GeeksforGeeks
A computer science portal for geeks
Practice IDE Q&A GeeksQuiz

Write a function that counts the number of times a given int occurs in a Linked List

Given a singly linked list and a key, count number of occurrences of given key in linked list. For example, if given linked list is 1->2->1->2->1->3->1 and given key is 1, then output should be 4.

Algorithm:

```
1. Initialize count as zero.
2. Loop through each element of linked list:
     a) If element data is equal to the passed number then
        increment the count.
3. Return count.
```

Implementation:

C/C++

```
// C/C++ program to count occurrences in a linked list
#include<stdio.h>
#include<stdlib.h>

/* Link list node */
struct node
{
    int data;
    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_data)
{
    /* allocate node */
    struct node* new_node =
            (struct node*) malloc(sizeof(struct node));
    /* put in the data  */
    new_node->data  = new_data;

    /* 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;
}

/* Counts the no. of occurences of a node
   (search_for) in a linked list (head)*/
int count(struct node* head, int search_for)
{
    struct node* current = head;
    int count = 0;
    while (current != NULL)
    {
        if (current->data == search_for)
           count++;
        current = current->next;
    }
    return count;
}

/* Drier program to test count function*/
int main()
{
    /* Start with the empty list */
    struct node* head = NULL;

    /* Use push() to construct below list
     1->2->1->3->1  */
    push(&head, 1);
    push(&head, 3);
    push(&head, 1);
    push(&head, 2);
    push(&head, 1);

    /* Check the count function */
    printf("count of 1 is %d", count(head, 1));
    return 0;
}
```

Java

```
// Java program to count occurrences in a linked list
class LinkedList
```

```
{
    Node head;  // head of list

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

    /* Inserts a new Node at front of the list. */
    public void push(int new_data)
    {
        /* 1 & 2: Allocate the Node &
                  Put in the data*/
        Node new_node = new Node(new_data);

        /* 3. Make next of new Node as head */
        new_node.next = head;

        /* 4. Move the head to point to new Node */
        head = new_node;
    }

    /* Counts the no. of occurences of a node
    (search_for) in a linked list (head)*/
    int count(int search_for)
    {
        Node current = head;
        int count = 0;
        while (current != null)
        {
            if (current.data == search_for)
                count++;
            current = current.next;
        }
        return count;
    }

    /* Drier function to test the above methods */
    public static void main(String args[])
    {
        LinkedList llist = new LinkedList();

        /* Use push() to construct below list
          1->2->1->3->1  */
        llist.push(1);
        llist.push(2);
        llist.push(1);
        llist.push(3);
        llist.push(1);

        /*Checking count function*/
        System.out.println("Count of 1 is "+llist.count(1));
    }
}
// This code is contributed by Rajat Mishra
```

Python

```
# Python program to count the number of time a given
# int occurs in a linked list

# Node class
class Node:

    # Constructor to initialize the node object
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:

    # Function to initialize head
    def __init__(self):
        self.head = None

    # Counts the no . of occurances of a node
    # (seach_for) in a linkded list (head)
    def count(self, search_for):
        current = self.head
        count = 0
        while(current is not None):
            if current.data == search_for:
                count += 1
            current = current.next
        return count

    # Function to insert a new node at the beginning
    def push(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node

    # Utility function to print the linked LinkedList
    def printList(self):
        temp = self.head
        while(temp):
            print temp.data,
            temp = temp.next


# Driver program
llist = LinkedList()
llist.push(1)
llist.push(3)
llist.push(1)
llist.push(2)
llist.push(1)

# Check for the count function
print "count of 1 is %d" %(llist.count(1))

# This code is contributed by Nikhil Kumar Singh(nickzuck_007)
```

Output:

```
count of 1 is 3
```

Time Complexity: O(n)
Auxiliary Space: O(1)

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

AMCAT Test for Freshers

Engineering Job in 1000+ Companies. 3 Lakh Interview Calls Each Month

myamcat.com/Fresher-Engg-Jobs

15 Comments Category: Linked Lists Tags: Linked Lists

Related Posts:

- Merge two sorted linked lists such that merged list is in reverse order
- Compare two strings represented as linked lists
- Rearrange a given linked list in-place.
- Sort a linked list that is sorted alternating ascending and descending orders?
- Select a Random Node from a Singly Linked List
- Merge Sort for Doubly Linked List
- Point to next higher value node in a linked list with an arbitrary pointer
- Swap nodes in a linked list without swapping data

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