Compare two strings represented as linked lists

Given two linked lists, represented as linked lists (every character is a node in linked list). Write a function compare() that works similar to strcmp(), i.e., it returns 0 if both strings are same, 1 if first linked list is lexicographically greater, and -1 if second string is lexicographically greater.

Examples:

```
Input: list1 = g->e->e->k->s->a
       list2 = g->e->e->k->s->b
Output: -1

Input: list1 = g->e->e->k->s->a
       list2 = g->e->e->k->s
Output: 1

Input: list1 = g->e->e->k->s
       list2 = g->e->e->k->s
Output: 0
```

We strongly recommend that you click here and practice it, before moving on to the solution.

C++

```
// C++ program to compare two strings represented as linked
// lists
#include<bits/stdc++.h>
using namespace std;

// Linked list Node structure
struct Node
{
    char c;
    struct Node *next;
};

// Function to create newNode in a linkedlist
Node* newNode(char c)
```

```
{
    Node *temp = new Node;
    temp->c = c;
    temp->next = NULL;
    return temp;
};

int compare(Node *list1, Node *list2)
{
    // Traverse both lists. Stop when either end of a linked
    // list is reached or current characters don't match
    while (list1 && list2 && list1->c == list2->c)
    {
        list1 = list1->next;
        list2 = list2->next;
    }

    //  If both lists are not empty, compare mismatching
    //  characters
    if (list1 && list2)
        return (list1->c > list2->c)? 1: -1;

    // If either of the two lists has reached end
    if (list1 && !list2) return 1;
    if (list2 && !list1) return -1;

    // If none of the above conditions is true, both
    // lists have reached end
    return 0;
}

// Driver program
int main()
{
    Node *list1 = newNode('g');
    list1->next = newNode('e');
    list1->next->next = newNode('e');
    list1->next->next->next = newNode('k');
    list1->next->next->next->next = newNode('s');
    list1->next->next->next->next->next = newNode('b');

    Node *list2 = newNode('g');
    list2->next = newNode('e');
    list2->next->next = newNode('e');
    list2->next->next->next = newNode('k');
    list2->next->next->next->next = newNode('s');
    list2->next->next->next->next->next = newNode('a');

    cout << compare(list1, list2);

    return 0;
}
```

Run on IDE

Java

```
// Java program to compare two strings represented as a linked list

// Linked List Class
class LinkedList {

    Node head;  // head of list
    static Node a, b;

    /* Node Class */
    static class Node {
```

```
        char data;
        Node next;

        // Constructor to create a new node
        Node(char d) {
            data = d;
            next = null;
        }
    }

    int compare(Node node1, Node node2) {

        if (node1 == null && node2 == null) {
            return 1;
        }
        while (node1 != null && node2 != null && node1.data == node2.data) {
            node1 = node1.next;
            node2 = node2.next;
        }

        // if the list are diffrent in size
        if (node1 != null && node2 != null) {
            return (node1.data > node2.data ? 1 : -1);
        }

        // if either of the list has reached end
        if (node1 != null && node2 == null) {
            return 1;
        }
        if (node1 == null && node2 != null) {
            return -1;
        }
        return 0;
    }

    public static void main(String[] args) {

        LinkedList list = new LinkedList();
        Node result = null;

        list.a = new Node('g');
        list.a.next = new Node('e');
        list.a.next.next = new Node('e');
        list.a.next.next.next = new Node('k');
        list.a.next.next.next.next = new Node('s');
        list.a.next.next.next.next.next = new Node('b');

        list.b = new Node('g');
        list.b.next = new Node('e');
        list.b.next.next = new Node('e');
        list.b.next.next.next = new Node('k');
        list.b.next.next.next.next = new Node('s');
        list.b.next.next.next.next.next = new Node('a');

        int value;
        value = list.compare(a, b);
        System.out.println(value);

    }
}

// This code has been contributed by Mayank Jaiswal
```

