

# Swap Nodes in Pairs

## Leetcode 24.Swap Nodes in Pairs

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

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

### Example:

Given 1->2->3->4,

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

```
#Author:kilien
#Leetcode 24.Swap Nodes in Pairs
#思路：定义哨兵节点，互换相邻节点及后继节点，详见图解
#time:O(n) space:O(n)
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, x):
#         self.val = x
#         self.next = None

class Solution:
    def swapPairs(self, head: ListNode) -> ListNode:
        pre, pre.next = self, head
        while pre.next and pre.next.next:
            a = pre.next
            b = a.next
            pre.next, b.next, a.next = b, a, b.next
            pre = a
        return self.next
```

初始状态：

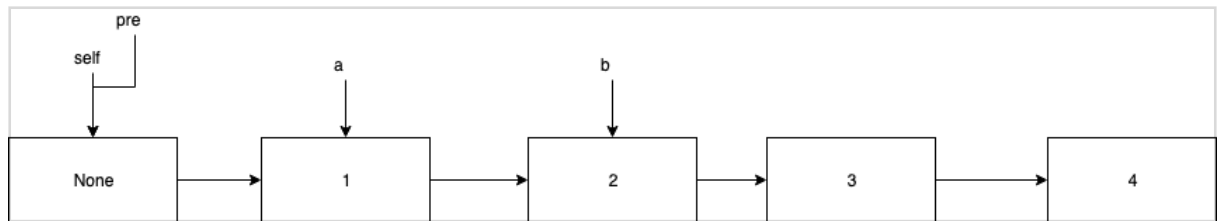
pre代表ListNode对象，指向自身None；

pre后继节点为head，即第一个节点

后续定义两个指针变量：a,b

a指向pre的后继节点

b指向a指针的后继节点，即value为2的节点



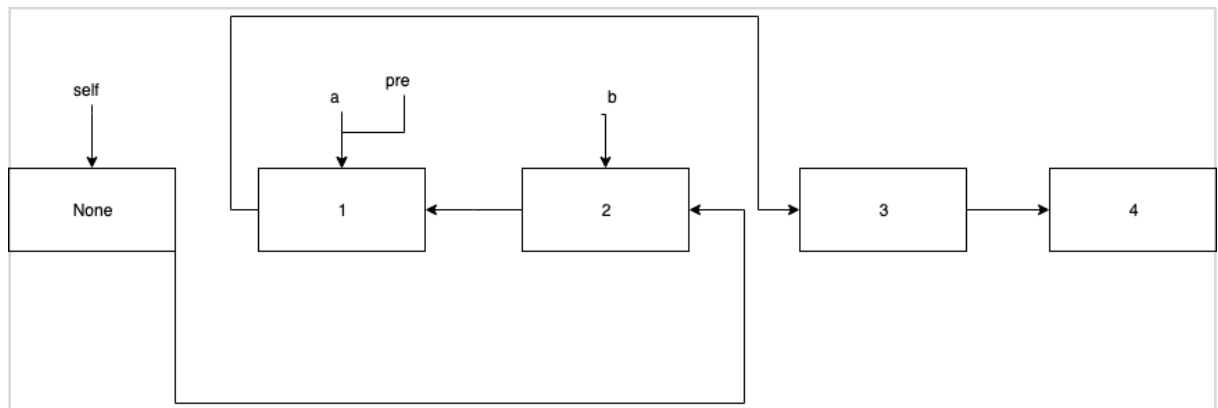
第一次循环：

将pre的后继指针指向b指针的节点，即value为2的节点

将b的后继指针指向a指针的节点，即value为1的节点

将a的后继指针指向b指针的后继指针所指节点，即value为3的节点

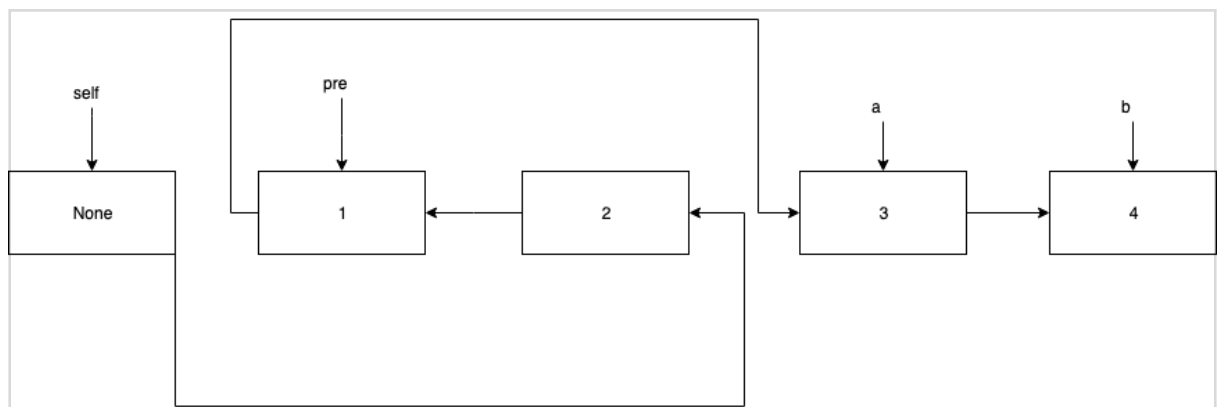
pre指针赋值为a指针所在位置



第一次循环：

a指向pre的后继节点，即value为3的节点

b指向a指针的后继节点，即value为4的节点

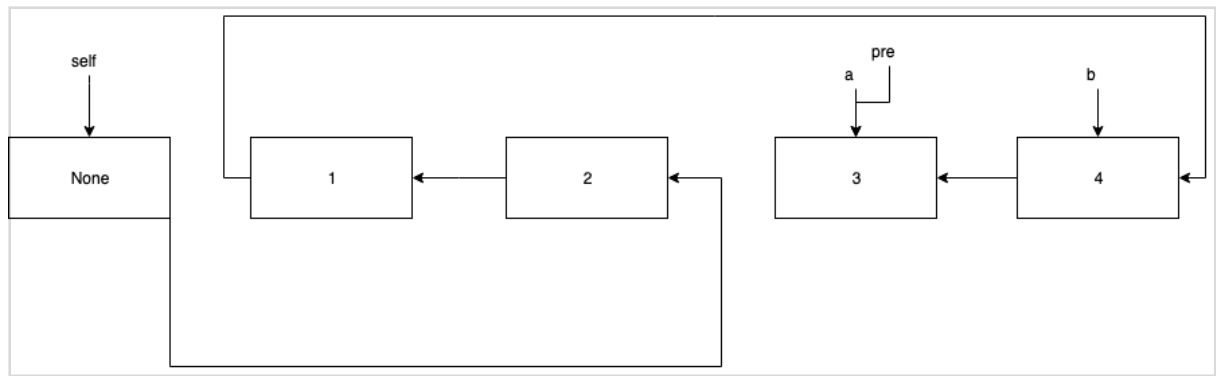


将pre的后继指针指向b指针的节点，即value为4的节点

将b的后继指针指向a指针的节点，即value为3的节点

将a的后继指针指向b指针的后继指针所指节点，即尾节点None

pre指针赋值为a指针所在位置



#algorithm#