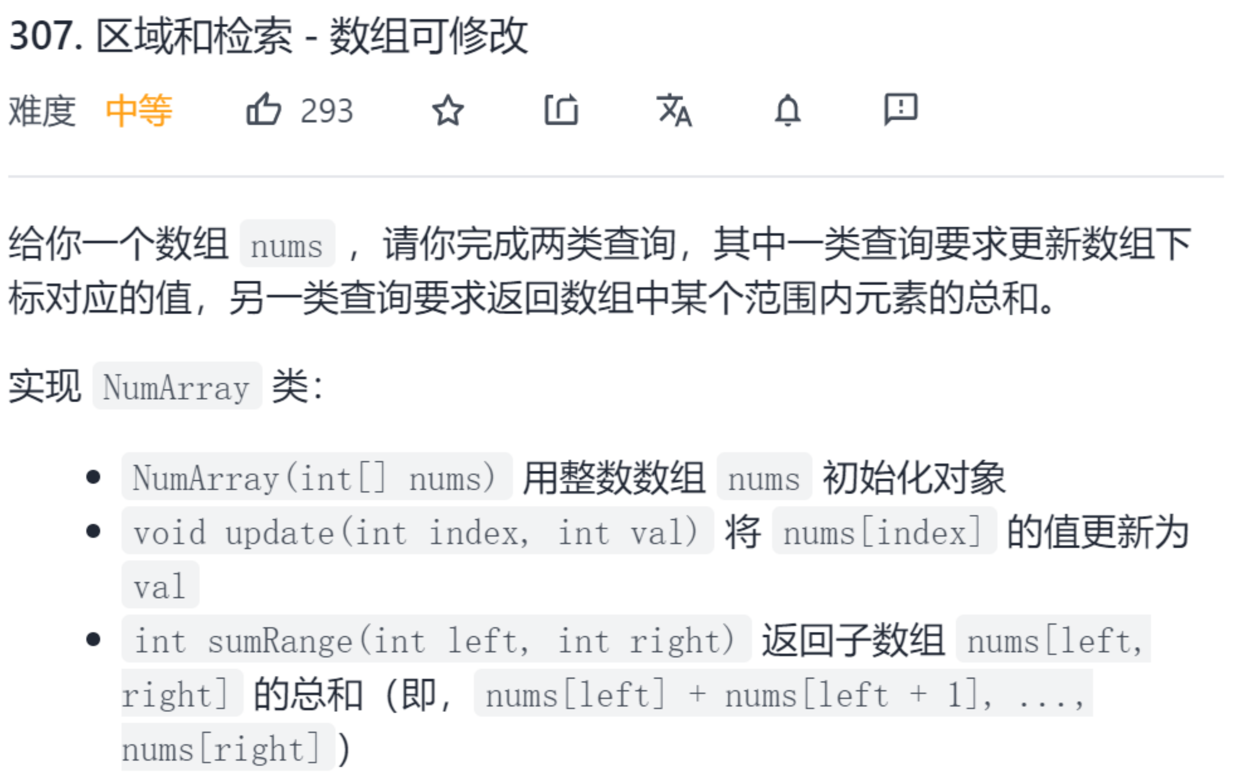
数据结构：线段树

题目描述：



307. 区域和检索 - 数组可修改

难度 中等 293

给你一个数组 `nums` ，请你完成两类查询，其中一类查询要求更新数组下标对应的值，另一类查询要求返回数组中某个范围内元素的总和。

实现 `NumArray` 类：

- `NumArray(int[] nums)` 用整数数组 `nums` 初始化对象
- `void update(int index, int val)` 将 `nums[index]` 的值更新为 `val`
- `int sumRange(int left, int right)` 返回子数组 `nums[left, right]` 的总和（即， `nums[left] + nums[left + 1], ...,` `nums[right]` ）

题目分析：

- 前缀和暴力修改超时
- 线段树加速

```
class NumArray {
private:
//线段树：修改点，查询区间
vector<int>tree;
vector<int>datas;
public:
    NumArray(vector<int>& nums) {
        datas=nums;
        int n=datas.size();
        tree.resize(4*n);
        buildtree(0,n-1,0);
    }
    //建树
    void buildtree(int l,int r,int tag){
        if(l==r){
            tree[tag]=datas[r];//datas[l];
            return;
        }
        int mid=(r+l)>>1;
        buildtree(l,mid,tag*2+1);
        buildtree(mid+1,r,tag*2+2);
        tree[tag]=tree[tag*2+1]+tree[tag*2+2];
    }
    //修改点
    void change_points(int l,int r,int tag,int index,int val){
        if(l==r){
            //此时index==l==r;
            tree[tag]=val;
            return;
        }
        int mid=(l+r)>>1;
        if(mid>=index)change_points(l,mid,2*tag+1,index,val);
        else change_points(mid+1,r,2*tag+2,index,val);
        //同时修改相关的和
        tree[tag]=tree[tag*2+1]+tree[tag*2+2];
    }
    void update(int index, int val) {
        change_points(0,datas.size()-1,0,index,val);
    }

    //寻找和
    int serch_sum(int left, int right, int l, int r, int tag) {
        if (left <= l && right >= r) {
            return tree[tag];
        }
        int mid = (r + l) >> 1;
        int res = 0;
        if (left <= mid)res += serch_sum(left, right, l, mid, 2 * tag +
1);
        if (right >= mid + 1)res += serch_sum(left, right, mid + 1, r, 2
* tag + 2);
        return res;
    }


    int sumRange(int left, int right) {
        int n=datas.size();
        return serch_sum(left,right,0,n-1,0);
    }
};

/**
```

```
 * Your NumArray object will be instantiated and called as such:
 * NumArray* obj = new NumArray(nums);
 * obj->update(index,val);
 * int param_2 = obj->sumRange(left,right);
 */
```