Run on IDE

Python

```
# Python program to compare two strings represented as
# linked lists

# A linked list node structure
class Node:

    # Constructor to create a new node
    def __init__(self, key):
        self.c = key ;
        self.next = None

def compare(list1, list2):

    # Traverse both lists. Stop when either end of linked
    # list is reached or current characters don't watch
    while(list1 and list2 and list1.c == list2.c):
        list1 = list1.next
        list2 = list2.next

    # If both lists are not empty, compare mismatching
    # characters
    if(list1 and list2):
        return 1 if list1.c > list2.c else -1

    # If either of the two lists has reached end
    if (list1 and not list2):
        return 1

    if (list2 and not list1):
        return -1
    return 0

# Driver program

list1 = Node('g')
list1.next = Node('e')
list1.next.next = Node('e')
list1.next.next.next = Node('k')
list1.next.next.next.next = Node('s')
list1.next.next.next.next.next = Node('b')

list2 = Node('g')
list2.next = Node('e')
list2.next.next = Node('e')
list2.next.next.next = Node('k')
list2.next.next.next.next = Node('s')
list2.next.next.next.next.next = Node('a')

print compare(list1, list2)

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

Run on IDE

Output:

```
1
```

Thanks to Gaurav Ahirwar for suggesting above implementation.

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

Company Wise Coding Practice Topic Wise Coding Practice

24 Comments Category: Linked Lists

Related Posts:

- Convert a Binary Tree to a Circular Doubly Link List
- Subtract Two Numbers represented as Linked Lists
- Rearrange a given list such that it consists of alternating minimum maximum elements
- Flatten a multi-level linked list | Set 2 (Depth wise)
- Decimal Equivalent of Binary Linked List
- Merge K sorted linked lists
- Check if a linked list is Circular Linked List
- Delete middle of linked list

(Login to Rate and Mark)

1.3 Average Difficulty : **1.3/5.0**
Based on **61** vote(s)

Add to TODO List

Mark as DONE

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

24 Comments **GeeksforGeeks** 1 **Login**

Recommend 1 **Share** **Sort by Newest**

Join the discussion…

JavaCoder • a month ago

Java Implementation:

When both are null, shouldn't it return 0 ?

```
if (node1 == null && node2 == null) {
return 0;
}
```

and here:

```
// if the list are different in size
if (node1 != null && node2 != null) {
return (node1.data > node2.data ? 1 : -1);
}
```

the comment should be "// if the data is different".

Reply • Share ›

Akshay Mittal • 3 months ago

There is bug in this code. It will return an incorrect answer(-1) when we will compare g->e->a->z and g->e->k. The reason is length of the linklists are not taken in account when there is a mismatch of a character.

1 Reply • Share ›

Musarrat_123 ➚ Akshay Mittal • 3 months ago

The correct answer is -1 only because the string geaz is lexicographically smaller than the string gek since 'a' comes before 'k'.
The comparison is not on the basis of lengths but on the basis of dictionary order.

2 Reply • Share ›

bhavik gujarati • 5 months ago

My solution:

http://ideone.com/g4erF9

Reply • Share ›

rishabh mamgain • 6 months ago

simple solution in c++ using List STL

http://code.geeksforgeeks.org/...

Reply • Share ›

Rishabh Kaushik • 7 months ago

```
#include<iostream>

using namespace std;

class List

{

private:

struct node

{

char letter;

node *link;

}*head_list;

