Day 43

Binary Search

Given a sorted array of n elements. Write a function to search a given element

Search a sorted array by repeatly dividing the search interval in half. Begin with an interval covering the whole array. If the value of search key is less than the item in the middle of the interval, narrow the interval to the lower half. otherwise narrow it to the upper half. Repeatly check until the value is found or the interval is empty.

Algorithm:

* Compare x with middle element
* if x matches with middle element, we return the mid index
* else if x is greater than the mid element, then x can only lie in right half subarray after the mid element. So we recur the mid element
* else recur for the left half.

Implementation:-

```
def binarysearch(arr, x):
    r = len(arr) - 1
    l = 0
    if r >= l:
        mid = l + (r - l) // 2
        if arr[mid] == x:
            return mid
```

```
elif arr[mid] > x:
    return binarysearch(arr, x)

else:
    return binarysearch(arr, x)

else:
    return -1
```

Input

arr = [2, 3, 4, 10, 40]

x = 10

binary search(arr, x)

Output :

3

time Complexity :

O(log n)

Linear Search

* Start from the leftmost element of arr and one by one Compare x with each element of array.

* if x match with an element, return the inden

* else, return, -1

```
def binary

def linear(arr, x):
    for i in range(0, len(arr)):
        if arr[i] == x:
            return i
    return -1
```

input
arr = [2, 3, 4, 10, 40]
x = 10
~~Search(~~
linear(arr, x)

output:
3

time Complexity:
O(n)