Searching Algorithm(s)

Linear Search

Binary Search

Shoaib Rauf

Linear Search:

- Linear search algorithm finds a given element in a list of elements with **O(n)** time complexity where **n** is total number of elements in the list.
- This search process starts comparing search element with the first element in the list.
- If both are matched then result is element found otherwise search element is compared with the next element in the list.
- Repeat the same until search element is compared with the last element in the list, if that last element also doesn't match, then the result is "**Element not found in the list**".

Linear Search Algorithm:

- Linear search is implemented using following steps.

Step 1 - Read the search element from the user.

Step 2 - Compare the search element with the first element in the list.

Step 3 - If both are matched, then display "Given element is found!!!" and terminate the function

Step 4 - If both are not matched, then compare search element with the next element in the list.

Step 5 - Repeat steps 3 and 4 until search element is compared with last element in the list.

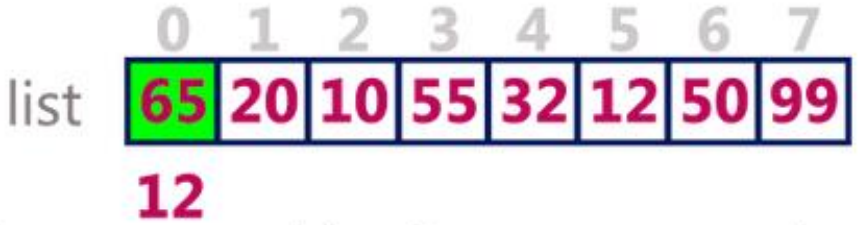
Step 6 - If last element in the list also doesn't match, then display "Element is not found!!!" and terminate the function.

Linear Search Example

search element **12**

Step 1:

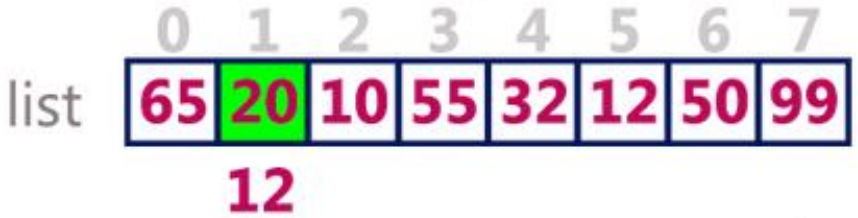
search element (12) is compared with first element (65)



Both are not matching. So move to next element

Step 2:

search element (12) is compared with next element (20)



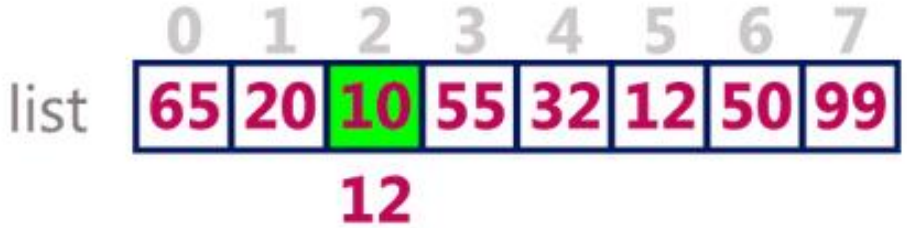
Both are not matching. So move to next element

Linear Search Example

search element **12**

Step 3:

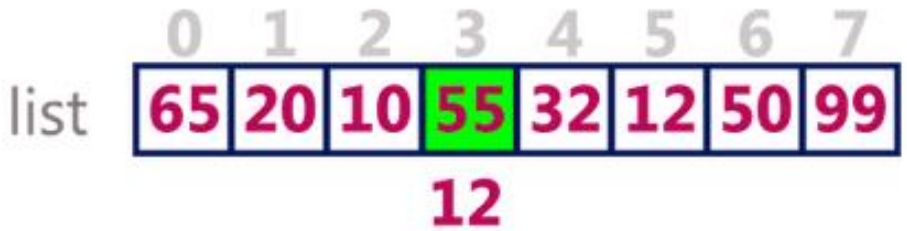
search element (12) is compared with next element (10)



