



链滴

【剑指 offer】复杂链表的复制

作者: [limanting](#)

原文链接: <https://ld246.com/article/1630034608122>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

题目

请实现 copyRandomList 函数，复制一个复杂链表。在复杂链表中，每个节点除了有一个 next 指针向下一个节点，还有一个 random 指针指向链表中的任意节点或者 null。

来源：力扣（LeetCode）

链接：<https://leetcode-cn.com/problems/fu-za-lian-biao-de-fu-zhi-lcof>

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

我的答案

```
/**
 * // Definition for a Node.
 * function Node(val, next, random) {
 *   this.val = val;
 *   this.next = next;
 *   this.random = random;
 * };
 */

/**
 * @param {Node} head
 * @return {Node}
 */
var copyRandomList = function (head) {
  const nodeMap = new Map();
  let currentNode = head;
  while (currentNode) {
    nodeMap.set(currentNode, new Node(currentNode.val, null, null));
    currentNode = currentNode.next;
  }
  currentNode = head;
  while (currentNode) {
    const nodeCopy = nodeMap.get(currentNode);
    nodeCopy.next = nodeMap.get(currentNode.next) || null;
    nodeCopy.random = nodeMap.get(currentNode.random) || null;
    currentNode = currentNode.next;
  }

  return nodeMap.get(head);
};
```

这个题目是需要把节点的关系进行复制，但是节点都是新的节点。最容易想到的办法就是，用一个map，将新的节点和旧的节点映射起来，最后再根据旧链表的关系，将新的节点联系起来，两次循环。但其这里做了一次冗余的

优化

首先是内存上的优化。如何不使用多余的存储完成复制？这里可以利用人为设置的规定，我们将新的点建立在被复制的节点后面，最后根据新的节点是旧节点的next进行关系的设置，设置完之后，将旧

点从链表中删除。

这里需要三次循环，第一次将新节点插入在被复制节点的后面，第二次将radom设置，第三次删除旧点。