Searching

Searching

Searching refers to the operation of finding an item from a list of items based on some key value.

Two Searching Methods

(1) Linear Search

(2) Binary Search

Linear Search

- A linear search is a technique for finding a particular value in a unsorted or sorted list.
- Search each item of list using key value, one at a time, until finding the item from the list.

Algorithm: Linear-Search (L, N, Key)

1. Set k := 1 and Loc := 0
2. Repeat Steps 3 and 4 while k <= N
3. If Key = L[k], then Set Loc := k, print: Loc and exit.
4. Set k := k+1
5. If Loc = 0 then Write: Item is not in List
6. Exit

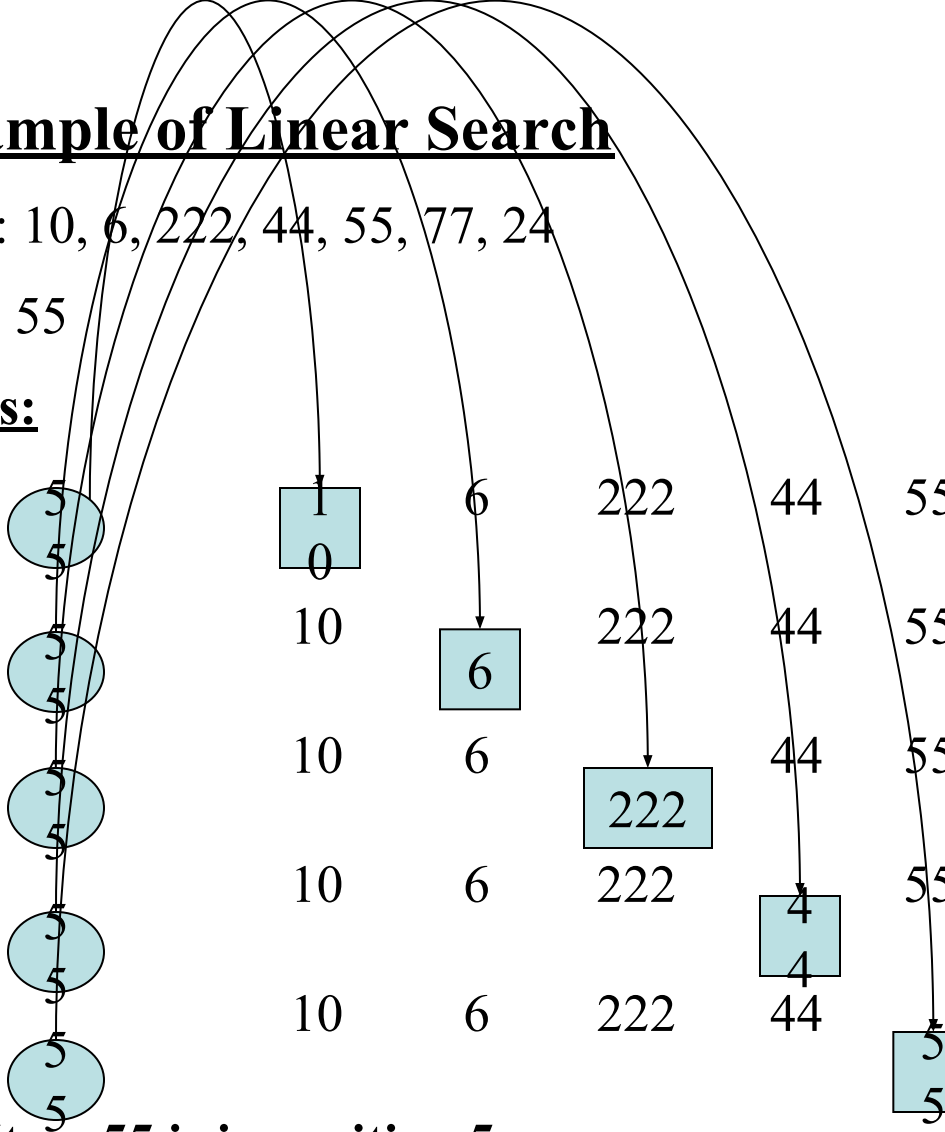
Here
L - The list of data items
N – Total no. of items in L
Key – Item to be searched.

Example of Linear Search

List : 10, 6, 222, 44, 55, 77, 24

Key: 55

Steps:



1. 10 6 222 44 55 77 24 → K=1
2. 10 6 222 44 55 77 24 → K=2
3. 10 6 222 44 55 77 24 → K=3
4. 10 6 222 44 55 77 24 → K=4
5. 10 6 222 44 55 77 24 → K=5

So, Item 55 is in position 5.

Complexity Analysis of Linear Search

(1) Worst Case

The worst case occurs when Item is the last element in the List or is not in the List.

So, C(n) = n = Θ(n).

(2) Average Case

The average case occurs when the availability of the Item in the List is equally likely to occur at any position in the List.

