27/08/2020

**Experiment No:1**

**BUBBLE SORT**

**AIM:**

To perform bubble sort in an array and to arrange the elements of the

array in ascending order.

**DATA STRUCTURE USED:**

Array is the data structure used.

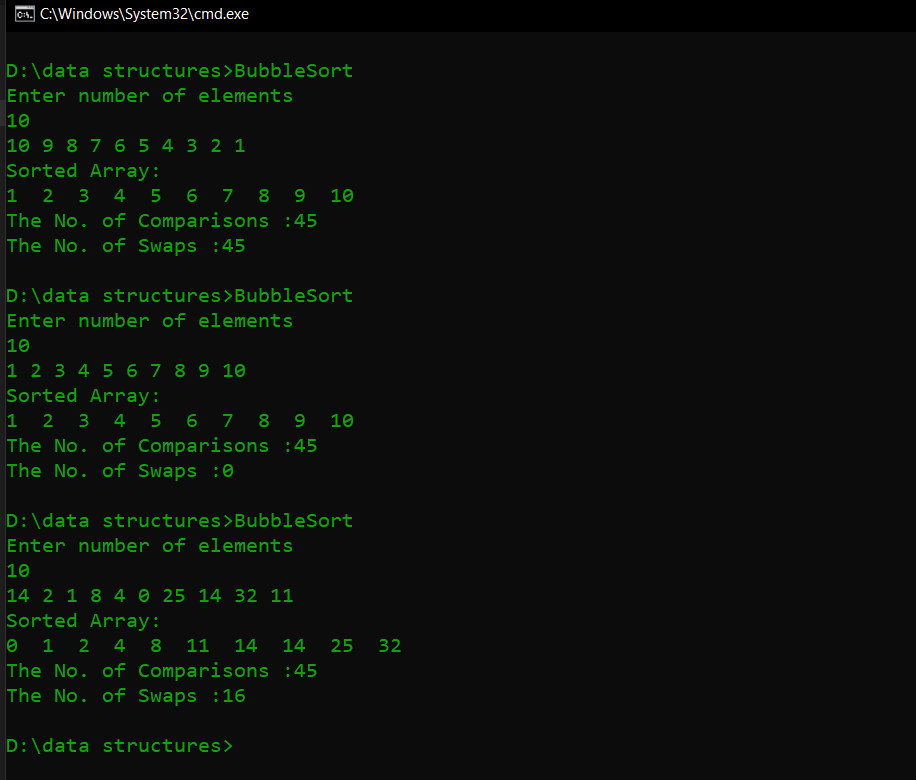
**ALGORITHM:**

1. Declare the array arr[size] and read the values of array
2. for i=0 to size-1
3. for j=0 to size-i-1
4. if arr[j]>arr[j+1]
5. swap arr[j] and arr[j+1]
6. endif
7. endfor
8. endfor
9. print the sorted array

**PROGRAM:**

#include <stdio.h>  
void main(){  
 int arr[100], i,j, n,temp,s=0,c=0;  
  
 printf("Enter number of elements\n");  
 scanf("%d", &n);  
  
 for (i=0; i<n; i++){  
 scanf("%d", &arr[i]);  
 }  
 for (i=0 ; i<n-1;i++){  
 for (j=0 ; j<n-i-1;j++){  
 c++;  
 if (arr[j]>arr[j+1]){  
 temp=arr[j];  
 arr[j]=arr[j+1];  
 arr[j+1]=temp;  
 s++;  
 }  
 }  
 }  
  
 printf("Sorted Array:\n");  
 for (i=0;i< n;i++){  
 printf("%d ", arr[i]);  
 }  
 printf("\nThe No. of Comparisons :%d\n", c);  
 printf("The No. of Swaps :%d\n", s);  
}

OUTPUT:



**RESULT:**

Bubble sort was performed in the array and the array elements were arranged in

ascending order. Also, the number of comparisons and swaps performed were found out. Number of comparisons performed was found to be n(n-1)/2 where n is the number of array elements (Except for best case).

