2.区间求和（差线段树）

整体

- 数组不变，区间查询：前缀和、树状数组、线段树；
- 数组单点修改，区间查询：树状数组、线段树；
- 数组区间修改，单点查询：差分、线段树；
- 数组区间修改，区间查询：线段树

1 前缀和

- 下标从1开始，处理边界问题
- 从0开始的问题，查询0的时候-1越界

一维前缀和

```
sum[i] = sum[i-1] + a[i]

//数组从0开始
//sum从1开始
vector<int> leftSum(n + 1, 0);
vector<int> rightSum(n + 1, 0);
for(int i = 0; i < n ; i++){
    leftSum[i + 1] = leftSum[i] + nums[i];
}

for(int i = n - 1; i >= 0; i--){
    rightSum[i] = rightSum[i + 1] + nums[i];
}
```

二维前缀和

```
vector<vector<int>>(row + 1, vector<int>(col + 1, 0));
for(int i = 1; i <= row; i++){
    for(int j = 1; j <= col; j++){
        sum[i][j] = sum[i-1][j] + sum[i][j-1] - sum[i-1][j-1] + matrix[i - 1][j
    }
}
```

求一段区间内的和

```
sum[r] - sum[l - 1]

//sum数组从1开始
sum[r2 + 1][c2 + 1] -sum[r2 + 1][c1] - sum[r1][c2 + 1] + sum[r1][c1];
```

2 差分

一维差分

- 在区间[l,r]中让数列都增加C

```
//a是sum的差分数组
int sum[n] = {}; [1,2,2,4]
int a[n] = {}; [1,1,0,2]
int c;
void dif(){
    //求差分数组
    for(i=1;i<=n;i++){
        a[i]=sum[i]-sum[i-1];
    }
    //在区间[l,r]中让数列都增加C
    a[l] += c;
    a[r + 1] -= c;
    //答案
    for(i=1;i<=n;i++){
        //sum[i] = sum[i - 1] + a[i];
        a[i]+=a[i-1];
    }

}
```

题目

- 1109. 航班预订统计

二维差分

```
//a是sum的差分数组
int sum[n][m] = {};
int a[n][m] = {};
int c;
void dif(){
    for(int i = 1; i <= n; i++){
        for(int j = 1;j <= m; j++){
            cin >> sum[i];
            a[i][j]=sum[i][j]-sum[i-1][j]-sum[i][j-1] + sum[i-1][j-1];
        }
    }
    a[x1][y1] += c;
    a[x2+1][y1] -= c;
    a[x1][y2+1] -= c;
    a[x2+1][y2+1] += c;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= m; j++){
            s[i][j] = s[i-1][j] + s[i][j-1] - s[i-1][j-1] + a[i][j];
            //a[i][j] += a[i - 1][j] + a[i][j - 1] - a[i - 1][j - 1];  //二维前缀
        }
    }
}
```

3 树状数组

基本概念

- 讲解

功能

- 单点修改+区间查询
- 最简单的树状数组支持两种操作，时间复杂度均为 O(logn)

代码

- 下标从1开始，具体情况要转换

```
int lowbit(int x) {
    return x & -x;
}
//单点修改
int tree[MAXN];
void update(int index, int val)
{
    for (int i = index; i < MAXN; i += lowbit(i))
        tree[i] += x;
}

//求前n项和
inline int query(int index)
{
    int ans = 0;
    for (int i = index; i > 0; i -= lowbit(i))
        ans += tree[i];
    return ans;
}

//区间查询
inline int query(int a, int b)
{
    return query(b) - query(a - 1);
}
```

题目

- 307. 区域和检索 - 数组可修改
- 1310. 子数组异或查询

线段树

讲解

```
//创建
//查询
//修改
```