Practical-4

Implement a function of sequential search and count the steps executed by function on various inputs for best case and worst case. Also write complexity in each case and draw a comparative chart.

Code:

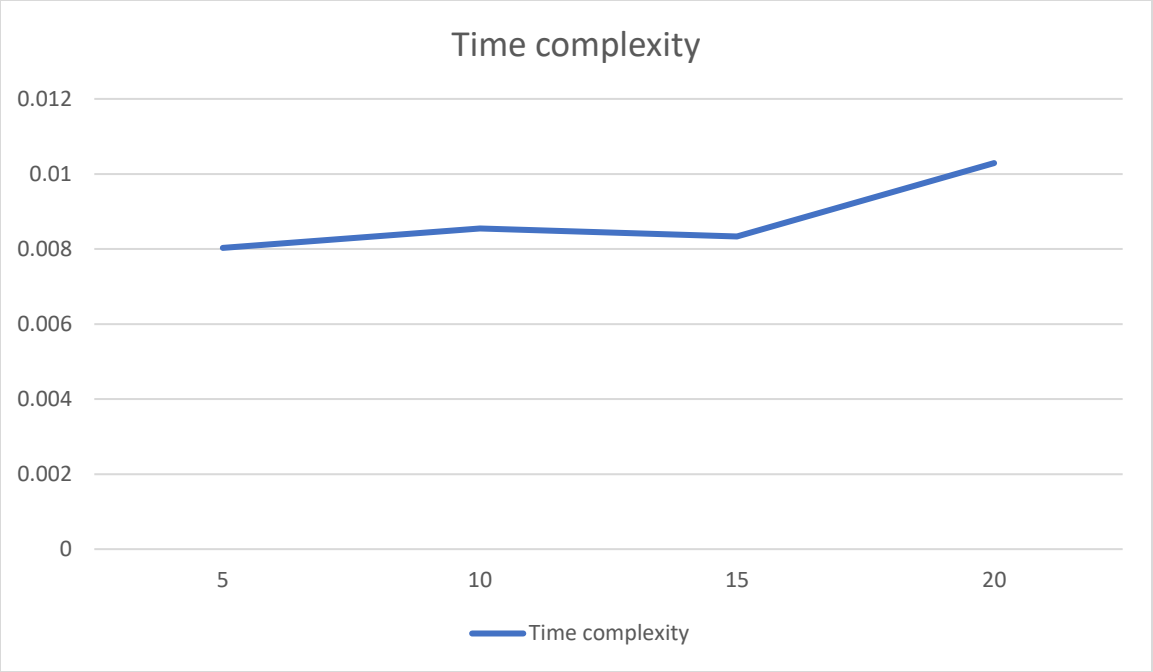
```
#include <iostream>
using namespace std;
void LinearSearch(int arr[], int len, int item){
 for(int i=0;i<len;i++){
 if(arr[i] == item){
 cout << item << " Found at index : " << i;
 return;
 }
 }
 cout << "Not found";

}
int main() {
 int arr[] = {10, 5, 15, 21, 30, 7};

 // calculating length of array
 int len = sizeof(arr)/sizeof(arr[0]);
```

```
 // item to be searched
 int item = 21;
 LinearSearch(arr, len, item);
 return 0;
}
```

Output:

BEST CASE:

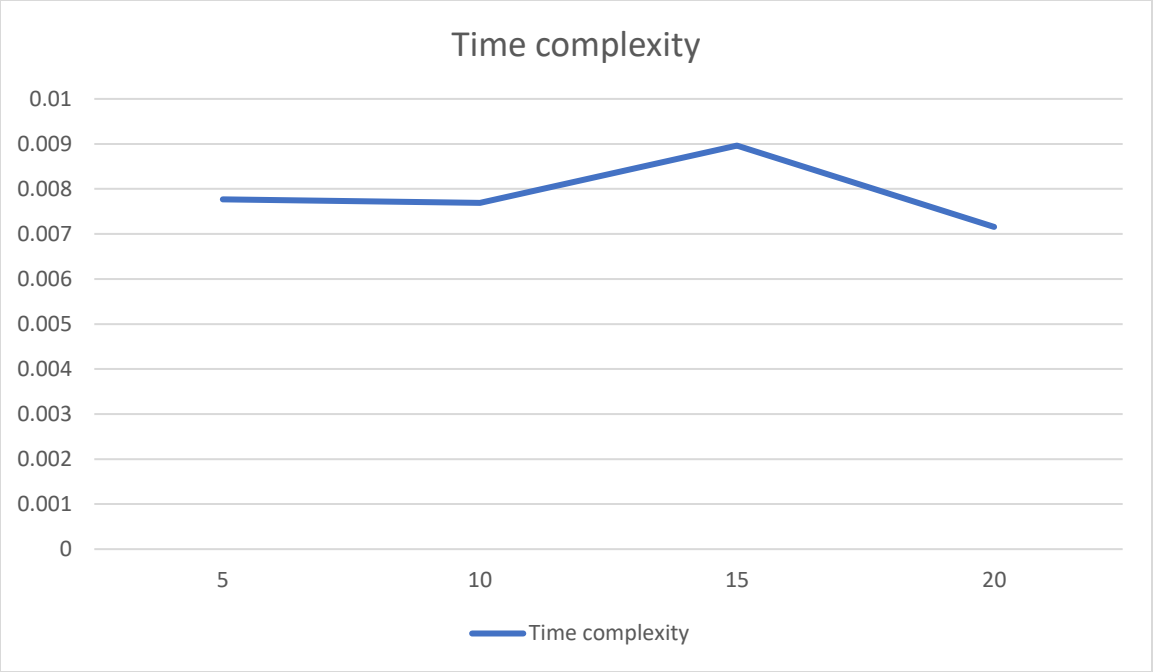
Size of Array	Time complexity
5	**0.008029**
10	**0.008545**
15	**0.00833**
20	**0.010289**



