Identical Linked Lists

Two Linked Lists are identical when they have same data and arrangement of data is also same. For example Linked lists a (1->2->3) and b(1->2->3) are identical. . Write a function to check if the given two linked lists are identical.

Method 1 (Iterative)
To identify if two lists are identical, we need to traverse both lists simultaneously, and while traversing we need to compare data.

```
#include<stdio.h>
#include<stdlib.h>

/* Structure for a linked list node */
struct node
{
  int data;
  struct node *next;
};

/* returns 1 if linked lists a and b are identical, otherwise 0 */
bool areIdentical(struct node *a, struct node *b)
{
  while(1)
  {
    /* base case */
    if(a == NULL && b == NULL)
    {  return 1; }
    if(a == NULL && b != NULL)
    {  return 0; }
    if(a != NULL && b == NULL)
    {  return 0; }
    if(a->data != b->data)
    {  return 0; }
```

Google™ Custom Search

GeeksforGeeks

53,528 people like GeeksforGeeks.

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

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
    /* If we reach here, then a and b are not NULL and their
       data is same, so move to next nodes in both lists */
    a = a->next;
    b = b->next;
  }
}

/* UTILITY FUNCTIONS TO TEST fun1() and fun2() */
/* 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;
}

/* Druver program to test above function */
int main()
{
  struct node *a = NULL;
  struct node *b = NULL;

  /* The constructed linked lists are :
   a: 3->2->1
   b: 3->2->1 */
  push(&a, 1);
  push(&a, 2);
  push(&a, 3);

  push(&b, 1);
  push(&b, 2);
  push(&b, 3);

  if(areIdentical(a, b) == 1)
    printf(" Linked Lists are identical ");
```

cloudera
RETHINK DATA
The Enterprise Data Hub

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

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
  else
    printf(" Linked Lists are not identical ");

  getchar();
  return 0;
}
```

Method 2 (Recursive)
Recursive solution code is much cleaner than the iterative code. You probably wouldn't want to use the recursive version for production code however, because it will use stack space which is proportional to the length of the lists

```
bool areIdentical(struct node *a, struct node *b)
{
  if (a == NULL && b == NULL)
  {   return 1;   }
  if (a == NULL && b != NULL)
  {   return 0;   }
  if (a != NULL && b == NULL)
  {   return 0;   }
  if (a->data != b->data)
  {   return 0;   }

  /* If we reach here, then a and b are not NULL and their
       data is same, so move to next nodes in both lists */
  return areIdentical(a->next, b->next);
}
```

Time Complexity: O(n) for both iterative and recursive versions. n is the length of the smaller list among a and b.

Please write comments if you find the above codes/algorithms incorrect, or find better ways to solve the same problem.

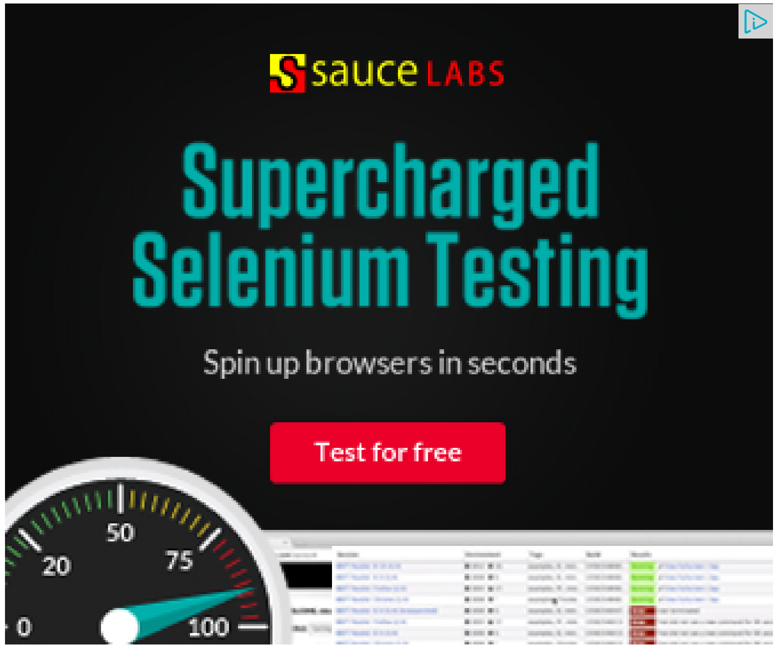
Shouldn't you expect a cloud with:

