知识精炼（十）

主讲人：邓哲也

Codeforces 383C

有一棵树，对这个树有两种操作：

1 x val：在 x 号节点上加 val，然后给 x 的每个儿子加 -val，然后给 x 的每个儿子的每个儿子加 val，直到没有儿子位置。

2 x：查询 x 号节点上的值。

n, m <= 200000

input

```
5 5
1 2 1 1 2
1 2
1 3
2 4
2 5
1 2 3
1 1 2
2 1
2 2
2 4
```

output

```
3
3
0
```

Codeforces 383C

我们可以把树按照深度分层。

根是第一层。

根的孩子是第二层。

……

每次如果要修改的点 x 是奇数层的。

那么 x 的所有后代中，奇数层都会加上 val，偶数层的都会减去 val。

Codeforces 383C

如果修改的点 x 是偶数层的。

那么 x 的所有后代中，偶数层的都会加上 val，奇数层的都会减去 val。

因为每个点所在层数的奇偶性是固定的。

因此我们只要用两个数据结构，分别维护奇数层的点和偶数层的点就行了。

Codeforces 383C

这样的话，我们就避免了对 + val 还是 - val 的讨论。

只要给 x 的所有孩子都加上 val / 减去 val 就行了。

那么如何实现对 x 的所有孩子都加上一个值呢？

Codeforces 383C

回忆 DFS 序。

一个点的所有后代在 DFS 序中是一个区间。

现在的目标是区间加+单点询问。

可以用差分序列+树状数组来维护。

Codeforces 383C

对于 A[1] … A[n]，它的差分序列就是：

B[1] = A[1],

B[2] = A[2]-A[1],

B[3] = A[3]-A[2],

…,

B[n] = A[n]-A[n-1]

此时 A[i] = SUM(B[1..i])

Codeforces 383C

对区间 [L, R] 加上 V

等价于 B[L] += V, B[R + 1] -= V

查询 A[i] 只要查询 B[1..i] 的前缀和即可。

时间复杂度 O((n + q) log n)

Codeforces 396C. On changing Tree

有一棵树，对这个树有两种操作：

1 x val k：在 x 号节点上加 val，然后给 x 的每个儿子加 val-k，然后给 x 的每个儿子的儿子加 val-2k，依次类推

2 x：查询 x 号节点上的值。

n， m <= 300000

input

```
3
1 1
3
1 1 2 1
2 1
2 2
```

output

```
2
1
```

Codeforces 396C. On changing Tree

我们仍然把树按照深度分层。

根是第一层。

根的孩子是第二层。

……

设 x 号节点的层数为 d[x]

Codeforces 396C. On changing Tree

如果修改的点是 x，其层数为 d[x]

那么对于 x 的后代 y，它要加的值就是

$$val - (d[y] - d[x]) * k$$

$$= (val + d[x] * k) - k * d[y]$$

注意第一项对于所有的 y 都是一样的。

第二项对于所有的 y，只要维护 d[y] 的系数之和即可。

Codeforces 396C. On changing Tree

(val + d[x] * k) - k * d[y]

仍然利用 DFS 序。

对于 x，要修改的节点是 DFS 序中的一个区间。

维护两个树状数组，分别维护 （val+d[x]*k）的和，和 -k 的和。

时间复杂度O((n+q) log n)

下节课再见