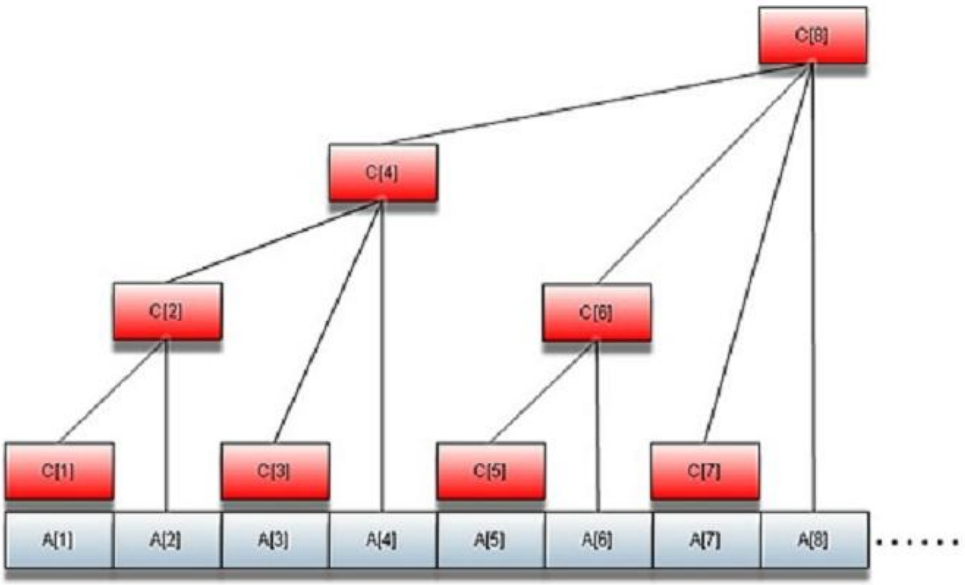
树状数组

题目清单及简易题解



Xiaodai@HIT-ACM Group

2011 年 1 月 30 日

简单题

0、POJ 2309 BST

此题和树状数组无关，是一道数学问题。主要帮助大家理解 lowbit。

1、HOJ 1867 经理的烦恼

此题是最基本的一维树状数组题目，直接进行简单的加一减一（通过判素）操作即可。

2、HOJ 1640 Mobile Phone

该题目是典型的二维树状数组的题目。

3、HOJ 2275 Number sequence

题目就是统计序列中 Ai < Aj > Ak（i < j < k）的个数，可以从前往后统计每个元素之前小于它的数的个数，在从后往前统计每个元素之后小于它的数的个数。然后乘积加和即可。

4、HOJ 2678 Stars

相当经典的树状数组题目，第一感觉容易给人二维树状数组的印象。不过数据范围显然不容许的，先对第一维排序，然后再统计每个位置之前的星星的个数。

5、POJ 3067 Japan

与 stars 极其相似，唯一的不同是上题是统计之前的个数，而这个统计之后的个数，当然我们可以用当前总数 i 减去之前的数即可得到。注意使用正确的数据类型。

6、POJ 2299 Ultra-QuickSort

求逆序数的题目。由于数域范围巨大，需要离散。

7、POJ 2481 Cows

将 E 从大到小排序，如果 E 相等按 S 排序，然后同 stars。

8、 POJ 2029 Get Many Persimmon Trees

O(n ^ 2)枚举起点，再用二维树状数组求其中的点数即可。

9、 HOJ 2686 Magic-Box

三维树状数组，裸到极致。

中等题

1、POJ 2155 Matrix

楼教主的题，经典的二维树状数组，反过来用。

2、POJ 3321 Apple Tree

这道题就是 PPT 中树状数组应用——固定形态的树的子树权和统计问题。难点不在于树状数组，而在于将树状映射为线性。没有相关基础的同学可以跳。

3、POJ 1990 MooFest

这题的难点是要用两个一维的树状数组，分别记录在它前面横坐标比它小的牛的个数，和在它前面横坐标比它小的牛的横坐标之和。

按音量排个序，那么式子为：

```
ans += 1LL * cow[i].volumn  * (count  * x - pre + total - pre - (i - count)  * x);
```

cow[i].volumn 为该牛的能够听到的音量。
count 为在第 i 只牛前面横坐标比它小的牛的个数。
pre 为在第 i 只牛前面横坐标比它小的牛的横坐标之和。
total 表示前 i - 1个点的 x 坐标之和。
分为横坐标比它小和横坐标比它大的两部分计算即可。

4、Hdu 3015 Disharmony Trees

跟上题方法相同，只要按它的要求离散化后，按高度降序排序，套用上题二个树状数组的方法即可。

5、HOJ 2430 Counting the algorithms

这题其实是个贪心，从左往右或者从右往左，找与它相同的删去即可。先扫描一遍记录第一次出现和第二次出现的位置，然后我们从右到佐，每删去一对，只需要更改左边的位置的树状数组即可，因为右边的不会再用到了。

