0138. 复制带随机指针的链表

ITCharge 大约 1 分钟

- 标签：哈希表、链表
- 难度：中等

题目链接

- 0138. 复制带随机指针的链表 - 力扣

题目大意

描述：给定一个链表的头节点 `head` ，链表中每个节点除了 `next` 指针之外，还包含一个随机指针 `random` ，该指针可以指向链表中的任何节点或者空节点。

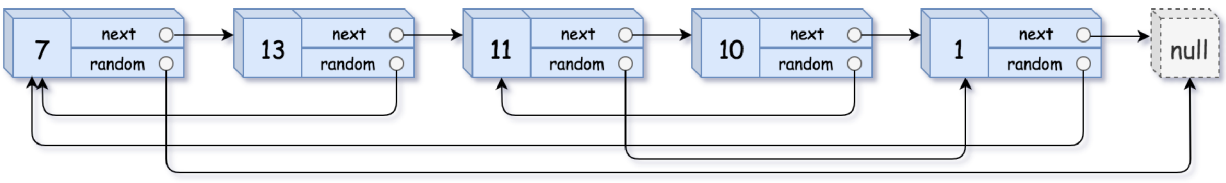
要求：将该链表进行深拷贝。返回复制链表的头节点。

说明：

- $0 \leq n \leq 1000$。
- $-10^4 \leq Node.val \leq 10^4$。
- `Node.random` 为 `null` 或指向链表中的节点。

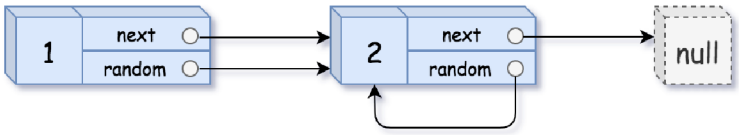
示例：

- 示例 1：




```py
输入：head = [[7,null],[13,0],[11,4],[10,2],[1,0]]
输出：[[7,null],[13,0],[11,4],[10,2],[1,0]]
```

- 示例 2：




```py
输入：head = [[1,1],[2,1]]
输出：[[1,1],[2,1]]
```

解题思路

思路 1：迭代

1. 遍历链表，利用哈希表，以 `旧节点：新节点` 为映射关系，将节点关系存储下来。
2. 再次遍历链表，将新链表的 `next` 和 `random` 指针设置好。

思路 1：代码

```py
class Solution:
    def copyRandomList(self, head: 'Node') -> 'Node':
        if not head:
            return None
        node_dict = dict()
        curr = head
        while curr:
            new_node = Node(curr.val, None, None)
            node_dict[curr] = new_node
            curr = curr.next
        curr = head
        while curr:
            if curr.next:
                node_dict[curr].next = node_dict[curr.next]
            if curr.random:
                node_dict[curr].random = node_dict[curr.random]
            curr = curr.next
        return node_dict[head]
```

思路 1：复杂度分析

- **时间复杂度**：$O(n)$。
- **空间复杂度**：$O(n)$。