Tutorial Link https://codequotient.com/tutorials/Searching an Element - Linear Search/5a12edf146765b2b63e3476b

TUTORIAL

Searching an Element - Linear Search

Chapter

The process of identifying or finding a particular record is called Searching. You often spend time in searching for any desired item. If the data is kept properly in sorted order, then searching becomes very easy and efficient. Any search is said to be successful or unsuccessful depending upon whether the element that is being searched is found or not. In this article you will get to know the basic concepts of searching in sorted and unsorted arrays that is used in data structures. Search can be done popularly in two ways: -

Linear Search

If we start from the first element of list, and compare each element with the element we are searching, it is called linear search. This method can be performed on a sorted or an unsorted list (usually arrays). In both cases, the search will start from array index 0 and each time it match the searched element and the element at current index. If they match it will return the index otherwise, it will move to next index. If the whole array is passed to a function and the searched element does not match then it will return a negative index. The time complexity of Linear search is O(n) as it may have to search for all the elements in worst case. In case of an array the general algorithm for linear search is as follows: -

```
X = searched_Element
index = 0
While(index < length_of_array)
     If: X == array[index] then RETURN index
     Else: index = index + 1
     End
End
RETURN -1
```

Following is the iterative implementation of linear search: -

Javascript

```
function linear_search(arr,n,x){
    let i;
    for (i=0; i<n; i++)
        if (arr[i] == x)    // Check each element of the array.
            return i;        // if found return the position
    return -1;    // otherwise return -1
}

function main(){
    let arr = [10,11,12,13,14,25,26,37,48,29]
    const x=25;    // Searched Element.

    let loc=linear_search(arr, arr.length, x);    // Call the search function

    if(loc != -1)
        console.log(`Element found at location : ${loc}`);
    else
        console.log(`Element not present in the array.`);
    return 0;
}

main()
```

C

```
#include<stdio.h>

int linear_search(int arr[], int n, int x)
```

```
{
  int i;
  for (i=0; i<n; i++)
    if (arr[i] == x)    // Check each element of the array.
      return i;        // if found return the position
  return -1;    // otherwise return -1
}

int main()
{
  int loc,x,array[]={10,11,12,13,14,25,26,37,48,29};

  x=25;    // Searched Element.

  loc=linear_search(array, 10, x);    // Call the search function

  if(loc != -1)
    printf("Element found at location : %d",loc);
  else
    printf("Element not present in the array.");
  return 0;
}

```

Java

```
import java.util.Scanner;
// Other imports go here
// Do NOT change the class name
class Main{
  static int linear_search(int arr[], int n, int x)
  {
    int i;
    for (i=0; i<n; i++)
      if (arr[i] == x)    // Check each element of the array.
        return i;        // if found return the position
    return -1;    // otherwise return -1
  }
  public static void main(String[] args)
  {
    int loc,x,array[]={10,11,12,13,14,25,26,37,48,29};
```

```
    x=25;    // Searched Element.

    loc=linear_search(array, 10, x);    // Call the search function

    if(loc != -1)
      System.out.print("Element found at location : " + loc);
    else
      System.out.print("Element not present in the array.");
  }
}
```

Python 3

```
def linear_search(array,n,x):
    for i in range(n):
        if array[i] == x:  # Check every element of the array
            return i       # If found , return the position
    return -1              # Otherwise return -1


if __name__ == '__main__':
    array = [10,11,12,13,14,25,26,37,48,29]
    x = 25 # Element to be searched
    loc = linear_search(array,len(array),x) # Calling the function
    if(loc !=-1):
        print('Element found at location :',loc)
    else:
        print('Element not present in array.')
```

C++

```
#include<iostream>
#include<cstdio>
#include<cmath>
using namespace std;

int linear_search(int arr[], int n, int x) {
    int i;
    for (i=0; i<n; i++)
        if (arr[i] == x)    // Check each element of the array.
```

```
            return i;        // if found return the position
    return -1;    // otherwise return -1
}

int main() {
    int array[]={10,11,12,13,14,25,26,37,48,29};
    int x=25;    // Searched Element.

    int loc=linear_search(array, 10, x);    // Call the search function

    if(loc != -1)
        cout<<"Element found at location : "<<loc;
    else
        cout<<"Element not present in the array.";
    return 0;
}
```

