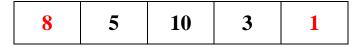
Lab 16: Learn to sort array using elementary sorting algorithms

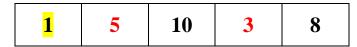
Sorting: Sorting is the process of arranging the data in a particular order.

Time complexity of Bubble sort, Selection sort and Insertion sort is n^2

Selection Sort Algorithm:



Step 1:



Step 2:



Step 3:



Step 4:



Selection Sort Code:

```
‡include<iostream>
using namespace std;
/oid selectionSort(int arr[], int n){
    for(int i=0; i<n-1; i++){
        int smallestIdx = i; //unsorted part starting
       for(int j=i+1; j<n; j++){</pre>
            if(arr[j] < arr[smallestIdx]){</pre>
                smallestIdx = j;
       swap(arr[i], arr[smallestIdx]);
/oid printArray(int arr[], int n){
    for(int i=0; i<n; i++){
       cout<<arr[i]<<" ";
int main (){
   int n = 5; //Array size
   int arr[]={8,5,10,3,1}; //unsorted array
   selectionSort(arr, n);
   printArray(arr, n);
   return 0;
```

```
1 3 5 8 10
------
Process exited after 0.557 seconds with return value 0
Press any key to continue . . .
```

Insertion Sort Algorithm: Step 1: Step 2: Step 3: Step 4:

Insertion Sort Code:

Bubble Sort Algorithm: In bubble sort we compare adjacent elements. 1st iteration: Step 1: Step 2: Step 3: **Step 4: Step 5:**

Note: After 1st iteration we can see that 1st largest element is moved to Last of the array.

2nd iteration:

Step 1:

	1	3	2	6	9
Step 2:					
	1	3	2	6	9
Step 3:					
	1	2	3	6	9
Step 4:					
	1	2	3	6	9

Note: Our loop will execute 4 times means iterations will be 4 as total elements are n = 5 so number of iterations will be n-1 = 4. But in above condition our array is sorted within 2 iterations it doesn't means that loop exits after 2 iterations, loop will continue to execute upto 4 iterations. But in 4^{th} iteration our number of steps will be reduced to 2.

Bubble Sort Code:

```
using namespace std;
roid bubbleSort(int arr[], int n){
   for(int i=0; i<n-1; i++){</pre>
       bool isSwap = false;
       for(int j=0; j<n-i-1; j++){</pre>
           if(arr[j] > arr[j+1]){      //Comparing adjusent elements
               swap(arr[j], arr[j+1]); //Pushing large element to the last
               isSwap = true;
       if(!isSwap){//array is already sorted
   printArray(int arr[], int n){
   for(int i=0; i<n; i++){</pre>
       cout<<arr[i]<<" ";
  main (){
   int arr[]={6,1,3,2,9}; //unsorted array
   bubbleSort(arr, n);
   printArray(arr, n);
```

```
1 2 3 6 9
------Process exited after 0.4733 seconds with return value 0
Press any key to continue . . .
```