

LINKED LIST

PROBLEM SET

- 21. Merge Two Sorted Lists
- 24. Swap Nodes in Pairs
- 83. Remove Duplicates from Sorted List
- 82. Remove Duplicates from Sorted List II
- 206. Reverse Linked List
- 92. Reverse Linked List II
- 234. Palindrome Linked List
- 237. Delete Node in a Linked List
- 203. Remove Linked List Elements
- 160. Intersection of Two Linked Lists
- 2. Add Two Numbers
- 61. Rotate List
- 445. Add Two Numbers II

NO. 21

MERGE TWO SORTED LISTS

PROBLEM DESCRIPTION

- Merge two sorted linked lists and return it as a new list. The new list should be made by splicing together the nodes of the first two lists

IDEA

- Create dummy node because we are not sure whose head is smaller

```
while list1_head and list2_head:  
    if list1_head < list2_head:  
        dummy.next = list1_head  
        list1_head = list1_head.next  
    else:  
        // similar to above case
```

- Because the length of list1 and list2 may not be the same, we must handle the tailing problem

```
cur.next = the_rest_of_list1 or the_rest_of_list2
```

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/21_Merge_Two_Sorted_Lists

NO. 24

SWAP NODES IN PAIRS

PROBLEM 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

IDEA

- Fetch next three nodes at a time (cur is dummy)



- `cur.next = next2`
- `next1.next = next3`
- `next2.next = next1`
- `cur = next1`

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/24_Swap_Nodes_in_Pairs

NO. 83

REMOVE
DUPLICATES FROM
SORTED LIST

PROBLEM DESCRIPTION

- Given a sorted linked list, delete all duplicates such that each element appear only once.
- For example,
- Given 1->1->2, return 1->2.
- Given 1->1->2->3->3, return 1->2->3.

IDEA

- Skip current node
 - `prev.next = cur.next`
- Use `set()` to tracking input node is duplicated or not
 - if yes, skip it
 - if no, add to `set()`

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/83_Remove_Duplicates_from_Sorted_List

NO. 83

82 REMOVE
DUPLICATES FROM
SORTED LIST II

PROBLEM DESCRIPTION

- Given a sorted linked list, delete all nodes that have duplicate numbers, leaving only distinct numbers from the original list.
- For example,
- Given 1->2->3->3->4->4->5, return 1->2->5.
- Given 1->1->1->2->3, return 2->3.

IDEA

- Similar to No.83 except
 - Need to consider the next node
 - If `cur.val` not in `setBook` and `cur.next` and `cur.val != cur.next.val`, then it's a valid node
 - If `cur.next` is `None`, then it's the last node
 - Further check if it is valid, if yes, add it. Otherwise, make `prev.next = None`
 - If we don't meet the above cases, the input node must be invalid

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/82_Remove_Duplicates_from_Sorted_List_II

NO. 206

REVERSE LINKED LIST

PROBLEM DESCRIPTION

- Reverse a singly linked list

IDEA

- Need to track the previous node

- Template

```
prev, cur = None, head
while cur:
    tmp = cur.next
    cur.next = prev
    prev = cur
    cur = tmp
return prev
```

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/206_Reverse_Linked_List

NO. 92

REVERSE
LINKED LIST II

PROBLEM DESCRIPTION

- Reverse a linked list from position m to n. Do it in-place and in one-pass
- For example
- Given 1->2->3->4->5->NULL, m = 2 and n = 4
- return 1->4->3->2->5->NULL

IDEA

- Find offset
- Reverse selected range
- Combine range

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/92_Reverse_Linked_List_II

NO. 234

PALINDROME
LINKED LIST

PROBLEM DESCRIPTION

- Given a singly linked list, determine if it is a palindrome.

IDEA

- Find mid point and separate list to two parts
- Reverse the second part
- Identify whether the first part is the same as the second part

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/234_Palindrome_Linked_List

NO. 237

DELETE NODE IN A LINKED LIST

PROBLEM DESCRIPTION

- Write a function to delete a node (except the tail) in a singly linked list, given only access to that node.

IDEA

- How to remove a certain node?
 - Make its previous node point to its next node
- What if we can only access the given node?
 - Copy the value of its next node to its own
 - Skip its next node

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/237_Delete_Node_in_a_Linked_List

NO. 203

REMOVE LINKED LIST ELEMENTS

PROBLEM DESCRIPTION

- Remove all elements from a linked list of integers that have value val

IDEA

- Use dummy node in case the target is the head
- Found target, skip it
 - `cursor.next = cur.next`

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/203_Remove_Linked_List_Elements

NO. 160

INTERSECTION OF TWO LINKED LISTS

PROBLEM DESCRIPTION

- Write a program to find the node at which the intersection of two singly linked lists begins.

IDEA

- Make list1 and list2 move forward
- Keep visited nodes to set()
- If the input node is already existed in set(), we found the answer

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/160_Intersection_of_Two_Linked_Lists

NO. 2

ADD TWO NUMBERS

PROBLEM DESCRIPTION

- You are given two non-empty linked lists representing two non-negative integers. The digits are stored in reverse order and each of their nodes contain a single digit. Add the two numbers and return it as a linked list.
- Example
 - Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)
 - Output: 7 -> 0 -> 8

IDEA

- Input order to low to high

if l1:

total += l1.val

l1 = l1.next

if l2:

total += l2.val

l2 = l2.next

current = (total + carry) % 10

carry = (total + carry) / 10

- Add calculated digit to the tail

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/2_Add_Two_Numbers

NO. 445

ADD TWO
NUMBERS II

PROBLEM DESCRIPTION

- You are given two non-empty linked lists representing two non-negative integers. The most significant digit comes first and each of their nodes contain a single digit. Add the two numbers and return it as a linked list.
- Example:
 - Input: (7 -> 2 -> 4 -> 3) + (5 -> 6 -> 4)
 - Output: 7 -> 8 -> 0 -> 7

IDEA

- Input order is from high to low
- Use queue to save the digits

```
total = 0
if queue1: total += queue1.pop()
if queue2: total += queue2.pop()
current = (carry + total) % 10
carry = (carry + total) / 10
```

- Add calculated digit to the front

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/445_Add_Two_Numbers_II

NO. 61

ROTATE LIST

PROBLEM DESCRIPTION

- Given a list, rotate the list to the right by k places, where k is non-negative.
- For example:
 - Given 1->2->3->4->5->NULL and k = 2,
 - return 4->5->1->2->3->NULL.

IDEA

- Calculate list size
- If it is a complete circle, simply return original head
- Find the tailing node and break its next relation

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/61_Rotate_List