Practical No 1

Aim : Write A Program To Check Time Complexity Of Different Algorithm For Sum Of n Natural Number.

Algorithm 1: Using For Loop

```
def sum (n):
    sum = 0
    startTime = time.perf_counter()
    for i in range(1,n+1):
        sum += i
    endTime = time.perf_counter()
    timeUsed = endTime - startTime
    print("The Sum Of Natural Number From 1 To ",n,"Is ",sum)
    print("Time Used ",timeUsed)
```

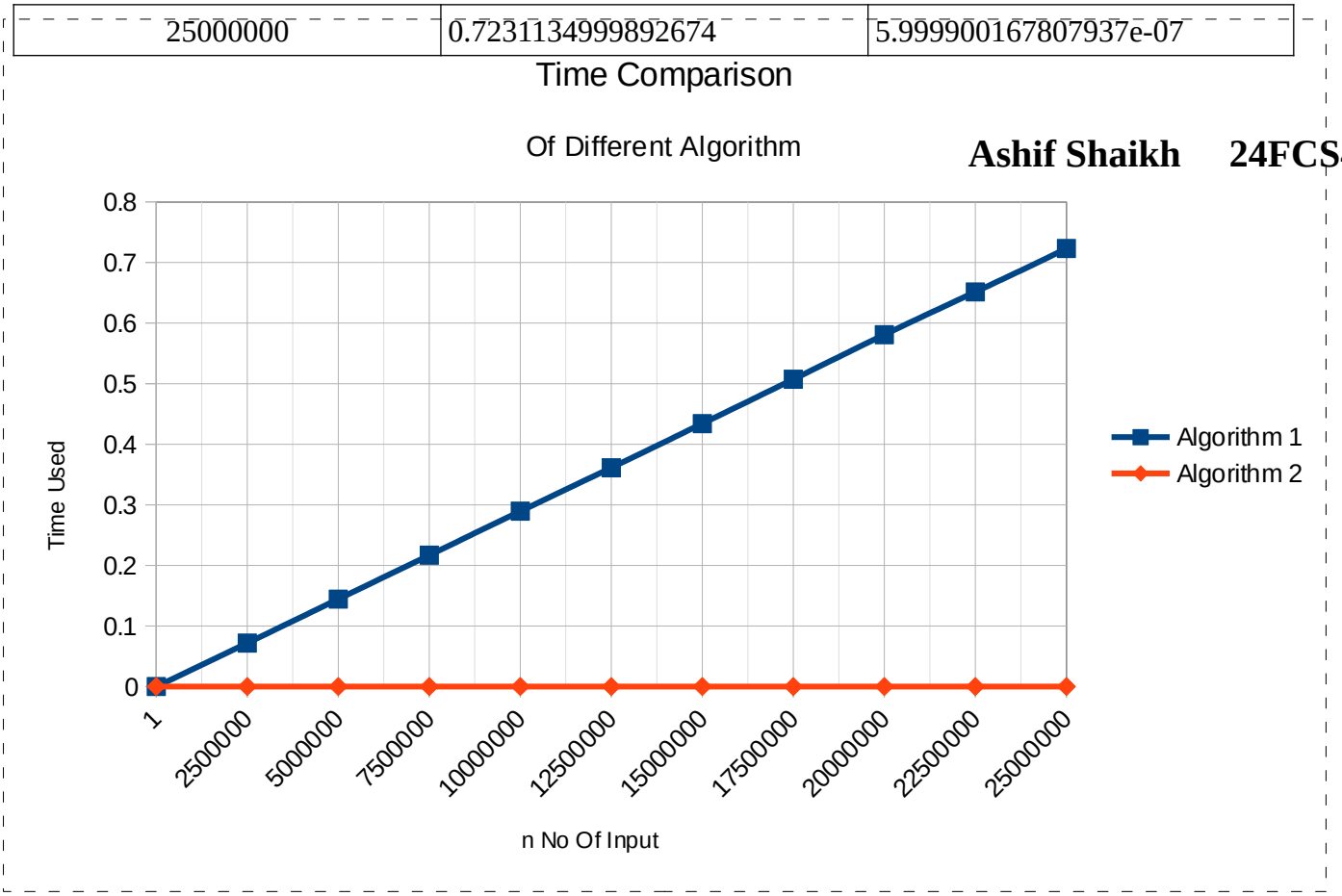
Algorithm 2: Using Formula

```
def sum (n):
    startTime = time.perf_counter()
    sum = (n*(n+1)/2)
    endTime = time.perf_counter()
    timeUsed = endTime - startTime
    print("The Sum Of Natural Number From 1 To ",n,"Is ",sum)
    print("Time Used ",timeUsed)
```

Comparison

n	Algorithm 1	Algorithm 2
1	6.00004568696022e-07	5.999900167807937e-07
2500000	0.07176560000516474	9.999930625781417e-07
5000000	0.1443721999967238	9.999930625781417e-07
7500000	0.21683170000324026	8.00006091594696e-07
10000000	0.289438899999368	6.00004568696022e-07
12500000	0.3612107999942964	7.00005330145359e-07
15000000	0.4340580999996746	7.00005330145359e-07
17500000	0.5072498000081396	6.00004568696022e-07
20000000	0.5808796000055736	7.00005330145359e-07
22500000	0.6514788999920711	7.00005330145359e-07

25000000	0.7231134999892674	5.999900167807937e-07



Time Complexity

Algorithm 1 :

- Used Loop For Calculating Sum Of n Natural Number.
- Time Increased As The Input Increases
- Thus Time Complexity Is **O(n)**

Algorithm 2 :

- Used Formula For Calculating Sum Of n Natural Number.
- Time Increased Is Negligible As The Input Increases
- Thus Time Complexity Is **O(1)**