So, C(n) = $1.\frac{1}{n} + 2.\frac{1}{n} + \ldots\ldots\ldots + n.\frac{1}{n}$

= $(1 + 2 + \ldots\ldots + n).\frac{1}{n}$

= $\frac{n(n+1)}{2}.\frac{1}{n}$

= $\frac{n+1}{2}$

= $O(n)$

Binary Search

- List of items should be sorted.
- It uses the divide-and-conquer technique.
- The elements of the list are not necessarily all unique. If one searches for a value that occurs multiple times in the list, the index returned by the binary search will be of the first-encountered equal element.
- Compare the given Key value with the element in the middle of the list. It tells which half of the list contains the Item.Then compare Key with the element in the middle of the correct half to determine which quarter of the list contains Item. Continue the process until finding Item in the list.

Algorithm: Binary-Search (L, N, Key)

1. Set Loc := 0, Beg := 1, End :=N and Mid := (Beg + End)/2
2. Repeat steps 3 and 6 while Beg <= End
3. If Key < L[Mid], then Set End := Mid – 1
4. Else if Key>L[Mid] then Set Beg := Mid + 1
5. Else if Key = List[Mid] then Loc:= Key, print: Loc and exit.
6. Set Mid := (Beg + End)/2
7. If Loc = 0 Write: Item is not in List.
8. Exit.

Here
L - The list of data items
N – Total no. of items in L
Key – Item to be searched.

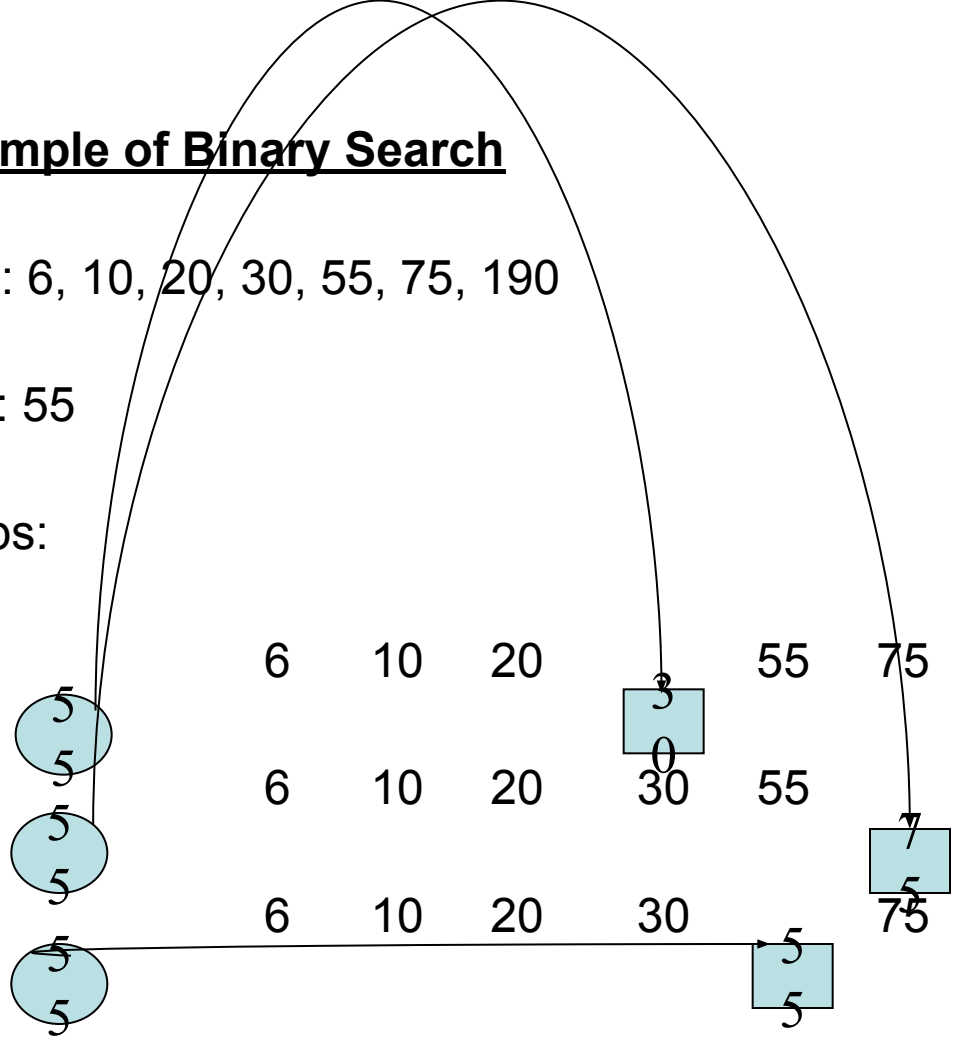
*

Example of Binary Search

List : 6, 10, 20, 30, 55, 75, 190

Key: 55

Steps:



1. 6 10 20 30 55 75 190 Beg=1, End=7,Mid=4
2. 6 10 20 30 55 190 Beg=5, End=7,Mid=6
3. 6 10 20 30 75 190 Beg=5, End=5,Mid=5

[Item Found]

So, Item 55 is in position 5.

*

Complexity of Binary Search

The complexity of binary search is measured by the number of comparisons to locate Item in List where List contains n elements. It is observed that each comparison reduces the sample size in half.

So, $C(n) = n + n/2 + n/4 + \ldots\ldots\ldots\ldots +1 = \log_2 n + 1 = 0(\log_2 n)$

END