树状数组 & 应用

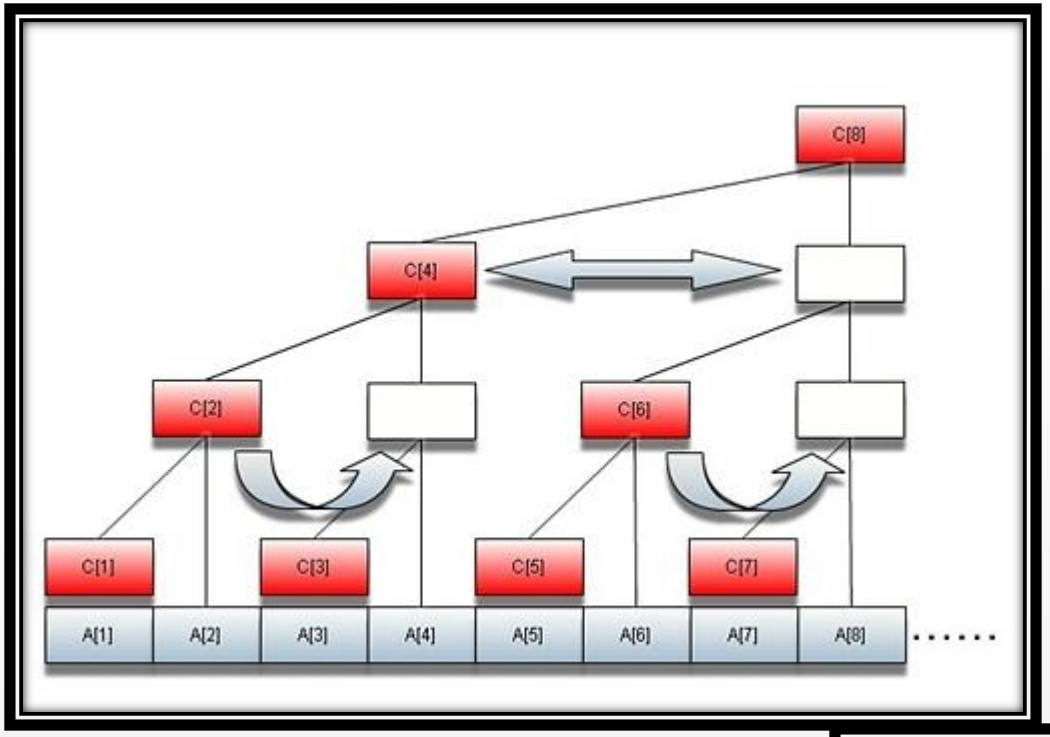
(POJ 2352 & 1195)

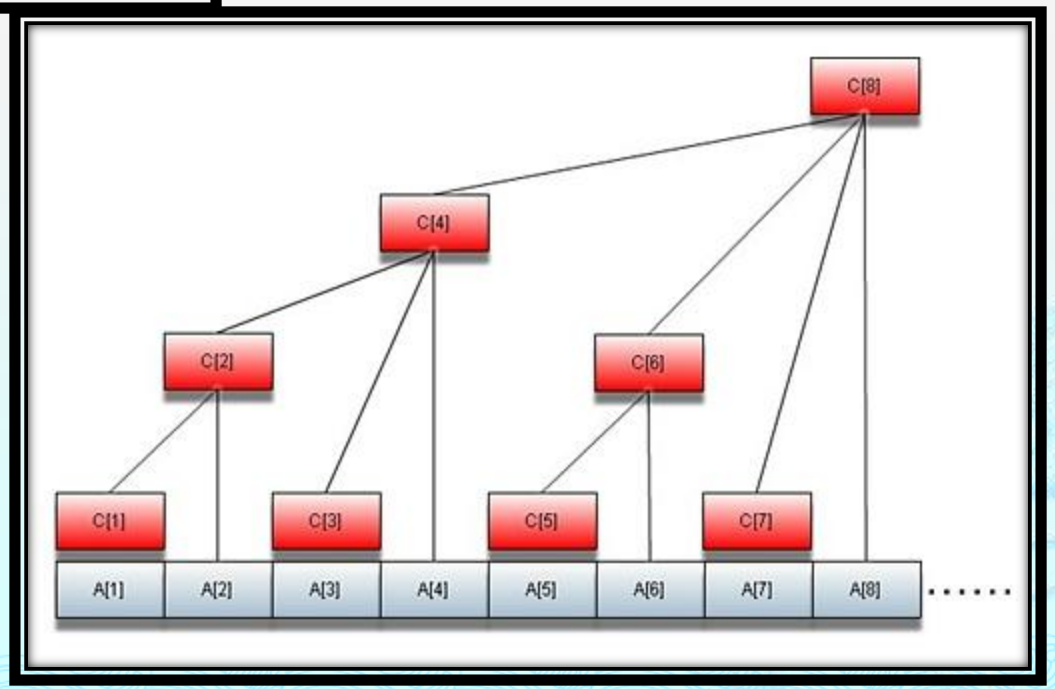
陈曙威

“树状数组时一种非常优雅的数据结构”