The output of above program is as below for different runs: -

```
Element found at location :5
```

Recursive implementation of linear

```
function rec_linear_search(arr,left,right,x) {
    let result;
    if (right < left)        // The array is exhausted so return -1
        return -1;
    if (arr[left] == x)        // If element found return position
        return left;
    // Call the function again with new subarray from next element.
    result = rec_linear_search(arr, left+1, right, x);

    return result;    // return the result to the calling function.
}

function main(){
    let arr = [10,11,12,13,14,25,26,37,48,29]
```

Javascript

```
    const x=13;    // Searched Element.

    let loc=rec_linear_search(arr,0, arr.length, x);
// Call the search function

    if(loc != -1)
        console.log(`Element found at location :
${loc}`);
    else
        console.log(`Element not present in the
array.`);
    return 0;
}

main()
```

C

```
#include<stdio.h>

int rec_linear_search(int arr[], int left, int right,
int x)
{
  int result;
  if (right < left)        // The array is exhausted so
return -1
    return -1;
  if (arr[left] == x)        // If element found return
position
    return left;
  // Call the function again with new subarray from
next element.
  result = rec_linear_search(arr, left+1, right, x);

  return result;    // return the result to the calling
function.
}

int main()
{
  int loc,x,array[]={10,11,12,13,14,25,26,37,48,29};
  x=13;            // element to be searched in the
array
  loc=rec_linear_search(array,0,10,x);
  if(loc != -1)
    printf("Element found at location : %d",loc);
  else
    printf("Element not present in the array.");
```

```

  return 0;
}

```

Java

```
import java.util.Scanner;
// Other imports go here
// Do NOT change the class name
class Main{
static int rec_linear_search(int arr[], int left, int right, int x)
{
  int result;
  if (right < left)        // The array is exhausted so return -1
    return -1;
  if (arr[left] == x)        // If element found return position
    return left;
  // Call the function again with new subarray from next element.
  result = rec_linear_search(arr, left+1, right, x);

  return result;    // return the result to the calling function.
}
  public static void main(String[] args)
  {
    int loc,x,array[]={10,11,12,13,14,25,26,37,48,29};
    x=13;    // Searched Element.
    loc = rec_linear_search(array, 0, 10, x);    // Call the search function
    if(loc != -1)
      System.out.print("Element found at location : " + loc);
    else
      System.out.print("Element not present in the array.");
  }
}
```

Python 3

```
def rec_linear_search(array,left,right,x):
    if (right < left):        # The array is exhausted so return -1
        return -1;
    if (array[left] == x):        # If element found
```

```
return position
        return left
    # Call the function again with new subarray from next element.
    result = rec_linear_search(array, left+1, right, x)

    return result    # return the result to the calling function.



if __name__ == '__main__':
    array = [10,11,12,13,14,25,26,37,48,29]
    x = 13 # Element to be searched
    loc = rec_linear_search(array,0,len(array),x) # Calling the function
    if(loc !=-1):
        print('Element found at location :',loc)
    else:
        print('Element not present in array.')
```

C++

```
#include<iostream>
#include<cstdio>
#include<cmath>
using namespace std;


int rec_linear_search(int arr[], int left, int right, int x) {
    int result;
    if (right < left)        // The array is exhausted so return -1
        return -1;
    if (arr[left] == x)        // If element found return position
        return left;
    // Call the function again with new subarray from next element.
    result = rec_linear_search(arr, left+1, right, x);

    return result;    // return the result to the calling function.
}


int main() {
    int array[]={10,11,12,13,14,25,26,37,48,29};
    int x=13;    // Searched Element.
    int n = sizeof(array)/sizeof(array[0]);
```

```
    int loc=rec_linear_search(array,0,n, x);    // Call the search function

    if(loc != -1)
        cout<<"Element found at location : "<<loc;
    else
        cout<<"Element not present in the array.";
    return 0;
}
```

The output of above program is as below for different runs: -

```
Element found at location :3
```