts the numbers stored in data variable using Selection Sort algorithms.

```
def SelectionSort(self):  
    // your code goes here
```

Example: RandomNumbers = [4,5,77,2,80,3]
SortedList = [2,3,4,5,77,80]

```
procedure selection sort  
    list: array of items  
    n: size of list  
  
    for i = 1 to n - 1  
        /* set current element as minimum*/  
        min = i  
  
        /* check the element to be minimum */  
  
        for j = i+1 to n  
            if list[j] < list[min] then  
                min = j;
```

```
        end if
    end for

    /* swap the minimum element with the current element*/
    if indexMin != i then
        swap list[min] and list[i]
    end if
end for

end procedure
```

Source: Tutorials Point

7. Write a function ***Search()*** that takes parameter *v* and returns the location of *v* in the list. The search should take place in $O(\)$ time. This can achieve by using Binary Search but for this, your function must check whether list is already sorted or not. If the list is not sorted, then it should call any of the sorting functions created in above tasks to sort the list.

```
def Search(self,value):
    // your code goes here
```

Example: List = [4,5,77,4,2,80,3,5]
SortedList = [2,3,4,5,77,80]
Value for Searching = 5
Location = 3