



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

NAME: GAURAV KISHOR PATIL

DIV: 2 ROLL NO:54

BATCH: C

Experiment No.10
Implementation of Binary Search Method
Date of Performance:
Date of Submission:



Experiment No. 10: Binary Search Method

Aim : Implementation of Binary Search Method

Objective: 1) Understand how to implement Binary Search algorithm.

Theory:

The improvement to searching method to reduce the amount of work can be done using binary searching. Binary searching is more efficient than linear searching if an array to be searched is in sorted manner.

Here an key item to be searched is compared with the item at middle of array. If they are equal search is completed. If the middle element is greater than key item searching proceeds with left sub array. Similarly, if middle element is less than key item than searching proceeds with right sub array and so on till the element is found.

For large arrays, this method is superior to sequential searching.

Algorithm

Algorithm : FIND(arr, x, first, last)

if (first > last)then

return -1

End if

mid = (first + last) / 2

if (arr[mid] = x)

return mid

End if

if (arr[mid] < x)

return find(arr, x, mid+1, last)

End if

return find(arr, x, first, mid-1)

Code:



```
#include<stdio.h>
#include<conio.h>

int binary_search(int a[],int n,int limit){
int mid,low,high;
low=0;
high=limit-1;

while(low<=high){
mid=(low+high)/2;
if(a[mid]==n){
return mid;

}
else if(n<a[mid]){
high=mid-1;
}
else if(n>a[mid]){

low=mid+1;
}
}
return 0;
}

void main(){
int arr[20];
int n,ele,f,i,j,temp;
clrscr();
printf("Enter the number of element you want ot insert:\n ");
scanf("%d",&n);
printf("Enter the number of element:\n ");
for(i=0;i<n;i++){
scanf("%d",&arr[i]);
}
for(i=0;i<=n-2;i++){
for(j=0;j<=n-2;j++){
if(arr[j]>arr[j+1]){
temp=arr[j];
arr[j]=arr[j+1];
arr[j+1]=temp;
}
}
}
printf("Enter the element you want ot search:\n ");
scanf("%d",&ele);
f=binary_search(arr,ele,n);
if(f==0){

}
else{
```



```
printf("Element is found at the index %d\n",f);  
;  
}  
getch();  
  
}
```

Output:

```
Enter the number of element you want ot insert:  
5  
Enter the number of element:  
12  
45  
66  
74  
2  
Enter the element you want ot search:  
66  
Element is found at the index 3  
-
```



Conclusion:

Time Complexity: $O(\log n)$ – Average/worst case time complexity would be $O(1)$ for the best case of time complexity.

In conclusion, the implementation of the Binary Search algorithm in the C programming language provides an efficient and widely used method for searching elements within a sorted array. By repeatedly dividing the search space in half, Binary Search quickly locates the target element with a time complexity of $O(\log n)$, making it suitable for large datasets. Its simplicity and effectiveness make it a fundamental tool in computer science and a go-to choice for searching and retrieval tasks, offering significant performance advantages over linear search methods.