一、题目说明

题目138. Copy List with Random Pointer，一个链表中的节点包括一个random指针，可以指向链表中的任何节点或者空，生成该链表的一个拷贝。难度是Medium！

二、我的解答

这个题目是赋值一个链表，难度在于random链表的指向。这里用计数方式实现：

```
class Solution{
    public:
        Node* copyRandomList(Node* head){
            if(head == NULL) return NULL;

            Node* p = head->next,*newHead,*newP,*cur;
            newHead = new Node(head->val);
            newP = newHead;


            while(p!=NULL){
                newP->next = new Node(p->val);
                newP = newP->next;
                p = p->next;
            }

            p = head;
            newP = newHead;
            int n = 0;
            while(p!=NULL){
                cur = p->random;
                if(cur!=NULL){
                    //从head开始遍历，数第几个
                    Node* p1 = head;
                    while(p1 !=cur){
                        p1 = p1->next;
                        n++;
                    }

                    Node*p2 = newHead;
                    while(n>0){
                        p2 = p2->next;
                        n--;
                    }
                    newP->random = p2;
                }

                p = p->next;
                newP = newP->next;
            }

            return newHead;
        }
};
```

性能如下：

```
Runtime: 12 ms, faster than 61.45% of C++ online submissions for Copy List with
Random Pointer.
Memory Usage: 13.4 MB, less than 100.00% of C++ online submissions for Copy List
with Random Pointer.
```

三、优化措施

由于random指针用的是“从头到尾”计数的方式实现，性能一般。可以用Hash方式加速：

```
class Solution{
    public:
        Node* copyRandomList(Node* head){
            if(head == NULL) return NULL;
            Node* cur = head;
            unordered_map<Node*,Node*> ump;
            //拷贝节点，复制val
            while(cur!=NULL){
                Node* copy = new Node(cur->val);
                ump[cur] = copy;
                cur = cur->next;
            }
            //复制next和random
            cur = head;
            while(cur!=NULL){
                ump[cur]->next = ump[cur->next];
                ump[cur]->random = ump[cur->random];
                cur = cur->next;
            }

            return ump[head];
        }
};
```

性能如下：

```
Runtime: 8 ms, faster than 89.21% of C++ online submissions for Copy List with
Random Pointer.
Memory Usage: 13.5 MB, less than 100.00% of C++ online submissions for Copy List
with Random Pointer.
```