234-回文链表

题述

234. 回文链表

难度 简单 1511

给你一个单链表的头节点 `head` ，请你判断该链表是否为回文链表。如果是，返回 `true` ；否则，返回 `false` 。

示例 1：

```
输入：head = [1,2,2,1]
输出：true
```

示例 2：

```
输入：head = [1,2]
输出：false
```

提示：

- 链表中节点数目在范围 $[1, 10^5]$ 内
- 0 <= Node.val <= 9

思路

数组模拟

- 最直接的想法就是把链表装成数组，然后再判断是否回文
- ```
  class Solution {
  public:
      bool isPalindrome(ListNode* head)
      {
          vector<int> vec;
          ListNode* cur=head;
          while(cur)
          {
              vec.push_back(cur->val);
              cur=cur->next;
          }

          //比较数组回文
          for(int i=0,j=vec.size()-1;i<j;i++,j--)
          {
              if(vec[i]!=vec[j])
  ```
  ```

```
            {
                return false;
            }
        }
        return true;
    }
};
```

- 一种优化方式是先求出链表长度，然后给定vector的初始长度，以避免vector每次添加节点重新开辟空间

题解

C++数组模拟优化版

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    bool isPalindrome(ListNode* head)
    {
        ListNode* cur=head;
        int length = 0;
        while(cur)
        {
            length++;
            cur=cur->next;
        }
        vector<int> vec(length,0);  //给定vector初始长度，避免每次重新开辟
        cur=head;
        int index =0;
        while(cur)
        {
            vec[index++]=cur->val;
            cur=cur->next;
        }

        //比较数组回文 类似双指针
        for(int i=0,j=vec.size()-1;i<j;i++,j--)
        {
            if(vec[i]!=vec[j])
            {
                return false;
            }
        }
        return true;
    }
};
```

思考


执行结果： 通过 显示详情 ›
添加备
执行用时： 196 ms ，在所有 C++ 提交中击败了 49.15% 的用户
内存消耗： 119.3 MB ，在所有 C++ 提交中击败了 42.10% 的用户
通过测试用例： 88 / 88
炫耀一下：
写题解，分享我的解题思路
提交结果 执行用时 内存消耗 语言 提交时间 备注
通过 196 ms 119.3 MB C++ 2022/09/20 10:36 添加