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

Here is a solution.

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:

```
#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));
     getchar();
}
```

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

Related Topics:

- Clone a linked list with next and random pointer | Set 2
- Given a linked list of line segments, remove middle points
- Construct a Maximum Sum Linked List out of two Sorted Linked Lists having some Common nodes
- Given a linked list, reverse alternate nodes and append at the end
- Pairwise swap elements of a given linked list by changing links
- Self Organizing List | Set 1 (Introduction)
- Merge a linked list into another linked list at alternate positions
- QuickSort on Singly Linked List

Tags: Linked Lists

Like 1 Tweet 0 +1 0

Writing code in comment? Please use **ideone.com** and share the link here.

24 Comments GeeksforGeeks Login

Recommend 2 Share Sort by Newest

Join the discussion…

veena • 6 months ago

Just a suggestion: Instead of Using an integer Variable to count(Size = 4 bytes in java) why not use the head of the list to hold the data? This will eliminate count variable completely.

public int getRepetation(SingleNode head, int numberToCheck){

if(head.item == numberToCheck){

head.item = 1;

}else{

head.item = 0;

}

SingleNode temp = head.link;

while(temp != null){

if(temp.item == numberToCheck){

see more

Reply • Share ›

Ansuraj Khadanga veena • a month ago

Correct. But it will alter the original linked list: may or may not be favorable for all situations.

Reply • Share ›

piyush • 9 months ago

nice

Reply • Share ›

sathish piyush • 5 months ago

Good one

Reply • Share ›

Deepesh Panjabi • 9 months ago

http://ideone.com/fg0BfP

Reply • Share ›

SANTOSH KUMAR MISHRA • 10 months ago

```
int CountNumber(node *head,int num)
{
node *ptr = head;
int count = 0;
while(ptr != NULL)
{
if(ptr->data == num)
++count;
ptr = ptr->next;
}
return count;
```

}

4 • Reply • Share ›

deepuanand • a year ago

Via Tail Recursion...

```
int count_n_in_ll(node_t *head,int n)
{
static int count = 0;
if(head == NULL) {
if(count == 0) {
printf("either element not present in list or linklist is empty\n");
return -1;
}
return count;
}
if(head->data == n)
count++;
return count_n_in_ll(head->next,n);
}
```

4 • Reply • Share ›

DS+Algo=Placement deepuanand • 8 months ago

What is tail recursion?

• Reply • Share ›

Adauta Garcia Ariel DS+Algo=Placement • 3 months ago

Tail recursión is a way of recursión that some compilers are able to detect this kind of recursión and generate optimized code. There are 2 condicións for consider a function tail recursive.

1.the last statement mus be the recursive call. (this is why is called "tail recursión"
2.the recursive call must no be part of a expresión ie. You cant use (+,-,*,etc) and example of this.

return 4*recursiveCall(n-1); //avoid this
The reason why compilers optimize code it is because the 'stack frame' or 'activation record' are overwritten', instead of push a new one on the stack.
Sorry for my english. Hope this be useful.

2 • Reply • Share ›

darkprotocol Adauta Garcia Ariel • a month ago

Thanks

• Reply • Share ›

Sandeep • a year ago

```
public void countRepeated(int n){

Node main = start;

int count = 0;

if(main.getData() == n){ //To check for start node
count++;
}

while(main.getLink() != null){ //To check for remaining nodes excluding //last node

if(main.getData() == n){
count++;
}

main = main.getLink();
}

if(main.getData() == n){ //To check for last node
count++;
}

System.out.println("The count of repeated number is : " + count);

}
```

• Reply • Share ›

ravikant • 5 years ago

Common people post questions like these :P
They spoil such a good site !!

5 • Reply • Share ›

a.rookie.programmer ravikant • a year ago

this site is for common people.. if u think u are an exceptional programmer either go find a better site or make ur own.. btw thanks gfg for posting this..