SYSTEM MONITORING

Plus the experts to help run it?

TRY MANAGED CLOUD ▸

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com



rackspace
the open cloud company

705 Subscribe

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 50 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 1 hour ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 1 hour ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 2 hours ago

Related Tpoics:

- 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
- Delete N nodes after M nodes of a linked list
- Design a stack with operations on middle element
- Swap Kth node from beginning with Kth node from end in a Linked List

6 Tweet 0 0

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

14 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

Himanshu Dagar • 3 months ago

can refer to below code for recursion method

http://ideone.com/Bx7UEb

Reply • Share ›

neelabhsingh • 6 months ago

what is problem in this method

```
bool areIdentical(struct node *)
{
while(1)
{
if((a==NULL)&&(b==NULL))
return 1;
if(a->data==b->data)
{
a=a->next;
b=b->next;

}
else
return 0;
}
}
```

Reply • Share ›

mahesh neelabhsingh • 5 months ago

@neelabhsingh consider two list viz. L1=1->2->NULL and L2=1->2->
L2 will give segmentation fault. Due to this condition (a->data==b->dat

AdChoices
▶ C++ Code
▶ Linked List
▶ Linked Data
AdChoices
▶ Programming C++
▶ Java Array
▶ Null Pointer
AdChoices
▶ C++ Example
▶ C++ Program
▶ Excel Function

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

error. Your code works for equal list only.

Reply • Share ›

neelabhsingh mahesh • 4 months ago

thanks for explanation.

Reply • Share ›

Sumit Gaur • 9 months ago

bool idendical (node *x, node *y).
{

if(x==NULL&&y==NULL)
return true;
return ((x->data==y->data)&&identical(x->next, y->next));
}

Reply • Share ›

Saurav Sahu Sumit Gaur • 8 months ago

That will cause Segmentation fault if both lists are not of equal length.

2 • Reply • Share ›

Deepak Singh • a year ago

thanks for such a beautiful explanation. now concept of linked list isn&#039t to

Reply • Share ›

cyberlynxs • 2 years ago

In method 2(recursive soln), the function is tail-recursive. If the compilers impl
compilers support it), a single stack-frame will be used. So, I think recursive s
please confirm it?

Reply • Share ›

Ashish • 4 years ago

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

check if a and b are pointing to same node, then the lists would be same by de

the case can also be a Y shaped two LL. so the moment address matches, fe

^ | v • Reply • Share ›

Sambasiva • 4 years ago

```
int areIdentical(list a, list b)
{
     for(; a && b && a->data == b->data; a = a->next, b = b->next);
     return !(a || b);
}
```

3 ^ | v • Reply • Share ›

abhikumar18 ➜ Sambasiva • 10 months ago

awesome yar...

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

^ | v • Reply • Share ›

piyush ➜ Sambasiva • 2 years ago

GREAT..........

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

^ | v • Reply • Share ›

kapil • 4 years ago

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

How about comparing two linked list having same set of elements and same r
here is that elements can be in any order.

Reply • Share ›

kartik kapil • 4 years ago

There can be two ways to solve this:

1) Sort both lists in O(mLogm + nLogn). After sorting, use the areIdent

2) Use Hashing

Reply • Share ›

Subscribe Add Disqus to your site

@geeksforgeeks, **Some rights reserved** **Contact Us!** Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com