6、TJU 3243 Blocked Road

这题主要在于如果判断是否连通，我们可以先用 j = Getsum(b) – Getsum(a – 1)，如果 j 等于(b – a)或者 Getsum(n) – j 等于(n – (b – a))，那么点 a, b 联通。

7、SPOJ 227 Ordering the Soldiers

这题与正常的树状数组题目正好想反，给定数组 b[i]表示 i 前面比 a[i]小的点的个数，求 a[]数组。

我们可以先想想朴素的做法，比如 b[] = {0, 1, 2, 0, 1}，我们用数组 c[i]表示还存在的小于等于 i 的个数，一开始 c[] = {1, 2, 3, 4, 5}，下标从1开始。

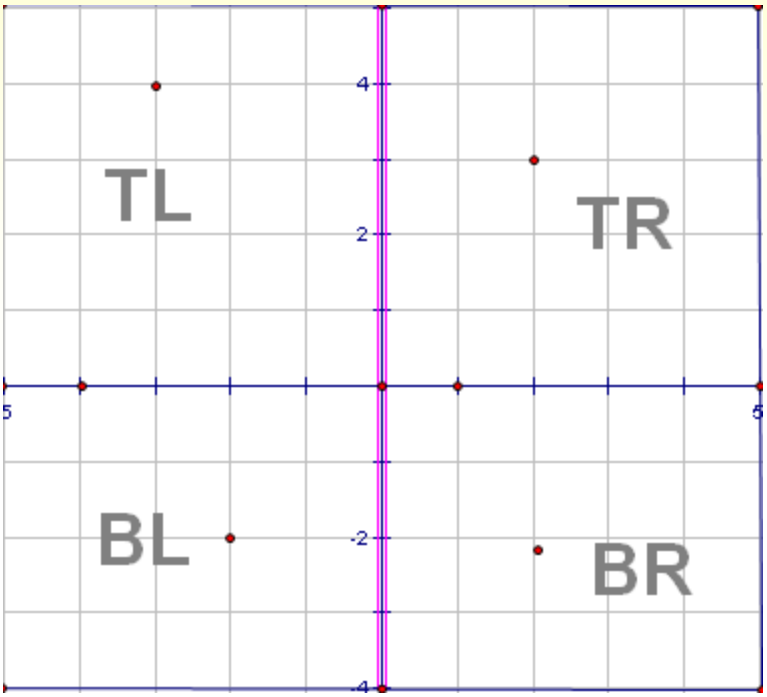
我们从右向左扫描 b[]数组，b[5] = 1，说明该点的数是剩下的数中第4大的，也就是小于等于它的有4个，即我们要找最小的 j 符合 c[j] = 4(这里可以想想为什么是最小的，不是最大的，挺好理解的)，而 c[]是有序的，所以可以用二分来找 j，复杂度为 O(logn)，但现在问题是每次更新 c[]要 O(n)的复杂度，这里我们就想到树状数组，c[i]表示还存在的小于等于 i 的个数，易见树状数组处理方法。处理每个位置的复杂度为 O(logn * logn)，总的复杂度为 O(n * logn * logn)。

8、HDU 2852 KiKi's K-Number

这题与上面那题类似，只是要求比 a 大的第 k 大的数，那我们用 Getsum(a)求出小于等于 a 的个数，那么就是要我们求第 k + Getsum(a)大的数，而删除操作只要判断 Getsum(a) – Getsum(a - 1)是否为0，为0则说明 a 不存在。

给力题

1、POJ 2464 Brownie Points II



首先有 n 个点，过每个点可以做 x,y 轴，把平面切成 BL, TL, TR, BR 四个部分，我们现在的问题是如果快速的计算这四个部分的点的个数。

这样我们可以先预处理，先按 y 坐标排序，求出每个点正左方和正右方的点的个数 LeftPoint[], RightPoint[]，复杂度为 O(n)，同样我们再以 x 坐标排序，求出每个点正上方和正下方点的个数 UpPoint[], DownPoint[]。还要求出比点 i y 坐标大的点的个数 LageY[]。注意：这里要进行下标映射，因为两次排序点的下标是不相同的。

然后按 x 坐标从小到大排序，x 坐标相等则 y 坐标从小到大排序。我们可以把 y 坐标放在一个树状数组中。

对于第 i 个点，求出 Getsum(y[i])即为 BL 的个数，然后 Update(y[i])。由于现在是第 i 点，说明前面有 i – 1个点， 那么

TL = i - 1 - LeftPoint[i]- BL;

TR = LargeY[i] - TL – UpPoint[i] ;

BR = n - BL - TL - TR - LeftPoint[i] - RightPont[i] - UpPoint[i] – DownPont[i] - 1;

这样我们就求出四个部分的点的个数，然后判断有没有当前解优，有的话就更新即可。

2、NOI 2008 糖果雨

CDQ 教母为 NOI 出的经典树状数组题目，有兴趣的同学自行了解，就不提供相关资料了。