13 • Reply • Share ›

Sudarshan ravikant • 2 years ago

Cool..man ..I have also astonished on this post but its ok...even a single person needs it its ok

8 • Reply • Share ›

abhishek08aug ravikant • 2 years ago
Wow! This is the most uncommon/retarded comment I came across ever on this site. :D
7 • Reply • Share ›

neha2210 ravikant • 2 years ago
Common people learn and become good programmers. I believe you think you were never a common person!
14 • Reply • Share ›

student ravikant • 4 years ago
what do you mean y common people? Are u a super hero or master of disasters something ? It is because of people like you that good is getting better and bad is getting worst
5 • Reply • Share ›

geeksforgeeks • 5 years ago
@Snehal: Time complexity is definitely O(n) but space complexity is O(1) as we are using constant extra space.
• Reply • Share ›

Shailedra geeksforgeeks • a year ago
i think Space complexity singly linked list is O(n)
• Reply • Share ›

Prateek Sharma geeksforgeeks • 2 years ago
I think Auxiliary space is o(1) but space complexity is o(n)...

```
/* Paste your code here (You may delete these lines if not writing code) */
```

2 • Reply • Share ›

GeeksforGeeks Prateek Sharma • 2 years ago
Thanks for pointing this out. We have updated the post.
• Reply • Share ›

Snehal • 5 years ago
I didnt get how it is O(1)?
anyway we need to traverse the complete linked list to count the occurrence of the element ?if you are assuming
n==(constant) and so it is o(1) , then it is wrong assumption,becoz at worst/base/avg case u need to move till end of the ll in the approach used by u
• Reply • Share ›

geeksforgeeks • 5 years ago

@Shikha: Thanks very much for pointing this out. We have corrected the space complexity.

^ | ⌄ • Reply • Share ›

Shikha • 5 years ago

Hi,
Space complexity is O(1) not O(n) here. (http://geeksforgeeks.org/?p=85...)

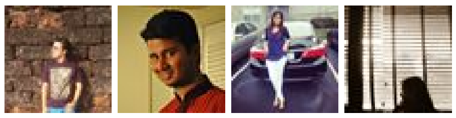
^ | ⌄ • Reply • Share ›

Subscribe Add Disqus to your site Privacy

-
- **GeeksforGeeks**

  Like

  93,235 people like GeeksforGeeks.

  Facebook social plugin
-
-
  - Interview Experiences
  - Advanced Data Structures
  - Dynamic Programming
  - Greedy Algorithms
  - Backtracking
  - Pattern Searching
  - Divide & Conquer
  - Mathematical Algorithms
  - Recursion
  - Geometric Algorithms
-

- ## Popular Posts
  - All permutations of a given string
  - Memory Layout of C Programs
  - Understanding "extern" keyword in C
  - Median of two sorted arrays
  - Tree traversal without recursion and without stack!
  - Structure Member Alignment, Padding and Data Packing
  - Intersection point of two Linked Lists
  - Lowest Common Ancestor in a BST.
  - Check if a binary tree is BST or not
  - Sorted Linked List to Balanced BST
- Follow @GeeksforGeeks Subscribe
- ## Recent Comments
  - lebron

    since the array size is 5, it takes constant...

    K'th Smallest/Largest Element in Unsorted Array | Set 3 (Worst Case Linear Time) · 3 hours ago
  - lebron

    merge sort

    K'th Smallest/Largest Element in Unsorted Array | Set 3 (Worst Case Linear Time) · 3 hours ago
  - Shubham Sharma

    You saved my time :)

    Searching for Patterns | Set 2 (KMP Algorithm) · 3 hours ago
  - Prakhar

    Why so many LOCs, if I'm not wrong (please...

    Largest Sum Contiguous Subarray · 4 hours ago
  - Aayush Gupta

    For R4 Q3, Another solution would be to use a...

    Amazon Interview Experience | Set 168 · 5 hours ago
  - EigenHarsha

For Power Of 2, We Simply Doing.. var1 =

Practo Interview Experience | Set 2 (Off-Campus) · 5 hours ago

-