TUTORIAL:-3

Name:- Krishna Pandey

Admission No:-U20CS110

➢ Insertion Sort RAM Model

Input :- A sequence of n numbers.

      <a1,a2,a3,….an>

Output :- A sequence of n numbers

      <a'1, a'2, a'3, …. a'n>

      Where a'1 ≤ a'2 ≤ a'3 ≤…. a'n.

Algorithm:-

	Insertion Sort(A)	Cost of each instruction
1.	for i=2 to A.lenght	C1
2.	Key = A[i]	C2
3.	j = j - 1	C3
4.	while j > 0 and A[j] > key	C4
5.	A[j+1] = A[j]	C5
6.	j = j - 1	C6
7.	A[j+1] = key	C7

Execution time:-

	Best	Worst
1.	n	n
2.	n – 1	n – 1
3.	n – 1	n – 1
4.	$\sum\limits_{i=2}^{n}1$	$\sum\limits_{i=2}^{n}i$
5.	0	$\sum\limits_{i=2}^{n}(i-1)$
6.	0	$\sum\limits_{i=2}^{n}(i-1)$
7.	1	n – 1

CALCULATION:-

* OAA Tutorial:3 *

& T(n) for Best case:-

$$T(n) = c_1(n) + c_2(n-2) + c_3(n-2) + c_4\left(\sum_{i=1}^{n}1\right) + c_7(n-3)$$

$$= c_1 n + c_2 n - c_2 + c_3 n - 1 + (4(n-1) + c_7(n-1)$$

$$= (c_1 + c_2 + c_3 + c_4 + c_7)n + (c_2 - c_2 - c_3 - c_4 - c_7)$$

$$= \boxed{an + b} \quad \text{where} \quad a \,\&\, b \text{ are constant}$$

$$\Rightarrow \boxed{T(n) = an + b} \Rightarrow \text{best case}$$

* T(n) for worst case:-

$$T(n) = c_1(n) + c_2(n-1) + c_3(n-1) + c_4\left(\sum_{i=2}^{n}i\right) + c_5\left(\sum_{i=2}^{n}(i-2)\right) + c_6\left(\sum_{i=2}^{n}(i-1)\right) + c_7(n-1)$$

For $\displaystyle\sum_{i=2}^{n}i = \frac{(n-1)(n+2)}{2} = A$ / for $\displaystyle\sum_{i=2}^{n}(i-1) = \frac{n(n-1)}{2} = A$

$$\Rightarrow T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4\frac{(n-1)(n+1)}{2} + c_5\frac{(n)(n-1)}{2} + c_6\frac{(n)(n-1)}{2} + c_7(n-1)$$

$$= c_1 n + c_2 n - c_2 + c_3 n - c_3 + c_4\frac{n^2}{2} + c_4\frac{n}{2} - c_4 + c_5\frac{n^2}{2} - c_5\frac{n}{2} + c_6\frac{n^2}{2} - c_6\frac{n}{2}$$
$$+ c_7 n - c_7$$

$$= \left(\frac{c_4 + c_5 + c_6}{2}\right)n^2 + \left(\frac{2c_1 + 2c_2 + 2c_3 + c_4 - c_5 - c_6 + 2c_7}{2}\right)n + (c_1 - c_2 - c_3 - c_5)$$

$$= an^2 + bn + c \quad \text{where } a, b, c \text{ are constant}$$

$$\Rightarrow \boxed{T(n) = an^2 + bn + c} \Rightarrow \text{for worst case}$$

CODE:-

```cpp
#include <bits/stdc++.h>
using namespace std;
#define rep(i, n) for (int i = 0; i < n; i++)
typedef long long ll;

void insertionSort(vector<ll> v)
{
    for (ll i = 0; i < v.size(); i++)
    {
        ll j = i-1;
        ll ritesh = v[i];
        while (j>=0 && v[j]>ritesh)
        {
            v[j+1]=v[j];
            j--;
        }
        v[j+1]=ritesh;
    }
}

int main()
{
    freopen("File 1.txt", "r", stdin);
    freopen("out1.txt", "a", stdout);
    vector<ll> v;
    ll ele;
    while (cin >> ele)
    {
        v.push_back(ele);
    }
    auto t1 = std::chrono::high_resolution_clock::now();
    insertionSort(v);
    auto t2 = std::chrono::high_resolution_clock::now();
    auto duration = std::chrono::duration_cast<std::chrono::nanoseconds>(t2 - t1).count();
    cout << "Time taken by File 1 is : " << duration << "ns" << endl;
    return 0;
}
```

Provide the details of Hardware/Software you used to implement the algorithm and to measure the time.