- 树状数组查询和修改复杂度都为log(n)级别，在思想上类似于线段树。
- 相比线段树，树状数组需要的空间较少，编程难度也较低，但适用范围比线段树小。

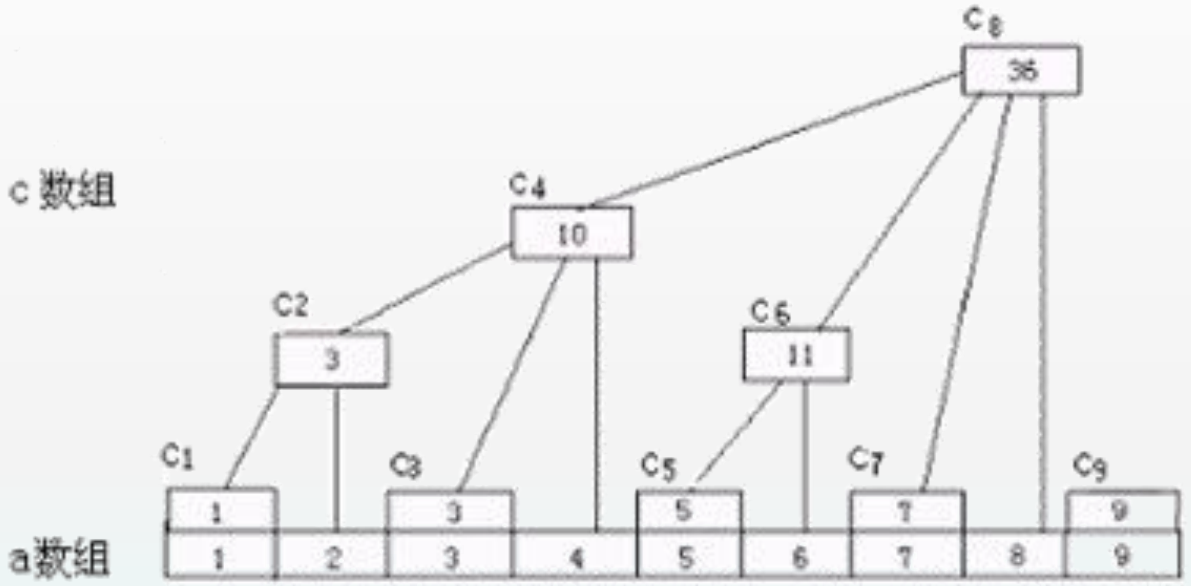
- 假设a[1...N]为原数组,定义c[1...N]为对应的树状数组:
- c[i] = a[i - 2^k + 1] + a[i - 2^k + 2] + ... + a[i]
- 其中k为i的二进制表示末尾0的个数,所以2^k即为i的二进制表示的最后一个1的权值.


C[8]
C[4]
C[2]
C[6]
C[1]
C[3]
C[5]
C[7]
A[1]
A[2]
A[3]
A[4]
A[5]
A[6]
A[7]
A[8]


C[8]
C[4]
C[2]
C[6]
C[1]
C[3]
C[5]
C[7]
A[1]
A[2]
A[3]
A[4]
A[5]
A[6]
A[7]
A[8]

- 令这棵树的结点编号为C1，C2...Cn。令每个结点的值为这棵树的值的总和，那么容易发现：

  C1 = A1
  C2 = A1 + A2
  C3 = A3
  C4 = A1 + A2 + A3 + A4
  C5 = A5
  C6 = A5 + A6
  C7 = A7
  C8 = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8
  ...
  C16 = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10 + A11 + A12 + A13 + A14 + A15 + A16


c 数组
a数组
C1
C2
C3
C4
C5
C6
C7
C8
C9
1
3
3
10
5
11
7
36
9
1
2
3
4
5
6
7
8
9

分析

- 通过建立一个辅助数组c(相当于增加了一倍空间),处理每一个原始数据的点时，仅完成这一点对应树状数组中那一点的修改，并把这个修改传递给它的父亲，辅助数组c就是为了记录这种传递的修改，而当修改到那个父亲点时，则把前面所传递的修改都考虑进去，在修改自身的同时也将修改值传递到更高一层。
- 也即每次修改无需向上直达树根，而是变成一个O(1)的操作，直到最后一次修改完后，才把前面累积的修改一路传递到树根
- （注：建立树状数组时没有这一步，因为树根也需要建立）

建立

- 首先是计算2^k
- (即为i的二进制表示的最后一个1的权值)
  - x&(x^(x-1))
- 利用机器补码的特点，这个函数可以改得更为方便
  - i&(-i)

