**WORKSHEET 1.2**

**1. Aim:**

Write a program to implement Bubble sort along with its complexity analysis.

**2. Problem Description:**

To implement Bubble Sort along with its complexity analysis.

**3. Algorithm:**

* Traverse from left and compare adjacent elements and the higher one is placed at right side.
* In this way, the largest element is moved to the rightmost end at first.
* This process is then continued to find the second largest and place it and so on until the data is sorted.

**4. Computational Complexity:-**

* **Best Case Complexity -** It occurs when there is no sorting required, i.e. the array is already sorted. The best-case time complexity of bubble sort is **O(n).**
* **Average Case Complexity -** It occurs when the array elements are in jumbled order that is not properly ascending and not properly descending. The average case time complexity of bubble sort is **O(n2).**
* **Worst Case Complexity -** It occurs when the array elements are required to be sorted in reverse order. That means suppose you have to sort the array elements in ascending order, but its elements are in descending order. The worst-case time complexity of bubble sort is **O(n2).**

**5. Pseudo Code :-**

bubbleSort(array)

n = length(array)

repeat

  swapped = **false**

**for** i = 1 to n - 1

**if** array[i - 1] > array[i], then

         swap(array[i - 1], array[i])

         swapped = **true**

         end **if**

   end **for**

   n = n - 1

 until not swapped

end bubbleSort

**6. Source Code:**

#include<iostream>

**using** **namespace** std;

**void** print(**int** a[], **int** n) //function to print array elements

    {

**int** i;

**for**(i = 0; i < n; i++)

    {

       cout<<a[i]<<" ";

    }

    }

**void** bubble(**int** a[], **int** n) // function to implement bubble sort

 {

**int** i, j, temp;

**for**(i = 0; i < n; i++)

    {

**for**(j = i+1; j < n; j++)

        {

**if**(a[j] < a[i])

            {

                temp = a[i];

                a[i] = a[j];

                a[j] = temp;

            }

        }

    }

 }

**int** main()

{

**int** i, j,temp;

**int** a[5] = {45, 1, 32, 13, 26};

**int** n = **sizeof**(a)/**sizeof**(a[0]);

    cout<<"Before sorting array elements are - \n";

    print(a, n);

    bubble(a, n);

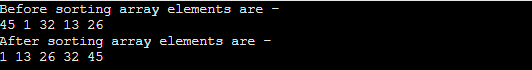
    cout<<"\nAfter sorting array elements are - \n";

    print(a, n);

**return** 0;

}

1. **Screenshot of Output:**



1. **Learning & Outcomes:**

* Learned about the linear sorting Algorithm, how it works, How much is the time complexity.
* Learned to create dynamic array using pointers.