Ritesh-Sakheta
HP Pavilion Gaming Laptop 15-ectxxx

Rename this PC

ⓘ Device specifications

Copy ∧

Device name	Ritesh-Sakheta
Processor	AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz
Installed RAM	8.00 GB (7.36 GB usable)
Device ID	AE464E57-3835-4508-9969-F0AB7F93BF67
Product ID	00327-36291-26081-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Related links   Domain or workgroup   System protection   Advanced system settings

⊞ Windows specifications

Copy ∧

| Edition | Windows 11 Home Single Language |
Version	21H2
Installed on	05-12-2021
OS build	22000.376
Experience	Windows Feature Experience Pack 1000.22000.376.0

Microsoft Services Agreement
Microsoft Software License Terms

Average case time:-

≡ out1.txt

1   Time taken by File 1 is : 4604000ns
2   Time taken by File 2 is : 47045000ns
3   Time taken by File 3 is : 1055858000ns
4   Time taken by File 4 is : 14658961000ns
5   Time taken by File 5 is : 280304891000ns
6   Time taken by File 6 is : 3294100373000ns
7

GRAPH FOR AVERAGE CASE
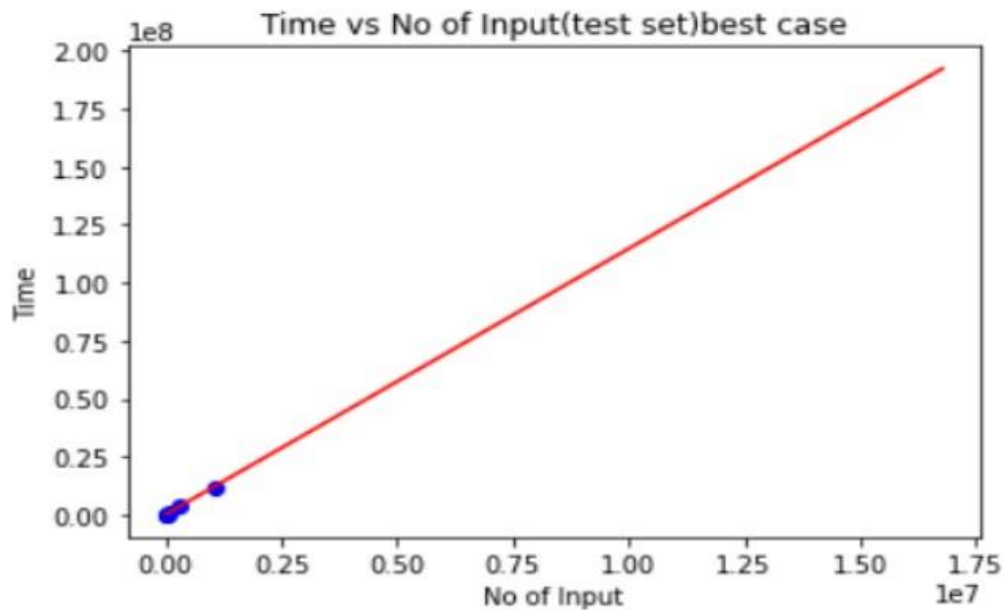

Time vs No of Input(test set)
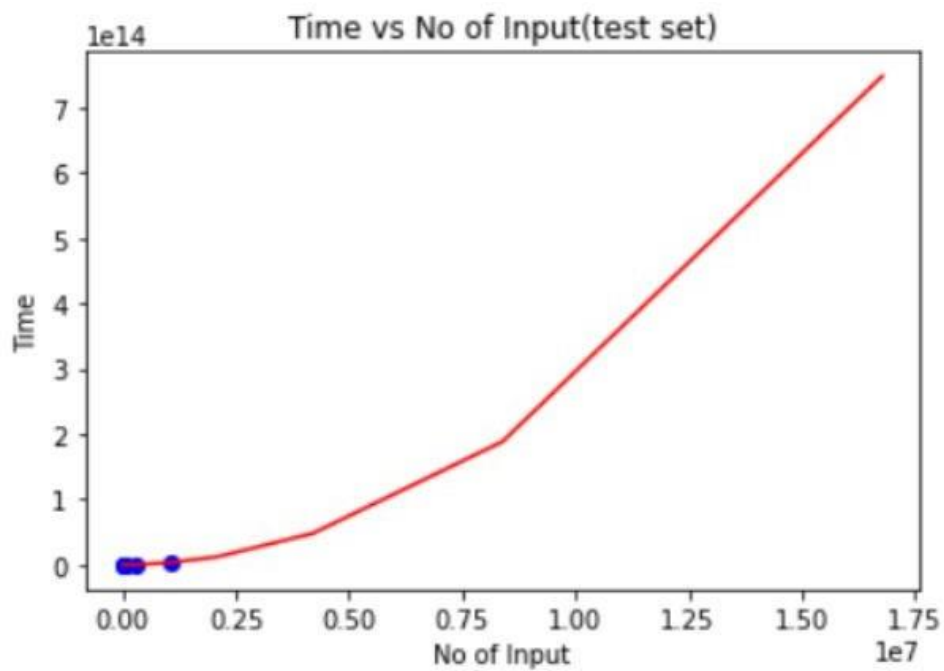
Best case time:-

```
≡ bestout1.txt
 1    Time taken by File 1 is : 0ns