WORST CASE:

Size of Array	Time complexity
5	**0.007772**
10	**0.007694**
15	**0.008963**
20	**0.007157**

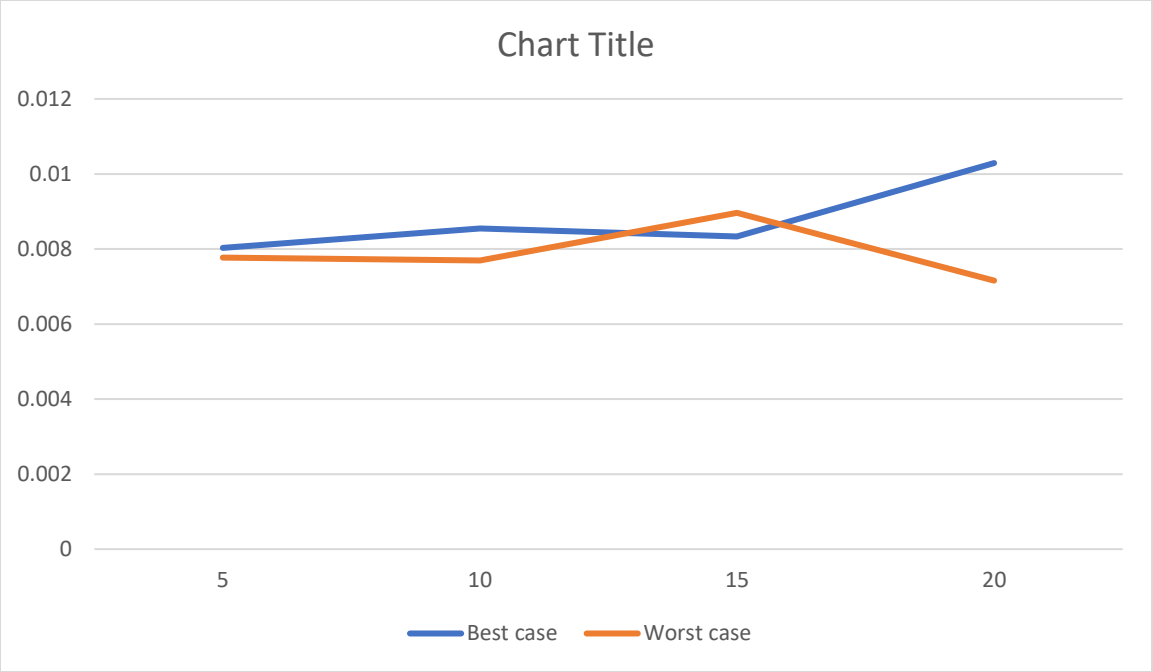
NAME :- Patel Vandan R. **BATCH :- AB3**
ENR NO :- 20012011130



COMPARISION CHART:

Size of Array	Best case	Worst case
5	0.008029	0.007772
10	0.008545	0.007694
15	0.00833	0.008963
20	0.010289	0.007157

NAME :- Patel Vandan R. **BATCH :- AB3**
ENR NO :- 20012011130



CONCLUSION:

IN THIS PRACTICAL WE EXAMINED THE TIME COMPLEXITY OF SEQUENTIAL SEARCH. FURTHER WE ANALYSED THE BEST AND WORST CASE TIME COMPLEXITY FOR LINEAR SEARCH AND DRAWN A COMPARTIVE GRAPH FOR BOTH CASE.

NAME :- Patel Vandan R. **BATCH :- AB3**
ENR NO :- 20012011130