Time complexity:

Best case – O(n)

Average case – O(n2)

Worst case – O(n2)

27/08/2020

**Experiment No:2**

**SELECTION SORT**

**AIM:**

To perform selection sort in an array and to arrange the elements of the array in

ascending order.

**DATA STRUCTURE USED:**

Array is the data structure used.

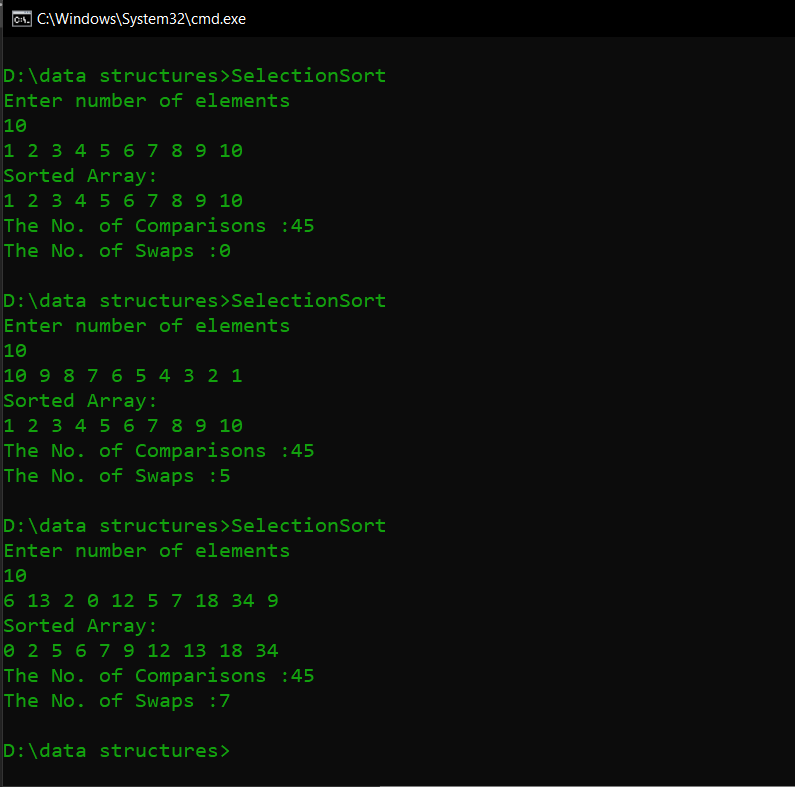
**ALGORITHM:**

1. Read the elements of the array arr[size]
2. for i=0 to size
3. pos=i
4. for j= i+1 to size
5. if arr[pos]>arr[j]
6. pos=j
7. end if
8. endfor
9. if pos !=j
10. swap arr[j] and arr[pos]
11. endif
12. endfor

**PROGRAM:**

#include <stdio.h>  
void main(){  
 int arr[100], n,i, j, pos,c=0,s=0,temp;  
  
 printf("Enter number of elements\n");  
 scanf("%d",&n);  
 for (i=0;i< n;i++){  
 scanf("%d", &arr[i]);  
 }  
 for (i=0;i<n-1;i++){  
 pos = i;  
 for (j=i+1;j<n;j++){ c++;  
 if (arr[pos]> arr[j])  
 pos=j;  
 }  
 if (pos!=i){  
 temp = arr[i];  
 arr[i] = arr[pos];  
 arr[pos] = temp;  
 s++;  
 }  
 }  
  
 printf("Sorted Array:\n");  
  
 for (i=0;i<n; i++){  
 printf("%d ",arr[i]);  
 }  
 printf("\nThe No. of Comparisons :%d\n", c);  
 printf("The No. of Swaps :%d\n", s);  
 return 0;  
}

**OUTPUT:**



**RESULT:**

Selection sort was performed in the array and the array elements were arranged in

ascending order. Also, the number of comparisons and swaps performed were found out. Number of comparisons performed was found to be n(n-1)/2 where n is the number of array elements.

Time complexity:

Best case – O(n2)

Average case – O(n2)

Worst case – O(n2)