

AOA EXPERIMENT 3

Name: Mohammed Arhaan Ali

Div: S11

Roll No: 02

AIM: To study and implement Binary Search algorithm.

THEORY:

Binary Search is an interval searching algorithm used to search for an item in the sorted list. It works by repeatedly dividing the list into two equal parts and then searching for the item that is the part where it can possibly exist.

Unlike linear search, there are a few conditions for applying binary search:

1. The list must be sorted.
2. Random access to the list member.

It means that we cannot apply the binary search in unsorted or linked data structures.

Algorithm for Binary Search in C

Let x be the element we are searching for and the array is sorted in the ascending order.

1. Compare x with the middle element of the array.
2. If x matches with the middle element, we return the index of the middle element.
3. Else if x is greater than the middle element, it means that x can only lie in the right half subarray after the middle element. So we repeat steps 1 and 2 for the right half subarray and leave the left half subarray from the algorithm
4. Else if x is smaller than the middle element, it means that x can only lie in the left half subarray before the middle element. So, we

repeat steps 1 and 2 for the left half subarray.

5. We will keep doing that till we find x or there is no element left in the subarray being considered.

As we can see, the binary search ignores the half elements of the subarray after each pass. Due to this, it is able to reduce the time required for searching the element in the array as compared to linear search.

CODE:

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

```
int binarySearch(int array[], int x, int low, int high) {
```

```
    while (low <= high) {  
        int mid = low + (high - low) / 2;
```

```
        if (array[mid] == x)  
            return mid;
```

```
        if (array[mid] < x)  
            low = mid + 1;
```

```
    else  
        high = mid - 1;  
}
```

```
    return -1;  
}
```

```
int main(void) {
```

```
    int n;  
    printf("Enter number of elements:");  
    scanf("%d", &n);
```

```

int arr[n];

printf("Enter elements in array:");
for (int i = 0; i < n; i++)
{
    scanf("%d", &arr[i]);
}

int x;
printf("Enter the element to be searched: ");
scanf("%d",&x);
int result = binarySearch(arr, x, 0, n - 1);
if (result == -1)
    printf("Not found");
else
    printf("Element is found at index %d", result);
return 0;
}

```

OUTPUT:

```

PS C:\Users\arhaa\OneDrive\Desktop\A0A> cd "c:\Users\arhaa\OneDrive\Desktop\A0A\" ; if ($?) { gcc binary
search.c -o binarysearch } ; if ($?) { .\binarysearch }
Enter number of elements:10
Enter elements in array:0 10 20 30 40 50 60 70 80 90
Enter the element to be searched: 50
Element is found at index 5
PS C:\Users\arhaa\OneDrive\Desktop\A0A> █

```

CONCLUSION: Thus we have successfully implemented Binary Search Algorithm.