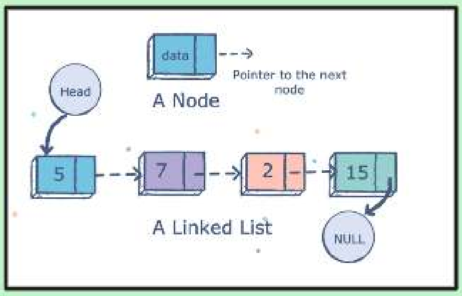
Swap Nodes in Pairs (Leetcode-24)

@manojofficialmj

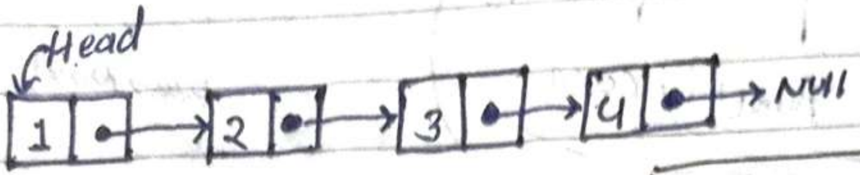


amazon Microsoft Meta Bloomberg

Leetcode-24 Swap Nodes in pairs

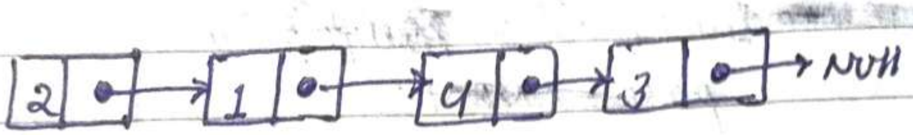
@manoj official MJ

Given Linked List



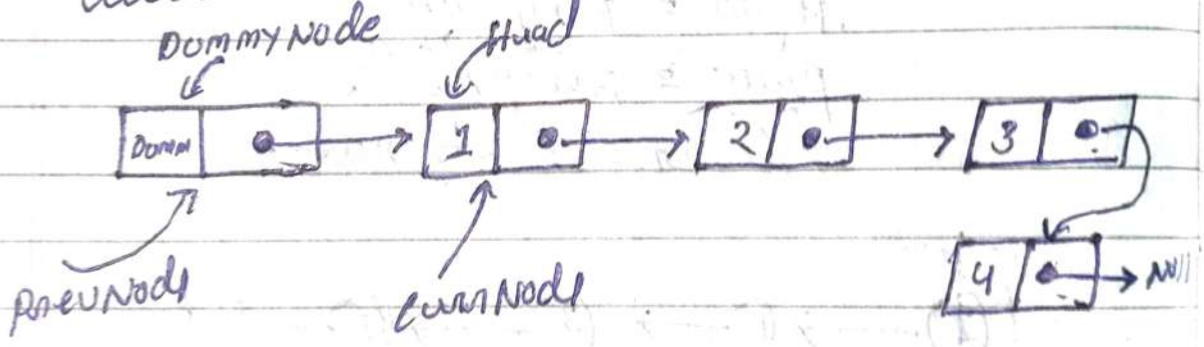
T.C. = O(N)
S.C. = O(1)

Output



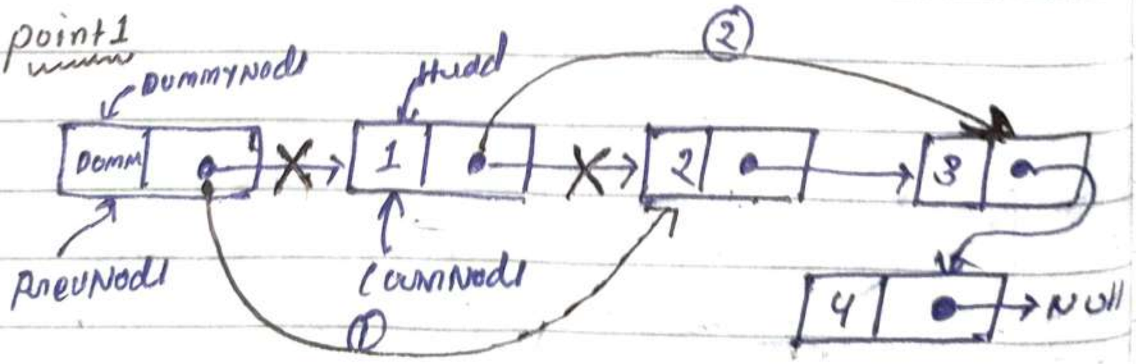
Approach

Initial state



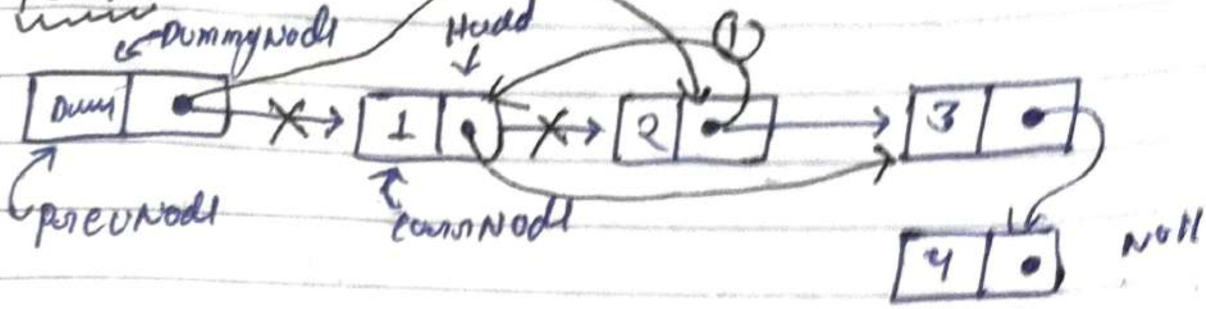
```
ListNode* DummyNode = new ListNode();
ListNode* prevNode = dummNode;
ListNode* CurrNode = Head;
```

Swap point1



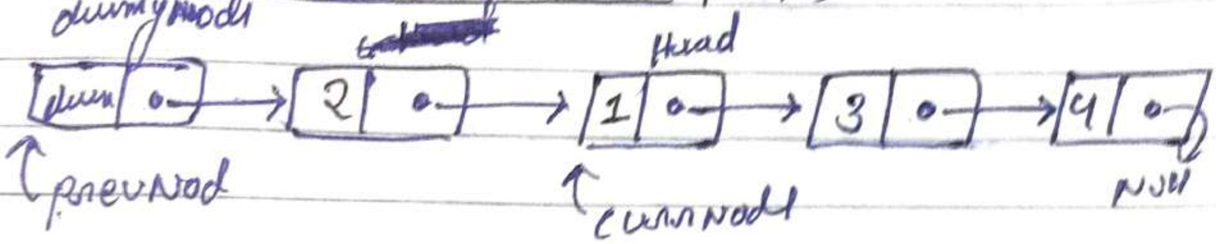
① prevNode→next = CurrNode→next;
② Curr→next = prevNode→next→next;

point2



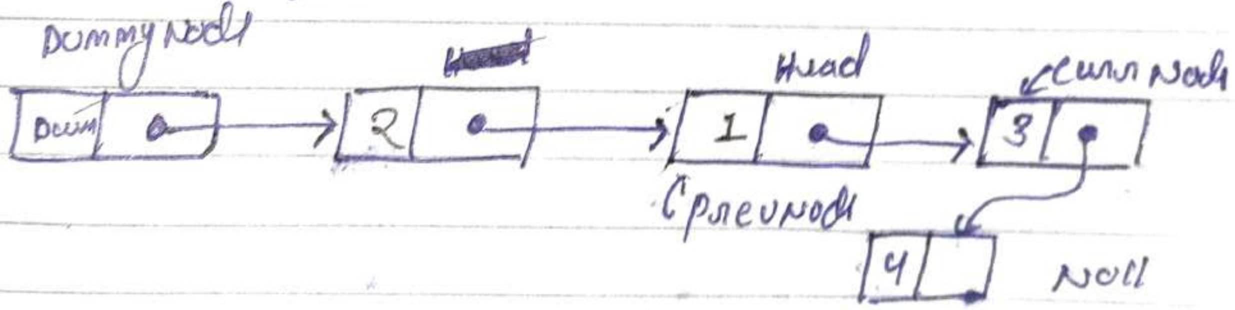
① prevNode→next→next = currNode;

1st Iteration swap(1,2)



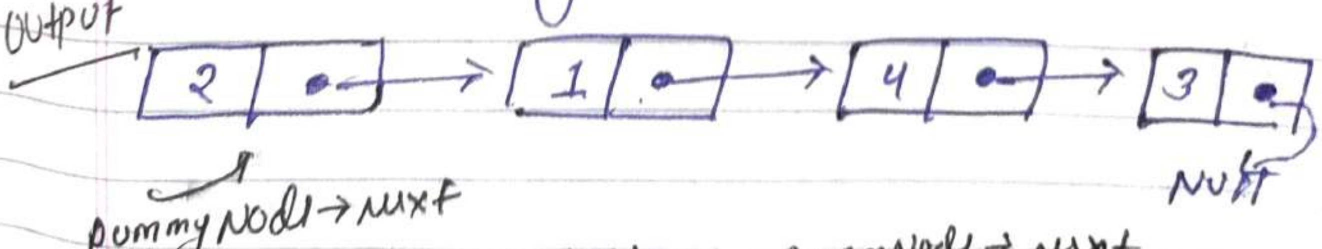
Now update

prevNode = currNode;
currNode = currNode→next;

