

Linear Search and Binary search

' Aim:

To write a program to perform linear search and binary search using python programming.

' Equipment's required:

1. Hardware – PCs
2. Anaconda – Python 3.7 Installation / Moodle-Code Runner

' Algorithm:

' Linear Search:

1. Start from the leftmost element of array[] and compare k with each element of array[] one by one.
2. If k matches with an element in array[] , return the index.
3. If k doesn't match with any of elements in array[], return -1 or element not found.

' Binary Search:

1. Set two pointers low and high at the lowest and the highest positions respectively.
2. Find the middle element mid of the array ie. $\text{arr}[(\text{low} + \text{high})/2]$
3. If $x == \text{mid}$, then return mid. Else, compare the element to be searched with m.
4. If $x > \text{mid}$, compare x with the middle element of the elements on the right side of mid. This is done by setting low to $\text{low} = \text{mid} + 1$.
5. Else, compare x with the middle element of the elements on the left side of mid. This is done by setting high to $\text{high} = \text{mid} - 1$.
6. Repeat steps 2 to 5 until low meets high

' Program:

i) #Use a linear search method to match the item in a list.

```
def linearSearch(array,n,k):
    for i in range(0,n):
        if (array[i]==k):
            return i
    return -1
array = eval(input())
k=eval(input())
n=len(array)
array.sort()
result = linearSearch(array,n,k)
if(result!=-1):
```

```

    print(array)
    print("Element not found")
else:
    print(array)
    print("Element found at index: ", result)

```

ii) # Find the element in a list using Binary Search(Iterative Method).

```

def binarySearchIter(array, k, low, high):
    while low <= high :
        mid = low + (high-low)//2
        if array [mid] == k:
            return mid
        elif array [mid] < k:
            low = mid + 1
        else:
            high = mid- 1
    return -1
array = eval(input())
array.sort()
k = eval(input())
result = binarySearchIter(array, k, 0, len(array)-1)
if(result == -1):
    print(array)
    print("Element not found")
else:
    print(array)
    print("Element found at index: ", result)

```

iii) # Find the element in a list using Binary Search (recursive Method).

```

def BinarySearch(arr, k, low, high):
    if high >= low:
        mid = low + (high - low)//2
        if arr[mid] == k:
            return mid
        elif arr[mid] > k:
            return BinarySearch(arr, k, low, mid-1)
        else:
            return BinarySearch(arr, k, mid +1, high)
    else:
        return -1
arr = eval(input())
arr.sort()

```

' Sample Input and Output:

Linear search:

Input	Result
[1, 8, 7, 9, 10] 7	[1, 7, 8, 9, 10] Element found at index: 1
[9, 10, 8, 7, 6] 20	[6, 7, 8, 9, 10] Element not found

	Input	Expected	Got	
✓	[1, 8, 7, 9, 10] 7	[1, 7, 8, 9, 10] Element found at index: 1	[1, 7, 8, 9, 10] Element found at index: 1	✓
✓	[10, 78, 68, 67, 56] 68	[10, 56, 67, 68, 78] Element found at index: 3	[10, 56, 67, 68, 78] Element found at index: 3	✓
✓	[78, 56, 77, 98, 95] 77	[56, 77, 78, 95, 98] Element found at index: 1	[56, 77, 78, 95, 98] Element found at index: 1	✓
✓	[9, 10, 8, 7, 6] 20	[6, 7, 8, 9, 10] Element not found	[6, 7, 8, 9, 10] Element not found	✓

Passed all tests! ✓

Binary search(Iterative):

Input	Result
[9, 6, 7, 8, 1, 3] 9	[1, 3, 6, 7, 8, 9] Element found at index: 5
[9, 10, 8, 7, 6] 20	[6, 7, 8, 9, 10] Element not found

	Input	Expected	Got	
✓	[9, 6, 7, 8, 1, 3] 9	[1, 3, 6, 7, 8, 9] Element found at index: 5	[1, 3, 6, 7, 8, 9] Element found at index: 5	✓
✓	[10, 78, 68, 67, 56] 68	[10, 56, 67, 68, 78] Element found at index: 3	[10, 56, 67, 68, 78] Element found at index: 3	✓
✓	[9, 10, 8, 7, 6] 20	[6, 7, 8, 9, 10] Element not found	[6, 7, 8, 9, 10] Element not found	✓
✓	[78, 56, 77, 98, 95] 77	[56, 77, 78, 95, 98] Element found at index: 1	[56, 77, 78, 95, 98] Element found at index: 1	✓

Passed all tests! ✓

Binary search(recursive):

Input	Result
[9, 6, 7, 8, 1, 3] 9	[1, 3, 6, 7, 8, 9] Element found at index: 5
[9, 10, 8, 7, 6] 20	[6, 7, 8, 9, 10] Element not found

	Input	Expected	Got	
✓	[9, 6, 7, 8, 1, 3] 9	[1, 3, 6, 7, 8, 9] Element found at index: 5	[1, 3, 6, 7, 8, 9] Element found at index: 5	✓
✓	[10, 78, 68, 67, 56] 68	[10, 56, 67, 68, 78] Element found at index: 3	[10, 56, 67, 68, 78] Element found at index: 3	✓
✓	[9, 10, 8, 7, 6] 20	[6, 7, 8, 9, 10] Element not found	[6, 7, 8, 9, 10] Element not found	✓
✓	[78, 56, 77, 98, 95] 77	[56, 77, 78, 95, 98] Element found at index: 1	[56, 77, 78, 95, 98] Element found at index: 1	✓

Passed all tests! ✓

Result

Thus the linear search and binary search algorithm is implemented using python programming.