024. Swap Node in Pairs

024 Swap Nodes in Pairs

• Linked List

Description

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.

1. Thought line

2. Linked List

```
2 * Definition for singly-linked list.
 3 * struct ListNode {
        int val:
         ListNode *next;
        ListNode(int x) : val(x), next(NULL) {}
 6 *
7 * };
9 class Solution {
10 public:
    ListNode* swapPairs(ListNode* head) {
12
          ListNode* dummyHead = new ListNode (0);
13
          dummyHead->next = head;
14
        ListNode* ptrBeforeOdd = dummyHead;
15
        ListNode* ptrOdd = dummyHead;
          ListNode* ptrEven = dummyHead;
16
17
          while (ptrBeforeOdd !=nullptr){
18
            // Only when odd and even both exist, continue
19
              if (ptr0dd->next != nullptr && ptr0dd->next->next !=nullptr){
20
                 ptr0dd = ptr0dd->next;
21
                 ptrEven = ptrOdd->next;
22
                 ptr0dd->next = ptrEven->next;
23
                  ptrEven->next = ptr0dd;
                  ptrBeforeOdd->next = ptrEven;
25
                  ptrBeforeOdd = ptrEven = ptrOdd;
26
27
              else break:
28
29
          return dummyHead->next;
30
```