一、题目说明

题目146. LRU Cache，设计并实现一个LRU Cache，支持get和put操作。难度是Medium！时间复杂度要求是O(1)。

二、我的解答

时间复杂度要求是O(1)，只能通过hash实现。同时要维护一个容量capacity，当capacity满的时候，更新“最近最少使用的元素”。故需要Hash+LinkedList实现“哈希链表”。

```
class LRUCache {
    public:
        struct Node{
            int key,value;
            Node* next,*pre;
        };
        Node* head,*rear;

        LRUCache(int size){
            capacity = size;
            head = new Node();
            rear = new Node();
            head->pre = NULL;
            head->next = rear;
            rear->next = NULL;
            rear->pre = head;
        }
        int get(int key){
            if(cache.find(key)==cache.end()) return -1;
            Node* tmp = cache[key];

            //移除该节点
            tmp->pre->next = tmp->next;
            tmp->next->pre = tmp->pre;

            //插入链头
            head->next->pre = tmp;
            tmp->next = head->next;
            tmp->pre = head;
            head->next = tmp;
            return tmp->value;
        }
        void lru_delete() {
            if(cache.size() == 0) return;
            Node* tmp = rear->pre;
            rear->pre = tmp->pre;
            tmp->pre->next = rear;
            cache.erase(tmp->key);
            delete tmp;
        }
        void put(int key,int value){
            if(cache.find(key) != cache.end()) {
                //key已存在于链表中，更新值
                cache[key]->value = value;
                this->get(key);
                return;
```

```
            }

            //插入链表中
            if(cache.size() >= capacity)
                this->lru_delete();
            Node *tmp = new Node;
            tmp->key = key;
            tmp->value = value;
            tmp->pre = this->head;
            tmp->next = this->head->next;
            if(head->next != NULL) head->next->pre = tmp;
            this->head->next = tmp;
            cache.insert(pair<int, Node*>(key, tmp));
        }
    private:
        map<int,Node*> cache;
        //最大容量
        int capacity;
};
```

```
Runtime: 120 ms, faster than 46.13% of C++ online submissions for LRU Cache.
Memory Usage: 38.1 MB, less than 74.39% of C++ online submissions for LRU Cache.
```

三、优化措施

无