Search an element in a Linked List (Iterative and Recursive)

Write a function that searches a given key 'x' in a given singly linked list. The function should return true if x is present in linked list and false otherwise.

```
bool search(Node *head, int x)
```

For example, if the key to be searched is 15 and linked list is 14->21->11->30->10, then function should return false. If key to be searched is 14, then the function should return true.

Iterative Solution

```
2) Initialize a node pointer, current = head.
3) Do following while current is not NULL
    a) current->key is equal to the key being searched return true.
    b) current = current->next
4) Return false
```

Following is iterative implementation of above algorithm to search a given key.

C++

```
// Iterative C++ program to search
// an element in linked list
#include <bits/stdc++.h>
using namespace std;

/* Link list node */
class Node
{
    public:
    int key;
    Node* next;
};

/* Given a reference (pointer to pointer) to the head
of a list and an int, push a new node on the front
of the list. */
void push(Node** head_ref, int new_key)
{
    /* allocate node */
    Node* new_node = new Node();

    /* put in the key */
    new_node->key = new_key;

    /* link the old list off the new node */
    new_node->next = (*head_ref);

    /* move the head to point to the new node */
    (*head_ref) = new_node;
}

/* Checks whether the value x is present in linked list */
bool search(Node* head, int x)
{
    Node* current = head; // Initialize current
    while (current != NULL)
    {
        if (current->key == x)
            return true;
        current = current->next;
    }
    return false;
}

/* Driver program to test count function*/
int main()
{
    /* Start with the empty list */
    Node* head = NULL;
    int x = 21;

    /* Use push() to construct below list
    14->21->11->30->10 */
    push(&head, 10);
    push(&head, 30);
    push(&head, 11);
    push(&head, 21);
    push(&head, 14);

    search(head, 21)? cout<<"Yes" : cout<<"No";
    return 0;
}

// This is code is contributed by rathbhupendra
```

C

```
// Iterative C program to search an element in linked list
#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>

/* Link list node */
struct Node
{
    int key;
    struct Node* next;
};

/* Given a reference (pointer to pointer) to the head
  of a list and an int, push a new node on the front
  of the list. */
void push(struct Node** head_ref, int new_key)
{
    /* allocate node */
    struct Node* new_node =
            (struct Node*) malloc(sizeof(struct Node));

    /* put in the key  */
    new_node->key  = new_key;

    /* link the old list off the new node */
    new_node->next = (*head_ref);

    /* move the head to point to the new node */
    (*head_ref)    = new_node;
}

/* Checks whether the value x is present in linked list */
bool search(struct Node* head, int x)
{
    struct Node* current = head;  // Initialize current
    while (current != NULL)
    {
        if (current->key == x)
            return true;
        current = current->next;
    }
    return false;
}

/* Driver program to test count function*/
int main()
```

```
{
    /* Start with the empty list */
    struct Node* head = NULL;
    int x = 21;

    /* Use push() to construct below list
     14->21->11->30->10  */
    push(&head, 10);
    push(&head, 30);
    push(&head, 11);
    push(&head, 21);
    push(&head, 14);

    search(head, 21)? printf("Yes") : printf("No");
    return 0;
}
```

Java

```
// Iterative Java program to search an element
// in linked list

//Node class
class Node
{
    int data;
    Node next;
    Node(int d)
    {
        data = d;
        next = null;
    }
}

//Linked list class
class LinkedList
{
    Node head;    //Head of list

    //Inserts a new node at the front of the list
    public void push(int new_data)
    {
        //Allocate new node and putting data
        Node new_node = new Node(new_data);

        //Make next of new node as head
        new_node.next = head;

        //Move the head to point to new Node
        head = new_node;
    }

    //Checks whether the value x is present in linked list
    public boolean search(Node head, int x)
    {
        Node current = head;    //Initialize current
        while (current != null)
        {
            if (current.data == x)
                return true;    //data found
            current = current.next;
        }
        return false;    //data not found
    }

    //Driver function to test the above functions
    public static void main(String args[])
    {

        //Start with the empty list
        LinkedList llist = new LinkedList();

        /*Use push() to construct below list
        14->21->11->30->10  */
        llist.push(10);
        llist.push(30);
        llist.push(11);
        llist.push(21);
        llist.push(14);
        if (llist.search(llist.head, 21))
            System.out.println("Yes");
        else
            System.out.println("No");
    }
}
// This code is contributed by Pratik Agarwal
```

