LINEAR SEARCH

In Linear Search the list is searched sequentially and the position is returned if the key element to be searched is available in the list, otherwise -1 is returned. The search in Linear Search starts at the beginning of an array and move to the end, testing for a match at each item.

All the elements preceding the search element are traversed before the search element is traversed. i.e. if the element to be searched is in position 10, all elements form 1-9 are checked before 10.

Algorithm : Linear search implementation

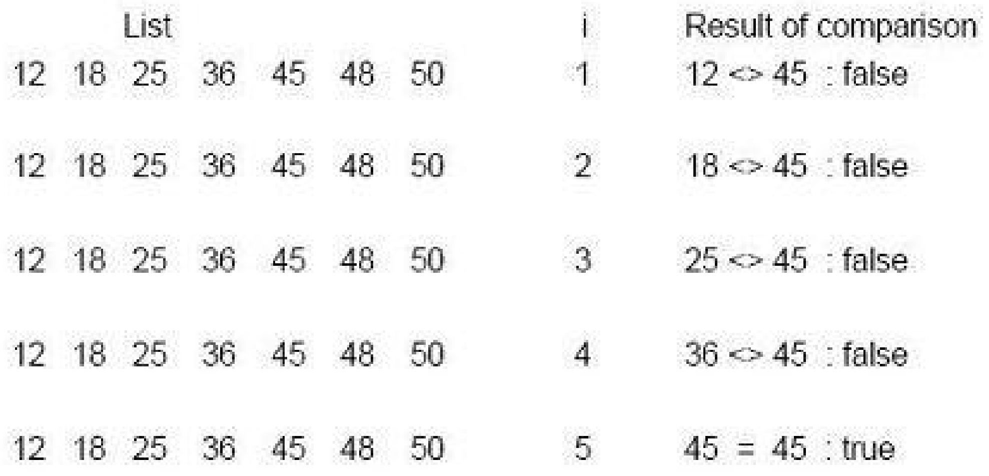
```
bool linear_search ( int *list, int size, int key, int* rec )
{
    // Basic Linear search
    bool found = false;
 int i;
    for ( i = 0; i < size; i++ )
          {
              if ( key == list[i] )
                    break;
           }
            if ( i < size )
               {
                    found = true;
                    rec = &list[i];
               }
             return found;
```

}

The code searches for the element through a loop starting form 0 to n. The loop can terminate in one of two ways. If the index variable i reach the end of the list, the loop condition fails. If the current item in the list matches the key, the loop is terminated early with a break statement. Then the algorithm tests the index variable to see if it is less than that size (thus the loop was terminated early and the item was found), or not (and the item was not found).

Ex.

Assume the element 45 is searched from a sequence of sorted elements 12, 18, 25, 36, 45, 48, 50. The Linear search starts from the first element 12, since the value to be searched is not 12 (value 45), the next element 18 is compared and is also not 45, by this way all the elements before 45 are compared and when the index is 5, the element 45 is compared with the search value and is equal, hence the element is found and the element position is 5.



List	i	Result of comparison
12 18 25 36 45 48 50	1	12 <> 45 : false
12 18 25 36 45 48 50	2	18 <> 45 : false
12 18 25 36 45 48 50	3	25 <> 45 : false
12 18 25 36 45 48 50	4	36 <> 45 : false
12 18 25 36 45 48 50	5	45 = 45 : true

LINEAR SEARCH COMPLEXITY

Time Complexity

Case	Time Complexity
Best Case	O(1)
Average Case	O(n)
Worst Case	O(n)

Space Complexity

Space Complexity	O(1)

Application of Linear Search Algorithm

- The linear search is applicable on both single and multi-dimensional arrays.
- It is less complex, effective, and easy to implement when the array contains a few elements.
- It is efficient when we need to search for a single element in an unordered array.