一维差分

给定一个数组a，应答若干次询问，每次询问给定三个数$l < r$和x，要求将$a[l], \ldots, a[r]$每个数都加上x。所有询问结束后，问a变成了什么。

考虑数组$d[i] = a[i] - a[i-1]$，规定$d[0] = 0$，容易看出，如果$a[l], \ldots, a[r]$每个数都加上x，操作之后的d和操作之前，只有$d[l]$和$d[r+1]$两个数发生了变化，由于$a[l]$变了而$a[l-1]$没变，所以$d[l]$加上了x；由于$a[r+1]$没变而$a[r]$变了，所以$d[r+1]$减了x。

从而对a的区间加x的操作反映在d里只是两次操作，可以$O(1)$的时间内实现。

考虑如何由d恢复a，容易看出a数组恰好是d的前缀和，从而对d求前缀和即得到a。

构造差分数组d的过程也可以这样想：

考虑每个数都是0的数组b，其差分数组d也全是0。我们将数组b经过若干次区间加数的操作变为数组a，那么对d进行相应的操作，d就也变成了a的差分数组。

显然，b数组可以通过依次进行将$b[i], 1 \le i \le n$加上$a[i]$，这n次操作，变为a，而将$b[i]$加上$a[i]$等价于将$d[i]$加$a[i]$，将$d[i+1]$减去$a[i]$。

例题

https://blog.csdn.net/qq_46105170/article/details/113794270

二维差分

类似一维差分，给定一个$n \times m$二维矩阵a，应答若干次询问，每次询问给定五个数x_1, y_1, x_2, y_2和c，要求将a中以x_1, y_1为左上角，x_2, y_2为右下角的子矩阵里，每个数都加上c。所有询问结束后，问a变成了什么。

同样道理，我们考虑某个矩阵d，使得d的前缀和数组恰好是a。

我们考虑如果对$d[i][j]$加上x，那么d的前缀和矩阵会怎么变。显然不包含(i, j)这个位置的前缀矩阵是不会变的，而包含的那些前缀矩阵的和都恰好加了x，也就是说，如果d的前缀和矩阵是p的话，以$p[i][j]$为左上角，$p[n][m]$为右下角的子矩阵里每个数都加了x。

而要将a中以x_1, y_1为左上角，x_2, y_2为右下角的子矩阵里，每个数都加上c，其等价于：

1. 将a中以x_1, y_1为左上角，n, m为右下角的子矩阵里，每个数都加上c

2. 将a中以x_2+1, y_1为左上角，n, m为右下角的子矩阵里，每个数都减去c
3. 将a中以x_1, y_2+1为左上角，n, m为右下角的子矩阵里，每个数都减去c
4. 将a中以x_2+1, y_2+1为左上角，n, m为右下角的子矩阵里，每个数都加上c

从而在d里对应的操作是：

1. 将$d[x_1][y_1]$加上c
2. 将$d[x_2+1][y_1]$减去c
3. 将$d[x_1][y_2+1]$减去c
4. 将$d[x_2+1][y_2+1]$加上c

可以常数时间操作。

求d的过程与一维类似，也有两种思路，一种是直接用a求，第二种是考虑从0矩阵变为a需要哪些操作，对d做相应操作。

例题

https://blog.csdn.net/qq_46105170/article/details/113794363

作业

用两种方法求差分数组，一维和二维

上面两道例题

https://blog.csdn.net/qq_46105170/article/details/119192784