

Swap Nodes in Pairs

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

Example:

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

Note:

Your algorithm should use only constant extra space.

You may not modify the values in the list's nodes, only nodes itself may be changed.

Solution in 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 NULL;

        ListNode *node1=head;
        ListNode *node2=head->next;

        if (node2) {
            // swap
            node1->next=swapPairs(node2->next);
            node2->next=node1;

            return node2;
        }
        else return node1;
    }
};
```