Worst Time complexity is 0(logn).-> ye tab hota h jab loop na ho code me.

Best Time complexity is 0(1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 4 | 5 | 8 | 10 |

Here we find X=10.

**Step -1** Take start and end.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Index | 0 | 1 | 2 | 3 | 4 |
| Element | 2 | 4 | 5 | 8 | 10 |

Start =0 mid end = 4

**Step – 2** find the middle(mid) element.

#include <bits/stdc++.h>

using namespace std;

bool BinarySearch(int a[], int target, int n){

int start = 0, end = n - 1;

while (start <= end){

int mid = (start + end) / 2;

if (a[mid] == target){

return true;

}else if (a[mid] > target){

end = mid - 1;

}else if (a[mid] < target){

start = mid + 1;

} } return false;

}

int main(){

int n;

cin >> n;

int a[n];

for (int i = 0; i < n; i++){

cin >> a[i];

}int target;

cin >> target;

bool ans = BinarySearch(a, target, n);

if (ans == true){

cout << "Target found !!!" << endl;

}else{

cout << "Target not found !!!" << endl;

}return 0;

}

Mid = (start+end)/2

Mid = (0+4)/2 = 2

**Conditions:**

1. **Arr[mid] =x -> true**
2. **Arr[mid] >x ->end = mid-1**
3. **Arr[mid] <x ->start = mid+1**

|  |  |
| --- | --- |
| **3** | **4** |
| **8** | **10** |

**Step -3**  Then

Again find mid = (3+4)/2 = 3

Then again check conditions and soon.

At last, we find the given element i.e. 10

Worse case = 0(log2n)