Leetcode Bootcamp Homework - 3

1. Palindrome Linked List

```
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next

class Solution:
    def isPalindrome(self, head):
        def revlist(node):
            prev = None
            curr = node
            while curr:
                next_node = curr.next
                curr.next = prev
                prev = curr
                curr = next_node
            return prev

        one= head
        two = head

        while two and two.next:
            one= one.next
            two= two.next.next


        start_of_second_half = one
        if two:
             start_of_second_half = one.next

        reversed_second_half = revlist(start_of_second_half)

        p1 = head
        p2 = reversed_second_half
        is_palindrome = True

        while p2:
            if p1.val != p2.val:
                is_palindrome = False
                break
            p1 = p1.next
            p2 = p2.next

        return is_palindrome
```

2. Reorder List

```
# Definition for singly-linked list.
# class ListNode(object):
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution(object):
    def reorderList(self, head):
        """
        :type head: Optional[ListNode]
        :rtype: None Do not return anything, modify head in-place
instead.
        """
        if not head or not head.next:
            return

        slow, fast = head, head.next
        while fast and fast.next:
            slow = slow.next
            fast = fast.next.next

        firstHalf_head = head
        secondHalf_head = slow.next

        slow.next = None

        prev = None
        curr = secondHalf_head
        while curr:
            next_node = curr.next
            curr.next = prev
            prev = curr
            curr = next_node

        reversed_secondHalf = prev
        p1 = firstHalf_head
        p2 = reversed_secondHalf

        while p2:
            temp1 = p1.next
            temp2 = p2.next
            p1.next = p2
            p2.next = temp1

            p1 = temp1
            p2 = temp2
```

3. Set Matrix Zeros

```
class Solution(object):
    def setZeroes(self, matrix):
        """
        :type matrix: List[List[int]]
        :rtype: None Do not return anything, modify matrix in-place
instead.
        """
        m = len(matrix)
        n = len(matrix[0])

        first_col_0 = False

        for i in range(m):
            if matrix[i][0] == 0:
                first_col_0 = True

            for j in range(1, n):
                if matrix[i][j] == 0:
                    matrix[i][0] = 0
                    matrix[0][j] = 0
        for i in range(1, m):
            for j in range(1, n):
                if matrix[i][0] == 0 or matrix[0][j] == 0:
                    matrix[i][j] = 0

        if matrix[0][0] == 0:
            for j in range(n):
                matrix[0][j] = 0
```

```
if first_col_0:
    for i in range(m):
        matrix[i][0] = 0
```