Python

```
# Iterative Python program to search an element
# in linked list

# Node class
class Node:

    # Function to initialise the node object
    def __init__(self, data):
        self.data = data # Assign data
        self.next = None # Initialize next as null

# Linked List class
class LinkedList:
    def __init__(self):
        self.head = None # Initialize head as None

    # This function insert a new node at the
    # beginning of the linked list
    def push(self, new_data):

        # Create a new Node
        new_node = Node(new_data)

        # 3. Make next of new Node as head
        new_node.next = self.head

        # 4. Move the head to point to new Node
        self.head = new_node

    # This Function checks whether the value
    # x present in the linked list
    def search(self, x):

        # Initialize current to head
        current = self.head

        # loop till current not equal to None
        while current != None:
            if current.data == x:
                return True # data found

            current = current.next

        return False # Data Not found


# Code execution starts here
if __name__ == '__main__':

    # Start with the empty list
    llist = LinkedList()

    ''' Use push() to construct below list
        14->21->11->30->10 '''
    llist.push(10);
    llist.push(30);
    llist.push(11);
    llist.push(21);
    llist.push(14);

    if llist.search(21):
        print("Yes")
    else:
        print("No")

# This code is contributed by Ravi Shankar
```

C#

```
// Iterative C# program to search an element
// in linked list
using System;

// Node class
public class Node
{
    public int data;
    public Node next;
    public Node(int d)
    {
        data = d;
        next = null;
    }
}

// Linked list class
public class LinkedList
{
    Node head; // Head of list

    // Inserts a new node at the front of the list
    public void push(int new_data)
    {
        // Allocate new node and putting data
        Node new_node = new Node(new_data);

        // Make next of new node as head
        new_node.next = head;

        // Move the head to point to new Node
        head = new_node;
    }

    // Checks whether the value x is present in linked list
    public bool search(Node head, int x)
    {
        Node current = head; // Initialize current
        while (current != null)
        {
            if (current.data == x)
                return true; // data found
            current = current.next;
        }
        return false; // data not found
    }

    // Driver code
    public static void Main(String []args)
    {

        // Start with the empty list
        LinkedList llist = new LinkedList();

        /*Use push() to construct below list
        14->21->11->30->10 */
        llist.push(10);
        llist.push(30);
        llist.push(11);
        llist.push(21);
        llist.push(14);

        if (llist.search(llist.head, 21))
            Console.WriteLine("Yes");
        else
            Console.WriteLine("No");
    }
}

// This code contributed by Rajput-Ji
```

Output:

```
Yes
```

Recursive Solution

```
bool search(head, x)
1) If head is NULL, return false.
2) If head's key is same as x, return true;
2) Else return search(head->next, x)
```

Following is recursive implementation of above algorithm to search a given key.

C

```c
// Recursive C program to search an element in linked list
#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>
/* Link list node */
struct Node
{
    int key;
    struct Node* next;
};

/* Given a reference (pointer to pointer) to the head
  of a list and an int, push a new node on the front
  of the list. */
void push(struct Node** head_ref, int new_key)
{
    /* allocate node */
    struct Node* new_node =
            (struct Node*) malloc(sizeof(struct Node));

    /* put in the key  */
    new_node->key  = new_key;

    /* link the old list off the new node */
    new_node->next = (*head_ref);

    /* move the head to point to the new node */
    (*head_ref)    = new_node;
}

/* Checks whether the value x is present in linked list */
bool search(struct Node* head, int x)
{
    // Base case
    if (head == NULL)
        return false;

    // If key is present in current node, return true
    if (head->key == x)
        return true;

    // Recur for remaining list
    return search(head->next, x);
}

/* Driver program to test count function*/
int main()
{
    /* Start with the empty list */
    struct Node* head = NULL;
    int x = 21;

    /* Use push() to construct below list
     14->21->11->30->10  */
    push(&head, 10);
    push(&head, 30);
    push(&head, 11);
    push(&head, 21);
    push(&head, 14);

    search(head, 21)? printf("Yes") : printf("No");
    return 0;
}
```

Java

```java
// Recursive Java program to search an element
// in linked list


// Node class
class Node
{
    int data;
    Node next;
    Node(int d)
    {
        data = d;
        next = null;
    }
}

// Linked list class
class LinkedList
{
    Node head;    //Head of list

    //Inserts a new node at the front of the list
    public void push(int new_data)
    {
        //Allocate new node and putting data
        Node new_node = new Node(new_data);

        //Make next of new node as head
        new_node.next = head;

        //Move the head to point to new Node
        head = new_node;
    }

    // Checks whether the value x is present
    // in linked list
    public boolean search(Node head, int x)
    {
        // Base case
        if (head == null)
            return false;

        // If key is present in current node,
        // return true
        if (head.data == x)
            return true;

        // Recur for remaining list
        return search(head.next, x);
    }

    // Driver function to test the above functions
    public static void main(String args[])
    {
        // Start with the empty list
        LinkedList llist = new LinkedList();

        /* Use push() to construct below list
           14->21->11->30->10  */
        llist.push(10);
        llist.push(30);
        llist.push(11);
        llist.push(21);
        llist.push(14);

        if (llist.search(llist.head, 21))
            System.out.println("Yes");
        else
            System.out.println("No");
    }
}
// This code is contributed by Pratik Agarwal
```

Output:

```
Yes
```

Python

```python
# Recursive Python program to
# search an element in linked list

# Node class
class Node:

    # Function to initialise
    # the node object
    def __init__(self, data):
        self.data = data # Assign data
        self.next = None # Initialize next as null

class LinkedList:

    def __init__(self):
        self.head = None # Initialize head as None

    # This function insert a new node at
    # the beginning of the linked list
    def push(self, new_data):

        # Create a new Node
        new_node = Node(new_data)

        # Make next of new Node as head
        new_node.next = self.head

        # Move the head to
        # point to new Node
        self.head = new_node


    # Checks whether the value key
    # is present in linked list
    def search(self, li, key):

        # Base case
        if(not li):
            return False

        # If key is present in
        # current node, return true
        if(li.data == key):
            return True

        # Recur for remaining list
        return self.search(li.next, key)

# Driver Code
if __name__=='__main__':

    li = LinkedList()

    li.push(1)
    li.push(2)
    li.push(3)
    li.push(4)

    key = 4

    if li.search(li.head,key):
        print("Yes")
    else:
        print("No")

# This code is contributed
# by Manoj Sharma
```

