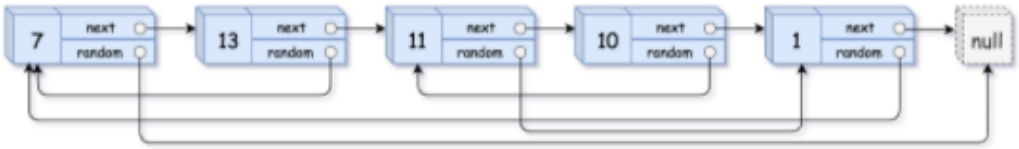
复杂链表的复制

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 指针指向下一个节点，还有一个 random 指针指向链表中的任意节点或者 null。

示例 1：



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]]

```
/*
// Definition for a Node.
class Node {
public:
    int val;
    Node* next;
    Node* random;

    Node(int _val) {
        val = _val;
        next = NULL;
        random = NULL;
    }
};
*/
class Solution {
public:
    Node* copyRandomList(Node* head) {
        if(head==nullptr)
            return nullptr;
        struct Node* cur=head;
        while(cur)
        {
            //(1)拷贝数据
            struct Node* newnode=(struct Node*)malloc(sizeof(struct Node));
            newnode->val=cur->val;
            newnode->next=cur->next;
```

```
            cur->next=newnode;
            //更新 cur
            cur=newnode->next;
        }
        //(2)拷贝随机指针
        cur=head;
        while(cur)
        {
            struct Node* copy=cur->next;
            if(cur->random)
            {
                copy->random=cur->random->next;
            }
            else
                copy->random=NULL;
            cur=copy->next;//连续两个重复 每次后移两个
        }
        //(3)链表拆分
        cur=head;
        struct Node* newhead=cur->next;
        while(cur)
        {
            struct Node* copy=cur->next;
            struct Node* next=copy->next;
            cur->next=next;
            if(next)
                copy->next=next->next;
            cur=next;
        }
        return newhead;
    }
};
```