DAY-21

Searching Algorithms.

→ designed to check for an element or retrieve an element from any data structure where it is stored. There are two types.

1. Sequential Search.

→ the list or array is traversed sequentially and every element is checked.

example:- * Linear Search.

2. Interval Search :-

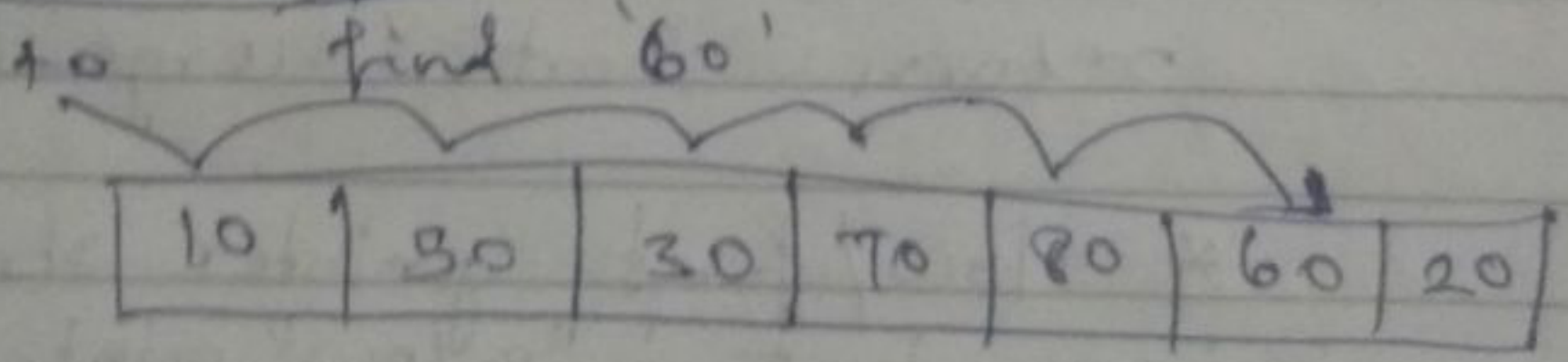
→ specially designed for searching in sorted data-structured.

→ More efficient than Linear search; and it devided the search space in half.

* Example :- Binary Search.

Geek for Geeks ref:

Linear search :-



Time complenity

Best Case :- O(1)

Worst Case :- O(n).

Implementation :-

```
def linearSearch (arr, n):
    for i in range (0, len(n)):
        if arr[i] = n:
            return "found at", i
    return "not found"
```

Binary Search :-

-> Search a sorted array by repeatedly dividing the search interval in half.

-> Begin with an interval covering the whole array.

-> if the value of the 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.

-> Repeatedly check until the value is found or the interval is empty.

* The idea of binary search is to use the information that the array is sorted and reduce the time complexity to O(logn).

Implementation :-

```
def binarySearch(arr, l, r, x):
    if r >= l:
        mid = l + (r - l) // 2
        if arr[mid] == x:
            return mid
        elif arr[mid] > x:
            return binarySearch(arr, l, mid-1, x)
        else:
            return binarySearch(arr, mid+1, r, x)
    else:
        return "Not found"
#call
result = binarySearch(arr, 0, len(arr)-1, x)
```

* Can be implemented also "Without recursion"