206. Reverse Linked List

March 5, 2016 | 466K views

★★★★★
Average Rating: 4.71 (217 votes)

Reverse a singly linked list.

Example:

```
Input: 1->2->3->4->5->NULL
Output: 5->4->3->2->1->NULL
```

Follow up:

A linked list can be reversed either iteratively or recursively. Could you implement both?

Solution

Approach #1 (Iterative) [Accepted]

Assume that we have linked list `1 → 2 → 3 → Ø`, we would like to change it to `Ø ← 1 ← 2 ← 3`.

While you are traversing the list, change the current node's next pointer to point to its previous element. Since a node does not have reference to its previous node, you must store its previous element beforehand. You also need another pointer to store the next node before changing the reference. Do not forget to return the new head reference at the end!

```
public ListNode reverseList(ListNode head) {
    ListNode prev = null;
    ListNode curr = head;
    while (curr != null) {
        ListNode nextTemp = curr.next;
        curr.next = prev;
        prev = curr;
        curr = nextTemp;
    }
    return prev;
}
```

Complexity analysis

- Time complexity : $O(n)$. Assume that n is the list's length, the time complexity is $O(n)$.
- Space complexity : $O(1)$.

Approach #2 (Recursive) [Accepted]

The recursive version is slightly trickier and the key is to work backwards. Assume that the rest of the list had already been reversed, now how do I reverse the front part? Let's assume the list is: $n_1 \to \dots \to n_{k-1} \to n_k \to n_{k+1} \to \dots \to n_m \to Ø$

Assume from node n_{k+1} to n_m had been reversed and you are at node n_k.

$n_1 \to \dots \to n_{k-1} \to \mathbf{n_k} \to n_{k+1} \leftarrow \dots \leftarrow n_m$

We want n_{k+1}'s next node to point to n_k.

So,

n_k.next.next = n_k;

Be very careful that n_1's next must point to Ø. If you forget about this, your linked list has a cycle in it. This bug could be caught if you test your code with a linked list of size 2.

```
public ListNode reverseList(ListNode head) {
    if (head == null || head.next == null) return head;
    ListNode p = reverseList(head.next);
    head.next.next = head;
    head.next = null;
    return p;
}
```

Complexity analysis

- Time complexity : $O(n)$. Assume that n is the list's length, the time complexity is $O(n)$.
- Space complexity : $O(n)$. The extra space comes from implicit stack space due to recursion. The recursion could go up to n levels deep.

Rate this article: ★★★★★

Previous Next

Comments: 93 Sort By

Type comment here... (Markdown is supported)

Preview Post

JokerLbz ★ 305 March 19, 2019 8:28 AM Report

I found a brilliant solution about Recursive with visual animation.
https://www.youtube.com/watch?v=MRe3UsRadKw

Read More

300 Share Reply

SHOW 7 REPLIES

moby ★ 90 July 27, 2018 8:16 AM

Python: Iterative & Recursive

```
class Solution(object):
    def reverseList(self, head):  # Iterative
        prev, curr = None, head
```

Read More

87 Share Reply

SHOW 5 REPLIES

MaxNRG ★ 128 April 18, 2018 2:29 PM Report

Runtime: 40 ms beats 99.86 % of python3 submissions.

Read More

45 Share Reply

SHOW 6 REPLIES

sofs1 ★ 727 March 2, 2019 11:09 AM

I still can't wrap my head around

head.next.next = head;
head.next = null;

Read More

38 Share Reply

SHOW 13 REPLIES

terriblewhiteboard ★ 3839 May 3, 2020 5:58 PM Report

I made a video if anyone is having trouble understanding the iterative solution
https://www.youtube.com/watch?v=jXArj8RBq0Q&feature=youtu.be

Read More

20 Share Reply

SHOW 1 REPLY

shwetas16 ★ 26 January 29, 2018 1:01 AM

What about pushing all values onto a stack and then building out the list as you pop from the stack? That would be O(n) time and O(n) space.

18 Share Reply

SHOW 12 REPLIES

dayatar ★ 46 February 19, 2019 1:05 AM Report

Python Solution(iterative)

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, x):
```

Read More

10 Share Reply

SHOW 1 REPLY

angel5 ★ 9 August 16, 2018 7:01 PM

```

/**
 * Definition for singly-linked list.
 * public class ListNode {
```

Read More

9 Share Reply

zhuqingguang ★ 6 December 14, 2017 1:51 PM Report

JavaScript Solution with O(n) time, O(1) space.

- iterative function

```
var reverseList = function(head) {
```

Read More

6 Share Reply

offer_wxy ★ 6 December 23, 2018 9:08 PM Report

```
class Solution(object):
    def reverseList(self, head):
        pre = None
        cur = head
```

Read More

5 Share Reply

‹ 1 2 3 4 5 6 ... 9 10 ›