24. Swap Nodes in Pairs ★

Question | Editorial Solution | My Submissions (/problems/swap-nodes-in-pairs/submissions/)

Total Accepted: **121025** Total Submissions: **331721** Difficulty: **Easy**

Given a linked list, swap every two adjacent nodes and return its head.

For example,

Given `1->2->3->4`, you should return the list as `2->1->4->3`.

Your algorithm should use only constant space. You may **not** modify the values in the list, only nodes itself can be changed.

Subscribe (/subscribe/) to see which companies asked this question

Show Tags

Show Similar Problems

Have you met this question in a real interview? Yes No

Discuss (https://leetcode.com/discuss/questions/oj/swap-nodes-in-pairs)

Pick One (/problems/random-one-question/)

C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* swapPairs(ListNode* head) {
        if (!head) return head;
        ListNode* rev1st = head->next, *rev2nd = head;
        if (!rev1st) return head;
        ListNode dummy(-1);
        ListNode *ptr = &dummy;

        while (rev1st) {
            ListNode* temp = rev1st->next;
            ptr->next = rev1st;
            ptr = ptr->next;
            ptr->next = rev2nd;
            ptr = ptr->next;
            if (temp) {
```

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

```
25                  rev2nd = temp;
26                  rev1st = rev2nd->next;
27              }
28              else {
29                  break;
30              }
31          }
32          if (!rev1st) {
33              ptr->next = rev2nd;
34              ptr = ptr->next;
35          }
36          ptr->next = NULL;
37          return dummy.next;
38      }
39  };
```

Notes

Custom Testcase

Run Code | Submit Solution