LeetCode 第 138 号问题：复制带随机指针的链表

> *本文首发于公众号「图解面试算法」，是 图解 LeetCode 系列文章之一。*
>
> *同步博客：https://www.algomooc.com*

题目来源于 LeetCode 上第 138 号问题：复制带随机指针的链表。题目难度为 Medium，目前通过率为 40.5% 。

题目描述

给定一个链表，每个节点包含一个额外增加的随机指针，该指针可以指向链表中的任何节点或空节点。

要求返回这个链表的**深拷贝。**

示例：

```
输入：
{"$id":"1","next":{"$id":"2","next":null,"random":{"$ref":"2"},"val":2},"random":
{"$ref":"2"},"val":1}

解释：
节点 1 的值是 1，它的下一个指针和随机指针都指向节点 2 。
节点 2 的值是 2，它的下一个指针指向 null，随机指针指向它自己。
```

题目解析

1. 在原链表的每个节点后面拷贝出一个新的节点
2. 依次给新的节点的随机指针赋值，而且这个赋值非常容易 cur->next->random = cur->random->next
3. 断开链表可得到深度拷贝后的新链表

之所以说这个方法比较巧妙是因为相较于一般的解法（如使用 Hash map ）来处理，上面这个解法 **不需要占用额外的空间。**

动画描述

代码实现

我发现带指针的题目使用 C++ 版本更容易描述，所以下面的代码实现是 C++ 版本。

```
class Solution {
public:
    RandomListNode *copyRandomList(RandomListNode *head) {
        if (!head) return NULL;
        RandomListNode *cur = head;
        while (cur) {
            RandomListNode *node = new RandomListNode(cur->label);
            node->next = cur->next;
            cur->next = node;
            cur = node->next;
```

```
        }
        cur = head;
        while (cur) {
            if (cur->random) {
                cur->next->random = cur->random->next;
            }
            cur = cur->next->next;
        }
        cur = head;
        RandomListNode *res = head->next;
        while (cur) {
            RandomListNode *tmp = cur->next;
            cur->next = tmp->next;
            if(tmp->next) tmp->next = tmp->next->next;
            cur = cur->next;
        }
        return res;
    }
};
```