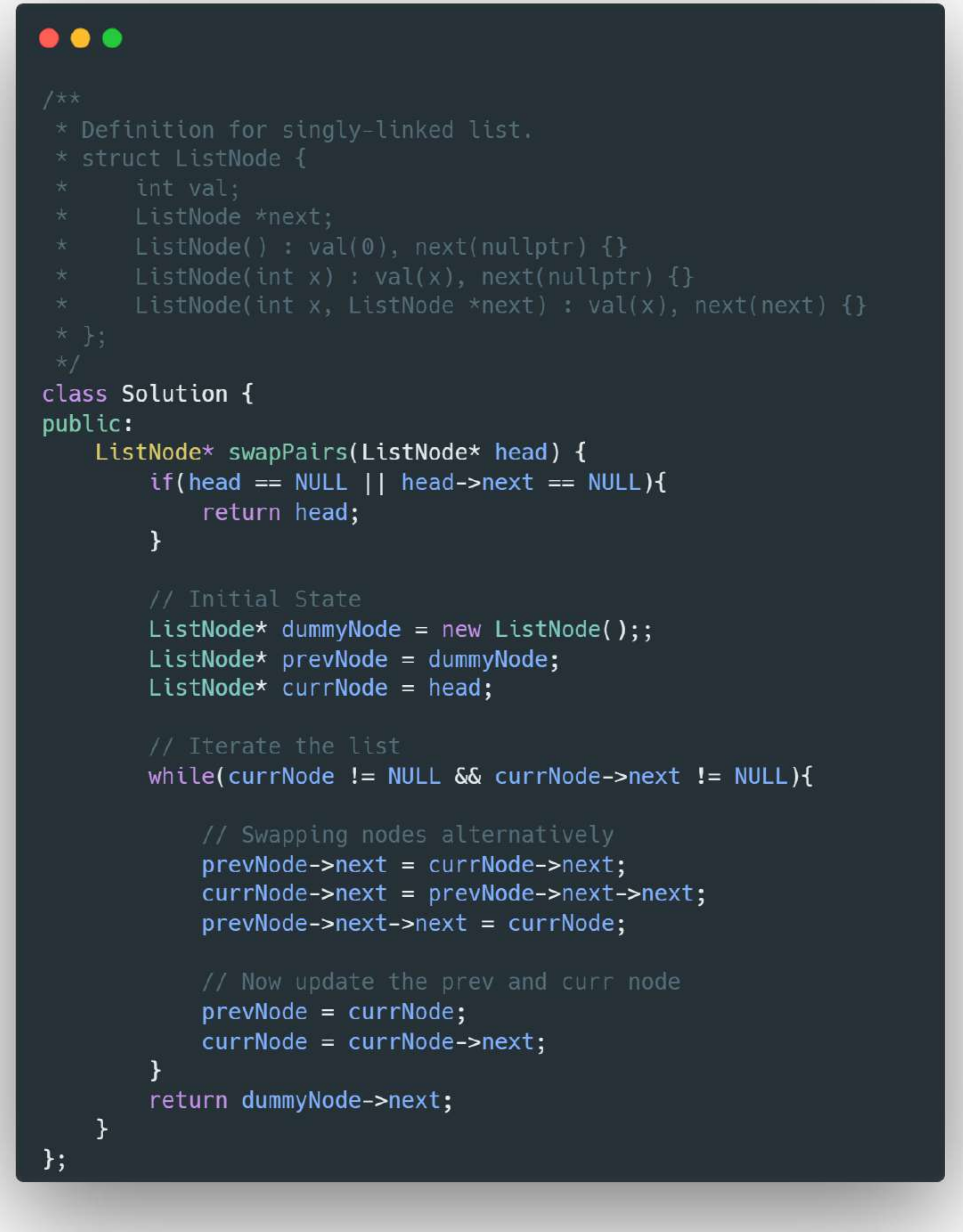


Repeat point1 and point 2 until
All swap Na Ho jaye

After All swaping Node



return dummNode → next



```cpp
/**
 * 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:
    ListNode* swapPairs(ListNode* head) {
        if(head == NULL || head->next == NULL){
            return head;
        }

        // Initial State
        ListNode* dummyNode = new ListNode();;
        ListNode* prevNode = dummyNode;
        ListNode* currNode = head;

        // Iterate the list
        while(currNode != NULL && currNode->next != NULL){

            // Swapping nodes alternatively
            prevNode->next = currNode->next;
            currNode->next = prevNode->next->next;
            prevNode->next->next = currNode;

            // Now update the prev and curr node
            prevNode = currNode;
            currNode = currNode->next;
        }
        return dummyNode->next;
    }
};
```