



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



DEPARTMENT OF INFORMATION TECHNOLOGY

Academic Year 2023-24

COURSE CODE: DJS22ITC403

DATE: 6.02.24

COURSE NAME: Design and Analysis of Algorithms Laboratory

CLASS: II-1

NAME: Archie Mehta

SAP:60003220125

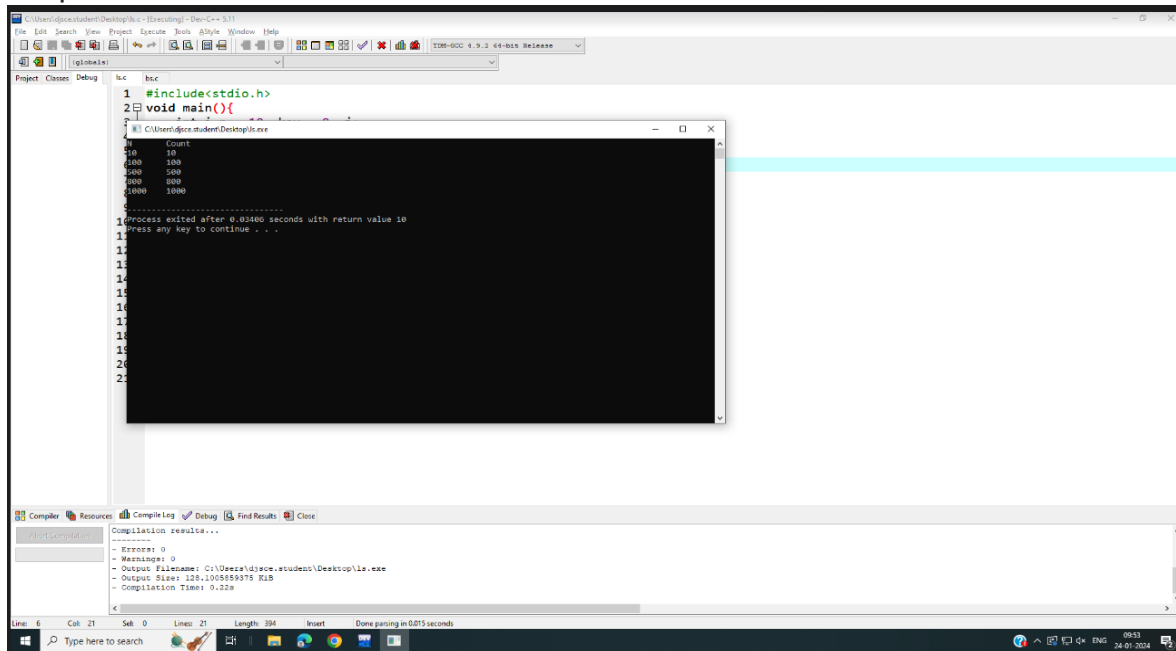
ROLL NO: I015

EXPERIMENT NO. 1

AIM / OBJECTIVE: Complexity analysis for Linear search.

```
#include<stdio.h>
void main(){
int i,n = 10, key = 9, j;
printf("N\tCount\n");
int size[5] = {10, 100, 500, 800, 1000};
for(j = 0;j <= 4;j++){
int n = size[j];
int arr[n];
int count = 0;
for(i = 0;i < n;i++){
arr[i] = 2 * i;
for(i = 0;i < n;i++){
if(arr[i] == key){
printf("%d", i);
break;
}
count++;
}
printf("%d\t%d\n",n, count);
}
}
```

Output:



```
1 #include<stdio.h>
2 void main(){
3     Count
4     10
5     100
6     500
7     800
8     1000
9     10000
10
11     Process exited after 0.03460 seconds with return value 10
12     Press any key to continue . . .
```

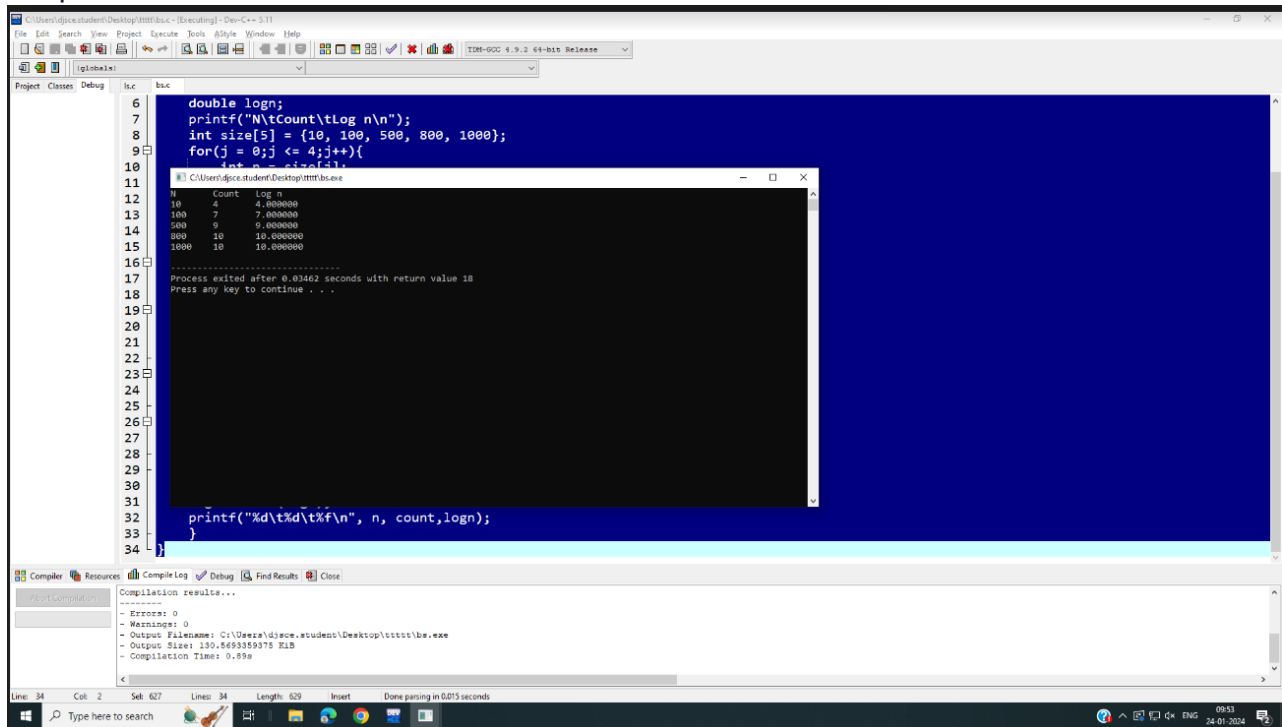
Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\jacew.student\Desktop\Le.exe
- Output Size: 128,100,859,979 KB
- Compilation Time: 0.22s

2)Time Complexity for Binary Search

```
#include<stdio.h>
#include<math.h>
void main(){
int n = 10, i, key = 3, j;
long count;
double logn;
printf("\n\tCount\tLog n\n");
int size[5] = {10, 100, 500, 800, 1000};
for(j = 0;j <= 4;j++){
int n = size[j];
int arr[n];
int left = 0, right = n;
for(i = 0;i < n;i++)
arr[i] = 2 * i;
count = 0;
while(left <= right){
int m = (left + right) / 2;
count++;
if(arr[m] == key){
printf("%d", i);
break;
}
else if(arr[m] > key){
right = m - 1;
}
else{
left = m + 1;
}
```

Output:



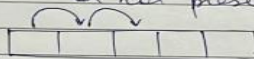
Experiment 1

* Linear Search

Linear search / sequential search is the most basic search technique. In this type of search, you go through the entire list and try to fetch a match for a single element. If you find a match then the address of the matching target element is returned.

Complexity:

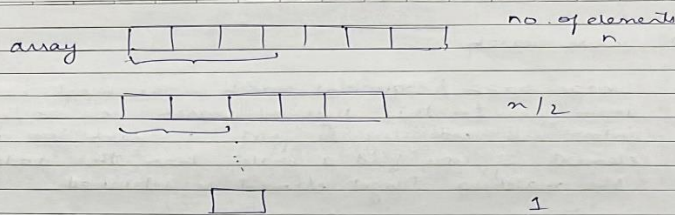
- 1) Best case: When the element to be found is the first element.
- 2) Average case: Element to be searched is in the middle of the array.
- 3) Worst case: Element to be found is at the last index or not present at all.

 size = n iteration = n
Complexity $O(n)$

* Binary Search

The Binary Search algorithm is used to search for an item from a sorted list of items. It works by repeatedly dividing in half the portion of the list that could contain the item until there is only one location to check if the value of item matches the target.

FOR EDUCATIONAL USE



FOR EDUCATIONAL USE