树状数组维护区间加操作

主讲人：邓哲也

POJ 3468

维护一个长度为 n 的数组 A[1..n]

支持两种操作：

C a b c	将A[a..b]都加上c
Q a b	询问A[a..b]的和

1 <= n, Q <= 100000

POJ 3468

假设对 A[L..R] 整体都加了 d

我们可以运用差分的思想，新建一个数组 B

也就是 B[L] += d, B[R + 1] -= d

设原始的 A 数组是 C

这样 $A[i] = C[i] + B[1] + B[2] + \cdots + B[i]$

$= C[i] + Sum(B[1..i])$

POJ 3468

定义 S[i] = A[1] + … + A[i]

询问 A[L..R] 的和也就是 S[R] - S[L-1]

S[i] = (C[1] + B[1]) + (C[2] + B[1] + B[2]) + … + (C[i] + B[1] + B[2] + … B[i])

= Sum(C[1..i]) + B[1] * i + B[2] * (i - 1) + … + B[i]

= Sum(C[1..i]) + (i + 1) * Sum(B[1..i]) - (B[1] + 2B[2] + 3B[3] + … + iB[i])

POJ 3468

Sum(C[i]) 就是 A 数组的前缀和。

Sum(B[i]) 用一个树状数组维护即可。

对于每次[L,R]修改，我们只要修改

B[L] += d, B[R+1] -= d

Sum(B[i]*i) 再用一个树状数组维护即可。

对于每次[L,R]修改，我们只要修改

B[L] += L * d, B[R+1] -= (R+1) * d

POJ 3468

因此，用两个数组维护原序列的差分序列就可以支持区间加和区间求和了。

时间复杂度单次 $O(\log n)$

差分是一个重要的思想。

这个问题同样可以拓展到二维上，支持矩形加和矩形求和。

下节课再见