



Name: Wahid Shaikh Roll no: 38

Experiment No 10: Binary Search Method

Aim: Implementation of Binary Search Method.

Objectives :

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 :

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 binarySearch(int arr[], int l, int r, int x)
```

```
{
```

```
    clrscr();
```

```
    while (l <= r) {
```

```
        int m = l + (r - l) / 2;
```

```
        if (arr[m] == x)
```

```
            return m;
```

```
        if (arr[m] < x)
```

```
            l = m + 1;
```

```
        else
```

```
            r = m - 1;
```

```
    }
```

```
    return -1;
```

```
}
```

```
int main(void)
```

```

{
    int arr[] = { 2, 3, 4, 10, 40 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int x = 10;
    int result = binarySearch(arr, 0, n - 1, x);
    (result == -1) ? printf("Element is not present"
                           " in array")
                  : printf("Element is present at "
                           "index %d",
                           result);

    getch();
    return 0;
}

```

Output :

```
Element is present at index 3
```

Conclusion :

In conclusion, the binary search algorithm is a powerful technique for efficiently locating elements in sorted arrays. Its logarithmic time complexity makes it an attractive choice for applications where quick search times are essential, especially in scenarios involving large datasets. The C program provided successfully demonstrates the implementation and application of this algorithm.