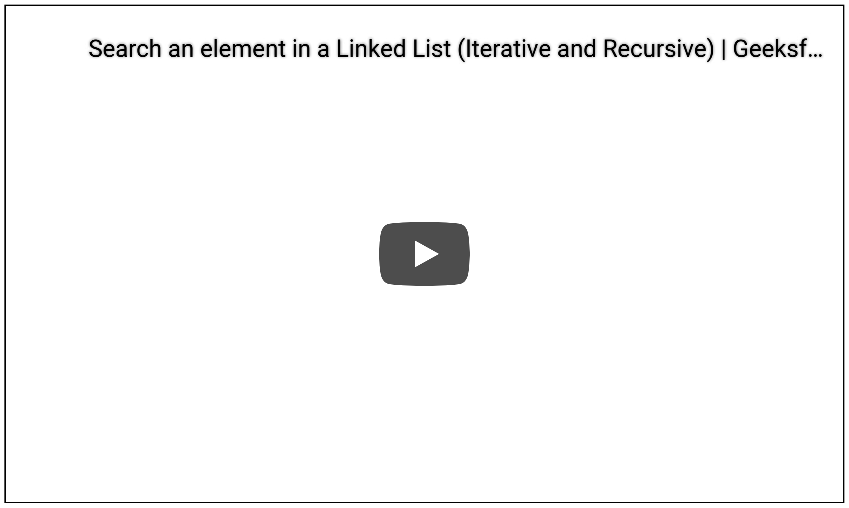
C#

```csharp
// Recursive C# program to search
// an element in linked list
```

```
using System;

// Node class
public class Node
{
    public int data;
    public Node next;
    public Node(int d)
    {
        data = d;
        next = null;
    }
}

// Linked list class
public class LinkedList
{
    Node head; //Head of list

    //Inserts a new node at the front of the list
    public void push(int new_data)
    {
        //Allocate new node and putting data
        Node new_node = new Node(new_data);

        //Make next of new node as head
        new_node.next = head;

        //Move the head to point to new Node
        head = new_node;
    }

    // Checks whether the value x is present
    // in linked list
    public bool search(Node head, int x)
    {
        // Base case
        if (head == null)
            return false;

        // If key is present in current node,
        // return true
        if (head.data == x)
            return true;

        // Recur for remaining list
        return search(head.next, x);
    }

    // Driver code
    public static void Main()
    {
        // Start with the empty list
        LinkedList llist = new LinkedList();

        /* Use push() to construct below list
        14->21->11->30->10 */
        llist.push(10);
        llist.push(30);
        llist.push(11);
        llist.push(21);
        llist.push(14);

        if (llist.search(llist.head, 21))
            Console.WriteLine("Yes");
        else
            Console.WriteLine("No");
    }
}

// This code is contributed by PrinciRaj1992
```

Output:

```
Yes
```


Search an element in a Linked List (Iterative and Recursive) | Geeksf...

This article is contributed by **Ravi**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Recommended Posts:

Find the middle of a given linked list in C and Java
Program for n'th node from the end of a Linked List
Write a function to get Nth node in a Linked List
Given only a pointer/reference to a node to be deleted in a singly linked list, how do you delete it?
Detect loop in a linked list
Write a function to delete a Linked List
Write a function that counts the number of times a given int occurs in a Linked List
Reverse a linked list
Given only a pointer to a node to be deleted in a singly linked list, how do you delete it?
Write a function to get the intersection point of two Linked Lists
Function to check if a singly linked list is palindrome
The Great Tree-List Recursion Problem.
Clone a linked list with next and random pointer | Set 1
Memory efficient doubly linked list
Given a linked list which is sorted, how will you insert in sorted way

Improved By : CodeDaemon, princiraj1992, Rajput-Ji, AnweshanTheProgrammer, rathbhupendra

Article Tags : Linked List

Practice Tags : Linked List

20

To-do Done

1.4

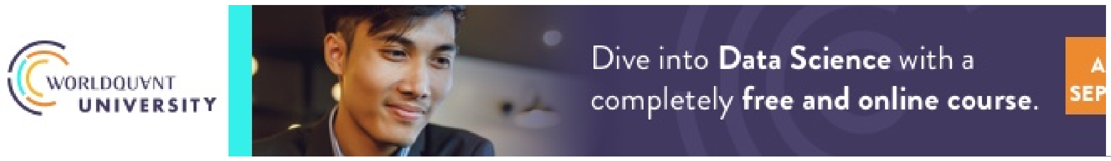
Based on **141** vote(s)

Feedback/ Suggest Improvement Notes Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments


WORLDQUANT
UNIVERSITY
Dive into Data Science with a completely free and online course.
A
SEP

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY
About Us
Careers
Privacy Policy
Contact Us

LEARN
Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE
Courses
Company-wise
Topic-wise
How to begin?

CONTRIBUTE
Write an Article
Write Interview Experience
Internships
Videos