链滴

复杂链表的复制

作者：yudake
原文链接：https://ld246.com/article/1518855064087
来源网站：链滴

题目描述

输入一个复杂链表（每个节点中有节点值，以及两个指针，一个指向下一个节点，另一个特殊指针指任意一个节点），返回结果为复制后复杂链表的head。（注意，输出结果中请不要返回参数中的节引用，否则判题程序会直接返回空）

解题思路

- 用map建立新结点和旧结点的联系；
  - 时间复杂度O(2n)，空间复杂度O(n)
- 将新结点new插入到旧结点old之后；
  - 遍历链表，创建并插入新结点，old->new；
  - 遍历链表，new.random = old.random.next；
  - 将链表拆分，奇数位结点就是旧结点old.next = old.next.next，偶数位结点就是新结点new.next = new.next.next。
  - 空间复杂度O(3n)，空间复杂度O(1)；

代码

代码一：

```
import java.util.HashMap;
public class Solution {
    public RandomListNode Clone(RandomListNode pHead) {
        if (pHead == null)
            return pHead;
        HashMap<RandomListNode, RandomListNode> map = new HashMap<>();
        for (RandomListNode p = pHead; p != null; p = p.next) {
            map.put(p, new RandomListNode(p.label));
        }
        for (RandomListNode p = pHead; p != null; p = p.next) {
            map.get(p).next = map.get(p.next);
            map.get(p).random = map.get(p.random);
        }
        return map.get(pHead);
    }
}
```

代码二（有点丑）：

```
public class Solution {
    public RandomListNode Clone(RandomListNode pHead) {
        if (pHead == null)
            return pHead;
        RandomListNode node = pHead;
        while (node != null) {
            RandomListNode newNode = new RandomListNode(node.label);
            newNode.next = node.next;
```

```
            node.next = newNode;
            node = newNode.next;
        }
        node = pHead;
        RandomListNode newNode = node.next;
        while (node != null) {
            if (node.random != null)
                newNode.random = node.random.next;
            node = newNode.next;
            if (node != null)
                newNode = node.next;
        }
        node = pHead;
        newNode = node.next;
        RandomListNode newHead = node.next;
        while (node != null) {
            node.next = newNode.next;
            node = node.next;
            if (node != null) {
                newNode.next = node.next;
                newNode = newNode.next;
            }
        }
        return newHead;
    }
}
```

原文链接：复杂链表的复制