Copy Linked List with Random Pointer LeetCode

Given a linked list with next and random pointers, create a deep copy of the list.

Approach 1: Copy the linked list with random pointers using a hash map

- Create a new list with the same values as the original list.
- While creating the new list, maintain a mapping between original nodes and their corresponding clone nodes using a hash map.
- Traverse both original and clone lists, and set random pointers for clone nodes based on the hash map.
- **Time Complexity: O(N), where N is the length of the linked list. We traverse the list twice and perform constant time operations.**
- **Space Complexity: O(N), as we use a hash map to store the mapping of original and clone nodes.**

Approach 2: Copy the linked list with random pointers (optimized approach)

- Step 1: Create a new list with the same values as the original list. Add the clone nodes right after the original nodes.
- Step 2: Traverse both original and clone lists. Set random pointers for clone nodes using original nodes' random pointers.
- Step 3: Traverse the combined list, adjust the random pointers of cloned nodes based on original nodes.
- Step 4: Separate the original and cloned lists, restoring their original structure.
- **Time Complexity: O(N), where N is the length of the linked list. We traverse the list three times and perform constant time operations.**
- **Space Complexity: O(1), as we perform the copy in-place without using extra space.**