



# ***Bangladesh University of Business and Technology***

## **Lab Report**

**Course Title: Algorithms Lab.**

**Course No: CSE 242.**

**Prepared By**

**Submitted To**

**Student's Name: Samin Sakif.**

**Md. Ahsanul Haque**

**ID: 16173103077.**

**Lecturer,**

**Program: B.Sc. Engg. in CSE.**

**Department of CSE, BUBT.**

**Intake: 36.**

**Mirpur-2, Dhaka -1216.**

**Section: 2.**

**Date of Submission:     /     /     .**

**Q1:** Write a code to sort an array using **insertion sort** algorithm. Write another code to sort an array using **selection sort** algorithm and also show their time complexity with a graph.

❖ Code to sort an array using **Insertion sort** algorithm:

```

1. #include <bits/stdc++.h>
2. #define MAX 200000
3. using namespace std;
4. long long j, i;
5. void insertionSort(long long A[], long long n)
6. {
7.     long long i = 1;
8.     while(i < n)
9.     {
10.         long long x = A[i];
11.         long long j = i - 1;
12.         while(j >= 0 && A[j] > x)
13.         {
14.             A[j+1] = A[j];
15.             j = j - 1;
16.         }
17.         A[j+1] = x;
18.         i = i + 1;
19.     }
20. }
21. int main()
22. {
23.     freopen("input.txt", "r", stdin);
24.     freopen("output.txt", "w", stdout);
25.     long long A[MAX];
26.     for(i=0; i<15; i++)
27.     {
28.         long long n;
29.         cin >> n;
30.         for(j=0; j<n; j++)
31.             A[j] = rand();
32.         clock_t time = clock();

```

```

33.     insertionSort(A, n);
34.     time = clock() - time;
35.     int ms = double(time) / CLOCKS_PER_SEC * 1000;
36.     cout<<ms << endl;
37.     }
38.     return 0;
39.     }

```

❖ Code to sort an array using **Selection sort** algorithm:

```

1.#include <bits/stdc++.h>
2.#define MAX 200000
3.using namespace std;
4.long long j, i;
5.void selectionsort(long long A[], long long n)
6.{
7.    long long i, j, min, temp;
8.    for(i=0; i<n-1; i++)
9.    {
10.        min = i;
11.        for(j=i+1; j<n; j++)
12.        {
13.            if(A[j] < A[min])
14.                min = j;
15.        }
16.        temp = A[i];
17.        A[i] = A[min];
18.        A[min] = temp;
19.    }
20. }
21. int main()
22. {
23.     freopen("input.txt", "r", stdin);
24.     freopen("output.txt", "w", stdout);

```

```

25.     long long A[MAX];

```

```

26.    for(i=0; i<15; i++)
27.    {
28.        long long n;
29.        cin >> n;
30.        for(j=0; j<n; j++)
31.            A[j] = rand();
32.        clock_t time = clock();
33.        selectionsort(A, n);
34.        time = clock() - time;
35.        int ms_selection = double(time) /
            CLOCKS_PER_SEC * 1000;
36.        cout<<ms_selection << endl;
37.    }
38.    return 0;
39.    }

```

### **Numbers of Input**

1. 10
2. 100
3. 500
4. 1500
5. 2500
6. 4000
7. 6000
8. 9000
9. 15000
10. 40000
11. 50000
12. 70000
13. 85000
14. 95000
15. 100000

### **Required time for individual input**

<b><u>Insertion sort</u></b>	<b><u>Selection sort</u></b>
1. 0	1. 0
2. 0	2. 0
3. 0	3. 0
4. 0	4. 0
5. 15	5. 15
6. 0	6. 15
7. 31	7. 47
8. 62	8. 109
9. 156	9. 297
10. 1156	10. 2141
11. 1766	11. 3328
12. 3484	12. 6532
13. 5094	13. 9500
14. 6438	14. 11782
15. 7187	15. 13062

## Graph:

