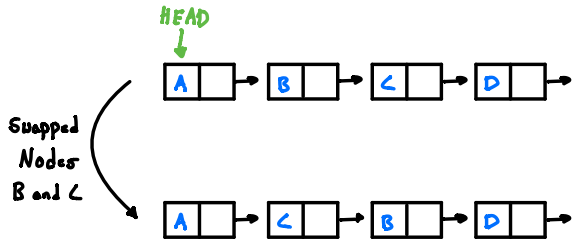
Singly Linked Lists- Node Swap

Node Swap. Two Cases: (Assume data entries are unique)

1. Node 1 and Node 2 are not head Nodes
2. Either Node 1 or Node 2 are head Nodes



Swap Nodes Function

```
def swap_nodes(self, key_1, key_2):

    if key_1 == key_2:
        return

    previous_1 = None
    current_1 = self.head

    while current_1  and current_1 .data != key_1:
        previous_1 = current_1
        current_1  = current_1.next

    previous_2 = None
    current_2 = self.head

    while current_2 and current_2.data != key_2:
        previous_2 = current_2
        current_2 = current_2.next

    if not current_1 or not current_2:
        return

    if  previous_1:
         previous_1.next = current_2
    else:
        self.head = current_2

    if previous_2:
       previous_2.next = current_1
    else:
        self.head = current_1

    current_1.next, current_2.next = current_2.next, current_1.next
```

if Given same node to swap, return

Search for Node 1

loop through linked list while keeping track of the current node and previous node.

For both nodes we like to swap

Search for Node 2

node doesnt exist

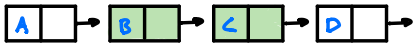
On next Page

Case 1: Node 1 and Node 2 are not head Nodes

```
1 if previous_1:
      previous_1.next = current_2
  else:
      self.head = current_2

2 if previous_2:
      previous_2.next = current_1
  else:
      self.head = current_1

3 current_1.next,current_2.next = current_2.next, current_1.next
```

We check if the previous node has a node. If True

If it has a node it essentially telling us it's not a head node.

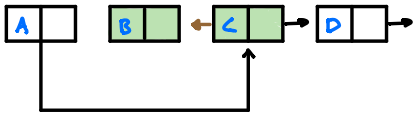


1

previous_1.next = Current_2 ~ Change Pointer A points to C

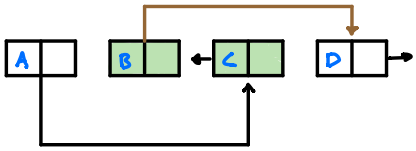
.next : Where node is pointing to

2



previous_2.next = Current_1 ~ Change Pointer, C points to B

3



~ Swapping the next nodes of node 1 and node 2 (B and C).

Current_1.next, Current_2.next = Current_2.next, Current_1.next

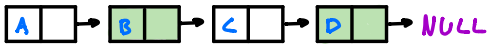
Can also be Coded like this:

```
temp = current_1.next
current_1.next = current_2.next    point to D
current_2.next = temp
```

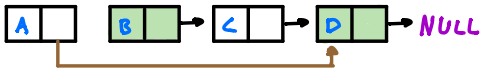
Swap Nodes B and D
(another case 1 example)

While loop kept track of our previous and current node

```
1  if previous_1:
       previous_1.next = current_2
   else:
       self.head = current_2

2  if previous_2:
       previous_2.next = current_1
   else:
       self.head = current_1

3  current_1.next,current_2.next = current_2.next,  current_1.next
```

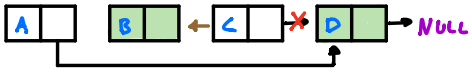


1



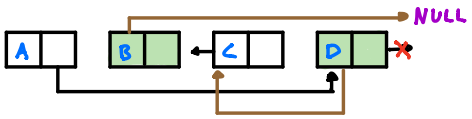
previous_1.next = current_2

2



previous_2.next = current_1

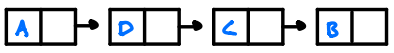
3



current_1.next, current_2.next = current_2.next, current_1.next

B → NULL

D → C

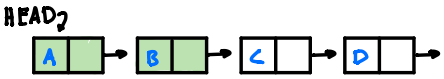


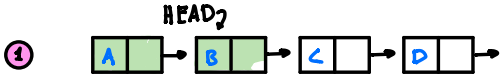
- Swapped

Case 2: Either Node 1 or Node 2 are head Nodes

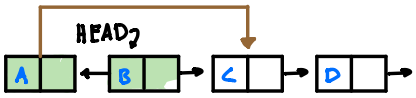
```
if  previous_1:
     previous_1.next = current_2
else:
    self.head = current_2

if previous_2:
   previous_2.next = current_1
else:
    self.head = current_1

current_1.next,current_2.next = current_2.next,  current_1.next
```

Variable Explorer: prev_1.data = None, Curr_1.data = A , prev_2.data = A , Curr_2.data = B



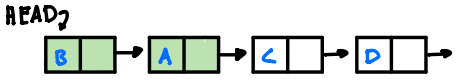


Self.head = current_2



prev_2.next = curr_1
prev_2.next = A

Curr_1.next, curr_2.next = Curr_2.next, curr_1.next



Swapped