Rajan Gautam 19BCP101

Div. II, CE 19 SOT, PDPU

Pandit Deendayal Petroleum University School of Technology

Design & Analysis of Algorithm (20CP209P)

B. Tech - Computer Science & Engineering (Sem-IV)

Table of Contents

Lab 1 Assignment: Comparison of Insertion Sort & Bubble Sort	1
AIM: To write a C/C++ Program to implement Insertion Sort & Bubble Sort	
ALGORITHMS:	
1. Insertion Sort Algorithm (Pseudocode)	
2. Bubble Sort Algorithm (Pseudocode)	1
CODE:	1
OUTPUT:	4
COMPARISON:	4

Lab 1 Assignment: Comparison of Insertion Sort & Bubble Sort

AIM: To write a C/C++ Program to implement Insertion Sort & Bubble Sort.

ALGORITHMS:

1. Insertion Sort Algorithm (Pseudocode)

```
INSERTION-SORT (A, n)

for j \in 2 to n

Do key \in A[j]

i \in j-1

while i > 0 and A[i] > key

Do A[i+1] \in A[i]

i \in i-1

A[i+1] = key
```

2. Bubble Sort Algorithm (Pseudocode)

```
BUBBLE-SORT (A, n)

For i \leftarrow 1 to N do

For j = 0 to N – 1 do

If A[j] > A[j+1] then

Temp = A[j]
A[j] = A[j+1]
A[j+1] = temp
```

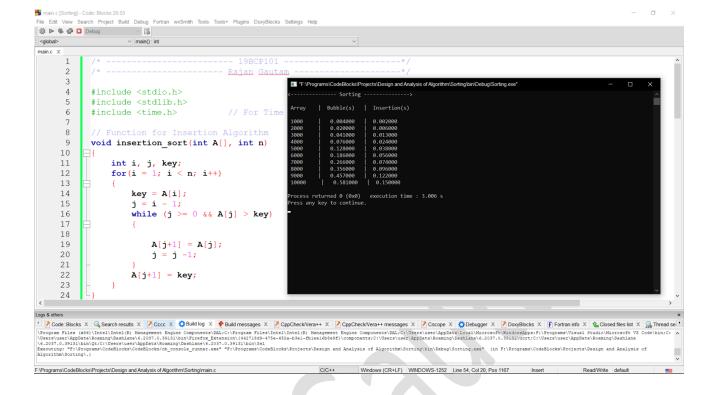
CODE:

```
1. /* -----*/
2. /* ------*/
```

```
3.
4. #include <stdio.h>
5. #include <stdlib.h>
6. #include <time.h>
                             // For Time Calculation
7.
8. // Function for Insertion Algorithm
9. void insertion sort(int A[], int n)
10. {
11.
        int i, j, key;
12.
        for(i = 1; i < n; i++)
13.
        {
14.
            key = A[i];
15.
            j = i - 1;
16.
            while (j \ge 0 \&\& A[j] > key)
17.
18.
19.
                A[j+1] = A[j];
20.
                j = j -1;
21.
22.
            A[j+1] = key;
23.
24. }
25.
26. // Function for Bubble Sort Algorithm
27. void bubble sort(int A[], int n)
28. {
29.
        int i, j, temp;
30.
        for (i = 0; i < n-1; i++)
31.
32.
          for (j = 0; j < n-i-1; j++)
33.
34.
                if (A[j] > A[j+1])
35.
36.
                    temp=A[j];
37.
                    A[j]=A[j+1];
38.
                    A[j+1]=temp;
39.
40.
41.
42. }
43.
44.
45. int main()
46. {
47.
        printf("<---->\n\n");
48.
49.
       int n = 1000, it = 0;
```

```
50.
  values
51.
       printf(" Array | Bubble(s) | Insertion(s) \n\n");
52.
53.
54.
       while (it++ < 10)
55.
        {
56.
           long int a[n], b[n];
57.
           for (int i = 0; i < n; i++)
58.
               // Generating Random Integer Array for each algorithm
59.
60.
61.
               a[i] = (rand() % n);
62.
               b[i] = (rand() % n);
63.
           }
64.
65.
           // For time calculation
66.
67.
           clock t start, end;
68.
69.
70.
           // For Bubble Sort Algorithm
71.
           start = clock();
72.
           bubble sort(a, n);
73.
           end = clock();
74.
75.
           time1[it] = ((double) (end - start) / CLOCKS PER SEC);
76.
77.
78.
         // For Insertion Algorithm
79.
           start = clock();
80.
           insertion sort(b, n);
81.
           end = clock();
82.
83.
           time2[it] = ((double)(end - start)/CLOCKS PER SEC);
84.
           // Printing the table of array size, time taken by Bubble
85.
  Sort and Insertion Algorithm
           printf(" %d
                        | %f | %f\n", n, time1[it],
86.
  time2[it]);
87.
88.
89.
           // Incrementing the value of n by 1000
90.
           n += 1000;
91.
       }
92.
       return 0;
93. }
```

OUTPUT:



COMPARISON:

