2020-07-18 - Handout – Linked Lists

Q1. Reverse Linked List ii

Link: https://leetcode.com/problems/reverse-linked-list-ii/

Reverse a linked list from position m to n. Do it in one-pass.

Note: $1 \le m \le n \le$ length of list.

```
Example:
Input: 1->2->3->4->5->NULL, m = 2, n = 4
Output: 1->4->3->2->5->NULL
```

Q2. Reorder List

Link: https://leetcode.com/problems/reorder-list/

Given a singly linked list L: $L_0 \to L_1 \to \ldots \to L_{n-1} \to L_n$,

reorder it to: $L_0 \to L_n \to L_1 \to L_{n-1} \to L_2 \to L_{n-2} \to \ldots$

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

```
Example 1:
Given 1->2->3->4, reorder it to 1->4->2->3.
Example 2:
Given 1->2->3->4->5, reorder it to 1->5->2->4->3.
```

Q3. Linked List Cycle ii

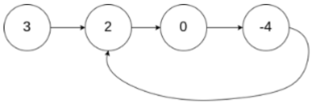
Link: https://leetcode.com/problems/linked-list-cycle-ii/

Given a linked list, return the node where the cycle begins. If there is no cycle, return null.

To represent a cycle in the given linked list, we use an integer pos which represents the position (0-indexed) in the linked list where tail connects to. If pos is -1, then there is no cycle in the linked list.

Note: Do not modify the linked list.

```
Example 1:
Input: head = [3,2,0,-4], pos = 1
Output: tail connects to node index 1
Explanation: There is a cycle in the linked list, where tail connects to the second node.
```

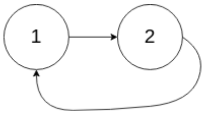


Example 2:

Input: head = [1,2], pos = 0

Output: tail connects to node index 0

Explanation: There is a cycle in the linked list, where tail connects to the first node.



Example 3:

Input: head = [1], pos = -1

Output: no cycle

Explanation: There is no cycle in the linked list.