Notebook

August 2, 2024

[ ]:
```python
# https://leetcode.com/problems/copy-list-with-random-pointer/description/

# use a hashmap to store the mapping of old node to new node
class Solution:
    def copyRandomList(self, head: 'Optional[Node]') -> 'Optional[Node]':
        if not head:
            return None
        old_to_new = {}

        curr = head
        while curr:
            old_to_new[curr] = Node(curr.val)
            curr = curr.next

        curr = head
        while curr:
            old_to_new[curr].next = old_to_new.get(curr.next)
            old_to_new[curr].random = old_to_new.get(curr.random)
            curr = curr.next

        return old_to_new[head]

# Interweaving

class Solution:
    def copyRandomList(self, head: 'Optional[Node]') -> 'Optional[Node]':
        if not head:
            return None

        curr = head
        while curr:
            new_node = Node(curr.val, curr.next)
            curr.next = new_node
            curr = new_node.next

        curr = head
        while curr:
```

```
            if curr.random:
                curr.next.random = curr.random.next
            curr = curr.next.next

        old_head = head
        new_head = head.next
        curr_old = old_head
        curr_new = new_head

        while curr_old:
            curr_old.next = curr_old.next.next
            curr_new.next = curr_new.next.next if curr_new.next else None
            curr_old = curr_old.next
            curr_new = curr_new.next

        return new_head
```

[ ]: