

Binary Search

A. Sorted Array

B. Two pointer

Val = 65

arr =

5	12	25	37	42	49	52	56	65	72	84	92	100
0	1	2	3	4	5	6	7	8	9	10	11	12

i mid j

$$\begin{aligned} \text{mid} &= \frac{i+j}{2} \\ &= \frac{0+12}{2} \\ &= \underline{6} \end{aligned}$$

```

if(arr[mid] < val)
    i = mid + 1;
else if(arr[mid] > val)
    j = mid - 1;
else
    return mid;

```

Diagram illustrating a binary search on a sorted array: [56, 65, 72, 84, 92, 100]. The array is shown in a grid with indices 7 to 12 below. The element 72 is circled, and an arrow labeled 'mid' points to it. A blue box highlights the first three elements (56, 65, 72).

$$mid = \frac{i+j}{2} = \frac{7+12}{2} = \frac{19}{2} = 9$$

$$\begin{aligned} \text{mid}_2 &= \frac{7+8}{2} \\ &= \frac{15}{2} \\ &= 7 \end{aligned}$$

```
int mid = (i + j) / 2;
if (arr[mid] == val)
    return mid;
else if (arr[mid] < val)
    i = mid + 1;
else
    j = mid - 1;
```

$$\begin{aligned} \text{mid} &= \frac{8+8}{2} \\ &= 8 \end{aligned}$$

$j \leq i$ $j \quad i$

arr =	5	12	25	37	42	49	52	56	65	72	84	92	100
	0	1	2	3	4	5	6	7	8	9	10	11	12

i mid j

while($i \leq j$)

{

int mid = $(i + j) / 2$;

if(arr[mid] == val) ~~X~~

return mid;

else if(arr[mid] < val)

$i = mid + 1$;

else

$j = mid - 1$;

}

return -1;

val = 60

65 < 60 X

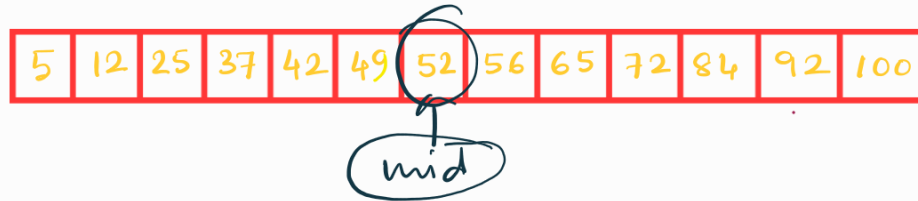
Space and Time Complexity

```

int BiSearch(int a[], int n, int val)
{
    int i=0, j=n-1;
    while(i<=j)
    {
        int mid = (i+j)/2;
        if(a[mid]==val)
            return mid;
        else if(a[mid]<val)
            i=mid+1;
        else
            j=mid-1;
    }
    return -1;
}
    
```

Space Complexity : $O(1)$

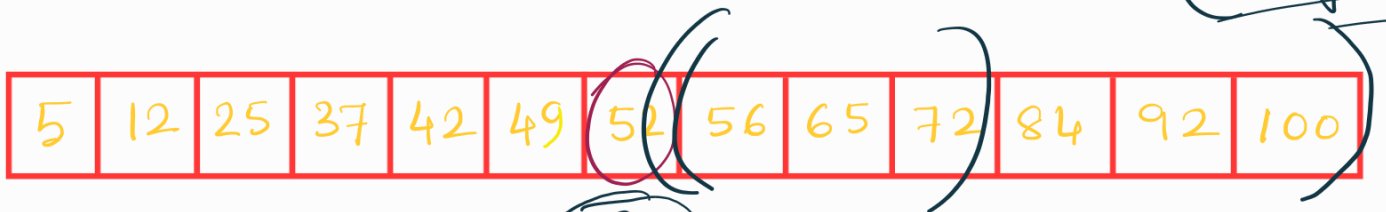
Time Complexity :-



Best Case : $O(1)$

Average and Worst Case

$O(\log n)$



$O(k) \Rightarrow$

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$\log n = \log(2^k)$$

$$\log n = k \log 2$$

$$\log n = k$$

$$n/2 \rightarrow n/2^1$$

$$n/4 \rightarrow n/2^2$$

$$n/8 \rightarrow n/2^3$$

$$n/16 \rightarrow n/2^4$$

$$n/2^k = 1 \Rightarrow n/2^k = 1$$

$$k = \log n$$

$$\underline{\underline{O(\log n)}}$$