public:
```

see more

• Reply • Share ›

Rohit • 7 months ago

Good implementation.

• Reply • Share ›

palindname • 8 months ago

typo in there,could change

if (list1 && !list2) return 1; //should be -1

if (list2 && !list1) return -1;

to

if ((!list1 && list2) || (list1 && !list2) return -1;`

• Reply • Share ›

Devendra Patil • 9 months ago

In JAVA:

http://code.geeksforgeeks.org/...

• Reply • Share ›

saiteja • 10 months ago

There is bug in this..
after scanning the common char in lists,it only compares the next char and in both the lists

example:-compare (geeksa and geeksb) it returns 'geeksb'.....
if u compare (geeksabc and geeksb) it still returns 'geeksb' as answer
thats the mistake
think about it...sorry for poor english

▲ | ▼ • Reply • Share ›

Sai Teja ➚ saiteja • 10 months ago
Sorry...this post is wrong...dont mind

▲ | ▼ • Reply • Share ›

Rahul Bhambri • 10 months ago
python version:
https://github.com/rbhambriiit...

▲ | ▼ • Reply • Share ›

Pankaj Dabade ➚ Rahul Bhambri • 9 months ago
I think if you are doing it in python then the easiest way is to convert this sequence to tuple. Tuples are comparable.

▲ | ▼ • Reply • Share ›

Rahul Bhambri ➚ Pankaj Dabade • 9 months ago
Still requires you to form a tuple from the given linked list, which you shall compare.

▲ | ▼ • Reply • Share ›

Supreeth • a year ago

```
import java.util.LinkedList;

public class ComparetwoStringsLinkedList {

//Compares the firstlinkedlist with secondlinkedlist
public static boolean doesntcontain (LinkedList l1, LinkedList l2){

for ( Object c : l1){

if (l1.contains(l2)){

//If firstlist contains secondlist then return true and also print 1

System.out.println(1);

return true;

}else {

//If firstlist doesn't contain secondlist then return false and also print -1
```

see more

• Reply • Share ›

Ishaan Arora • a year ago

Java Code for the same problem

```
import java.util.*;

import java.lang.*;

public class geek

{

public static void main(String[] args)

{

//System.out.println("let us begin");

LinkedList<string> l1=new LinkedList<string>();

LinkedList<string> l2=new LinkedList<string>();

l1.add("g");
```

see more

• Reply • Share ›

Akansh • a year ago

```
public int compare(Node listA, Node listB) {
if (listA == null || listB == null) {
return 1;
}

do {
if (listA == null || listB == null) {
return 1;
}

if (listA.getValue() == listB.getValue()) {
listA = listA.getNextNode();
listB = listB.getNextNode();
} else {
return -1;
}
} while (listA != null || listB !=null);

return 0;
}
```

• Reply • Share ›

Meenakshi • a year ago

int compare_lists(struct Node *list_1,struct Node *list_2)

{

string s1 = "";

string s2 = "";

while(list_1&&list_2)

{

s1.push_back(list_1->c);

s2.push_back(list_2->c);

list_1=list_1->next;

list_2=list_2->next;

if((list_1!=NULL)&&(list_2!=NULL))

see more

• Reply • Share ›

V_CODER • a year ago

//recursion:

int cmp(node *head1, node *head2)

{
if(head1 == NULL && head2 !=NULL) return -1;
if(head2 == NULL && head1 !=NULL) return 1;
else if(head1->data > head2->data) return 1;
else if(head2->data > head1->data) return -1;
else if(head1 == NULL && head2== NULL) return 0;

cmp(head1->next,head2->next);
}

//pls reply if any bug in code / logic

• Reply • Share ›

#InnerPeace • a year ago

Optimization of code

if (list1 && list2)
return (list1->c > list2->c)? 1: -1;
// If either of the two lists has reached end

else if (list1 && !list2) return 1;

else (list2 && !list1) return -1;

Note :No need to check each condition as only one condition would be match at a time.

Reply • Share ›

Mysterious Mind ➚ #InnerPeace • a year ago

return statement is there in body of if block. So later if condition won't be checked.

Reply • Share ›

#InnerPeace ➚ Mysterious Mind • a year ago

@Mysterious Mind Thanks for your inputs ! Agreed upon what you said.

Reply • Share ›

ramswish ➚ #InnerPeace • a year ago

This Doesn't work!!!
with your code it will never return 0(success case).
with working principle of if() elseif() else work

Reply • Share ›

Kartik ➚ #InnerPeace • a year ago

This optimization doesn't seem to work. Condition (!list2 && !list1) can also be true. That is why 0 is returned at the end. Please correct me if I am wrong.

Reply • Share ›