Both are not matching. So move to next element

Step 4:

search element (12) is compared with next element (55)



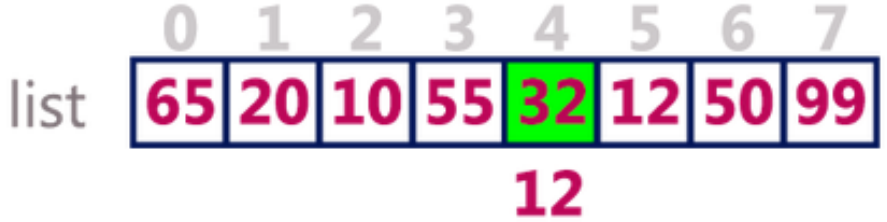
Both are not matching. So move to next element

Linear Search Example

search element **12**

Step 5:

search element (12) is compared with next element (32)



Both are not matching. So move to next element

Step 6:

search element (12) is compared with next element (12)

Both are matching. So we stop comparing and display element found at index 5.

Binary Search

- Binary search algorithm finds a given element in a list of elements with **O(log n)** time complexity where **n** is total number of elements in the list.
- The binary search algorithm can be used with only a sorted list of elements.
- That means the binary search is used only with a list of elements that are already arranged in an order.
- The binary search can not be used for a list of elements arranged in random order.
- This search process starts comparing the search element with the middle element in the list.
- If both are matched, then the result is "element found".

Binary Search

- Otherwise, we check whether the search element is smaller or larger than the middle element in the list.
- If the search element is smaller, then we repeat the same process for the left sublist of the middle element.
- If the search element is larger, then we repeat the same process for the right sublist of the middle element.
- We repeat this process until we find the search element in the list or until we left with a sublist of only one element.
- If that element also doesn't match with the search element, then the result is "Element not found in the list".

Binary Search Algorithm:

- Binary search is implemented using following steps.

Step 1 - Read the search element from the user.

Step 2 - Find the middle element in the sorted list.

Step 3 - Compare the search element with the middle element in the sorted list.

Step 4 - If both are matched, then display "Given element is found!!!" and terminate the function.

Step 5 - If both are not matched, then check whether the search element is smaller or larger than the middle element.

Step 6 - If the search element is smaller than middle element, repeat steps 2, 3, 4 and 5 for the left sublist of the middle element.

Step 7 - If the search element is larger than middle element, repeat steps 2, 3, 4 and 5 for the right sublist of the middle element.

Step 8 - Repeat the same process until we find the search element in the list or until sublist contains only one element.

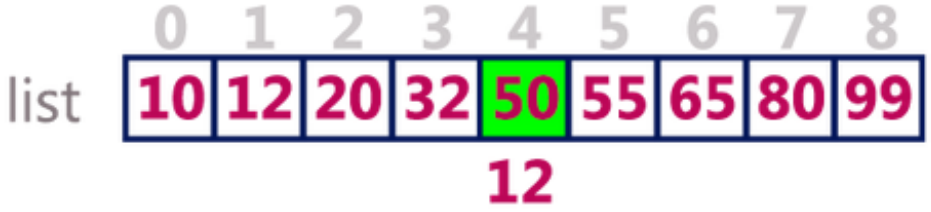
Step 9 - If that element also doesn't match with the search element, then display "Element is not found in the list!!!" and terminate the function.

Binary Search Example:

search element **12**

Step 1:

search element (12) is compared with middle element (50)



Both are not matching. And 12 is smaller than 50. So we search only in the left sublist (i.e. 10, 12, 20 & 32).

Step 2:

search element (12) is compared with middle element (12)

Both are matching. So the result is "Element found at index 1"