Linear Search : → In Linear Search, we search an element or value in a given array by travers the array from the starting, till the desired element or value is found.

↳ It is used ~~to~~ for unsorted and unordered small list of elements.

↳ It has a time complexity of O(n), which means the time is linearly dependent on the number of elements, which is not bad, but not that good too.

[0]	[1]	[2]	[3]	[4]	[5]	[6]
10	12	8	4	5	6	15

Suppose we are looking for 15 (key element = 15).
So, we start searching the key element from starting of the array, till the key value is found.
No. of comparision = 7.
If Number of data = n.
in this case Maximum No. of comparision possible = n
So, Time complexity : O(n).

↳ It has a very simple implementation.

Binary search.

—x—x

↳ Binary search is a searching technique used to search data.

↳ Binary search Technique is applicable only for Sorted list of data.

↳ Binary search Technique based on Divide and conquer approach.

↳ Time complexity: $O(\log_2 n)$

↳ In Binary Search, $\log_2 n$ no. of Passes are required.

↳ It is useful when there are large number of elements in an array.

Binary Search Algorithm Searches an element by comparing ~~an element~~ it with the middle most element of the array.

Mid = $\left(i+j/_2\right)$ where i is the starting index of the array and j is the last index of the array.

following Three cases are Possible.

↳ case1: if the elements being searched is found to be the middle most element, its index is returned.

↳ case2: if the element being searched is found to be greater than the middle most element, then its search is further continued in the right sub array of the most element.

↳ case3: if the element being searched is found to be smaller than the middle most element, then its search is further continued in the left sub array of the middle most element.

This iteration keeps on repeating on the sub arrays until the desired element is found. or size of the sub array reduces to zero.

Algorithm

```
Binarysearch (A, i, j, val)
{
    if (i ≤ j)
    {
        Mid = ⌈(i+j)/2⌉
        if val == A[Mid] return Mid
        else if val < A[Mid]
                return Binary search(A, i, mid-1, val);
            else
                return Binary search (A, mid+1, j, val)
    }
    return Not - found
}.
```

I/P. A

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
5	6	8	9	10	12	14	19

Key element = 11
or
value

Binary search (A, 0, 7, 11)

find mid = $\lceil (i+j)/2 \rceil$ = $(0+7)/2$ = 3

mid = 3

check ~~value~~ = A[mid] (11 == 9 ~~[illegible]~~)

value > A[3]. So, we have to search the element in ~~[illegible]~~ the right part of the middle element.

Binary search (A, mid+1, j, 11)

Binary search (A, 4, 7, 11)

[4]	[5]	[6]	[7]
10	12	14	19

i = 4, j = 7

find mid = $\lceil (i+j)/2 \rceil$ = $4+7/2$ = 5

check value == A[5] — No

Binary search (A, 4, 4, 11)   value < ~~[illegible]~~ A[5]
So, we have to search the key element in the left of the middle position.

4
10

find mid
= $(4+4)/2$ = 4

check value == A[4] — No.

value > A[4].

Hence we will search the element in the right portion of the array but the element has no left and right portion to search. So it will returns the search failure that means $\boxed{\text{Data not found}}$.

Q7 I/P: A

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
10	14	19	26	27	31	33	35	42	44.

Key element = 22

step 1: first of all find the mid Position

$i=0, j=9$

$Mid = (i+j/2) = \frac{0+9}{2} = 4$

Mid Position of the above list of data = 4

Now we will compare middle element (A[4]) which is 27 with 22

Since $22 < 27$

Hence we will search the element in the left portion of the array

[0]	[1]	[2]	[3]
10	14	19	26

Step 2:

[0]	[1]	[2]	[3]
10	14	19	26

Again find the Mid Position of the above array

$i=0, j=3$

So, $mid = (0+3)/_2 = 1$

Now we will compare middle element A[1] which is 14 with 22.

Since $22 > 14$

Hence we will search the element in the right position of the array.

[0]	[1]
19	26

Step 3:

[2]	[3]
19	26

Again find the mid position of the above array.

$i=2,\ j=3$

$\text{Mid} = (2+3)/2 = 2$

Now we will compare middle element A[2] which is 19 with 22

Since $22 > 19$

Hence we will search the element in the right portion of the array.

[3].
26

Step 4:

3
26

, in this case $i=3$ and also $j=3$

Again find mid position

So, $\text{Mid} = (3+3)/2 = 3$.

Now we will compare middle element A[3] which. is 26 with 22.

Since $22 < 26$

Hence we will search the element in the left portion of the array but the element has no left and right portion to search. So it will return the search failure. that means

$\boxed{\text{Data Not found.}}$