 2    Time taken by File 2 is : 0ns
 3    Time taken by File 3 is : 0ns
 4    Time taken by File 4 is : 1021000ns
 5    Time taken by File 5 is : 4001000ns
 6    Time taken by File 6 is : 11966000ns
 7
```
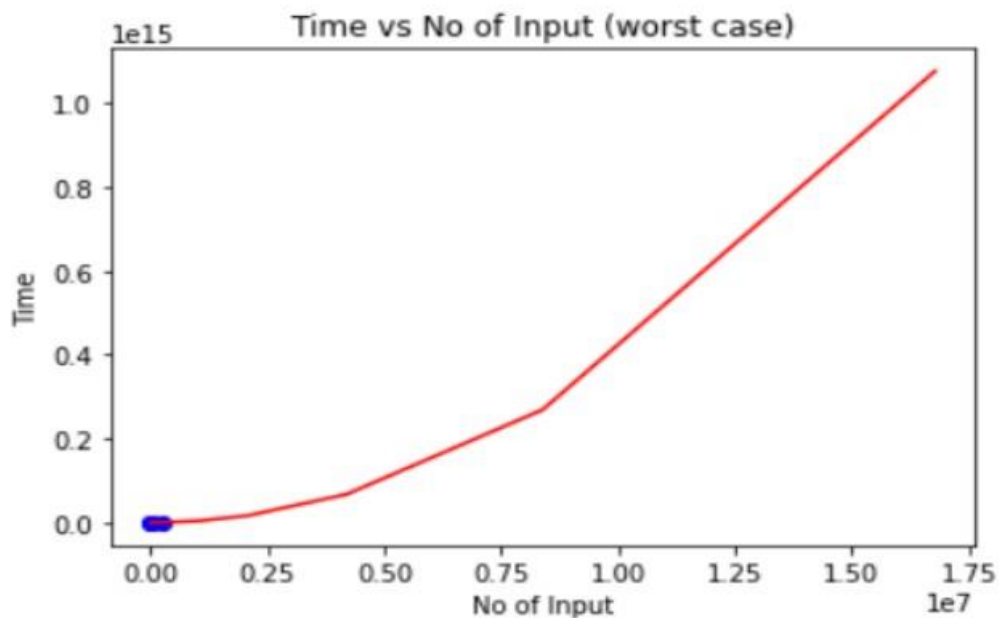
Worst case time:-

```
worseout1.txt
1    Time taken by File 1 is : 18316000ns
2    Time taken by File 2 is : 142965000ns
3    Time taken by File 3 is : 1057514000ns
4    Time taken by File 4 is : 16633601000ns
5    Time taken by File 5 is : 263454684000ns
6    Time taken by File 6 is : 4835273805000ns
7
```
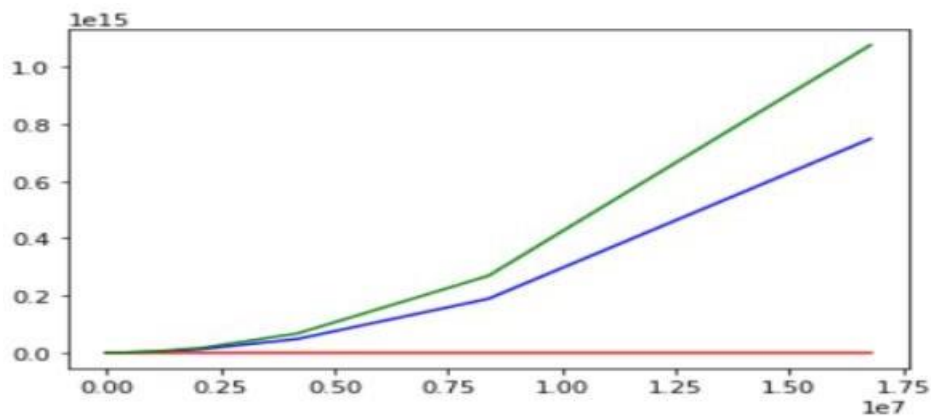
GRAPH FOR WORST CASE

2. Assume that you don't know the time complexity of above algorithms.

2.1 Can you predict the same based on your implementation?

By observing Graphs of each case:

Best case => Straight Line => So equation will be: An+B

Average Case/Worst Case => Parabolic graph => So equation will be: An^2+Bn+C

2.2 Do they match with theoretical time complexity? Yes/No.

⇒ Yes, it is…..

2.3 If yes, then write the time complexity of each algorithm. If no, then write the difference.

⇒ Time Complexity:

⇒ Best Case: An+B => BIG THETA(n)

⇒ Worst Case: An^2+Bn+C => BIG THETA(n^2)