  - *接下来的建立相当于每次修改一个节点，因此我们先看修改*

修改

- 如果要把a[i]增加m，可以通过调用如下函数实现

- *修改当前节点后，还要把这个结果传递给包含的祖先*
- *Lowbit(i)就是返回i最后一个1的权值，前面已经实现*

```
while (i<=n)
{
        a[i]+=v;
        i+=lowbit(i);
 }
```

- *建立就是在原来初始为0的数组上，*
- *对每个节点i调用一次add就可以完成*

查询

- 如果要统计a[1]到a[i]之间的和，可以通过调用如下函数实现

```
s=0
while (i>0)
{
        s=s + a[i];
        i=i - lowbit(i);
 }
```

- i = i – lowbit(i)这一步实际上等价于将n的二进制的最后一个1减去。而n的二进制里最多有log(n)个1，所以查询效率是log(n)的。

二维树状数组

C[x][y] = ∑ a[i][j], 其中 ,
x-lowbit(x) + 1 <= i <= x,
y-lowbit(y) + 1 <= j <= y.

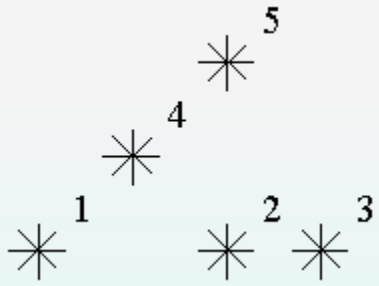
- 修改

```
for(int i = x; i <= N; i += lowbit(i))
  for(int j = y; j <= N; j += lowbit(i))
    C[x][y] += val;
```

- **查询**
  - ```
    int result = 0;
    for(int x = i; x > 0; x -= lowerbit(x))
        for(int y = j; y > 0; y -= lowerbit(y))
            result += C[x][y];
    ```
    ```

应用一（POJ 2352 Stars)

- *Astronomers often examine star maps where stars are represented by points on a plane and each star has Cartesian coordinates. Let the level of a star be an amount of the stars that are not higher and not to the right of the given star. Astronomers want to know the distribution of the levels of the stars.*
- You are to write a program that will count the amounts of the stars of each level on a given map.

- 在整数坐标0<=x,y,<=32000上有很多颗星。
- 每颗星的level就是有多少颗星既不比它高也不比它靠右
- 

- 给出每颗星的坐标。求每个level的星星数。

- 思路:
- 由于题目input的时候，已经是按照Y的升序，（如果Y相同，就按照X升序排列）
- 因此可以做到每录入一个数据，就计算出它的level同时update
- 用star为每个坐标为X的数据做记录（利用树状数组组织）
- Level统计的时候，利用树状数组的查询优势，在o(logn)时间内完成

```
int lowbit(int n).....
int sum(int n).....
void update(int n)
{
    //update all the data which may contain a[n]
    while(n<=32001){
        star[n]++;
        n+=lowbit(n);
    }
}
```

```
for(i = 0; i < n; i ++){
        scanf("%d%d",&x,&y);
        //update the level
        level[sum(x+1)]++;
        //register  the star
        update(x+1);
}
```

POJ 1195 MOBILE PHONES

- Description
- Suppose that the fourth generation mobile phone base stations in the Tampere area operate as follows. The area is divided into squares. The squares form an S * S matrix with the rows and columns numbered from 0 to S-1. Each square contains a base station. The number of active mobile phones inside a square can change because a phone is moved from a square to another or a phone is switched on or off. At times, each base station reports the change in the number of active phones to the main base station along with the row and the column of the matrix.

  Write a program, which receives these reports and answers queries about the current total number of active mobile phones in any rectangle-shaped area.

Instruction	Parameters	Meaning
0	S	Initialize the matrix size to S * S containing all zeros. This instruction is given only once and it will be the first instruction.
1	X Y A	Add A to the number of active phones in table square (X, Y). A may be positive or negative.
2	L B R T	Query the current sum of numbers of active mobile phones in squares (X, Y), where L <= X <= R, B <= Y <= T
3		Terminate program. This instruction is given only once and it will be the last instruction.

（命令0）问题首先定义一个S *S矩阵

（命令1）将点[X,Y]的值加上A（可正可负）

（命令2）要求输出规定矩阵内所有点的和

（就是L到R列，B到T 行）

（命令3）停止

- Sample Input
- 0 4
- 1 1 2 3
- 2 0 0 2 2
- 1 1 1 2
- 1 1 2 -1
- 2 1 1 2 3
- 3
- Sample Output
- 3
- 4

- 思路:
- 由于题目基本上只是在做查询和插入，因此我们可以完全不需要一个基本数组，而直接使用树状数组的就可以完成所有的数据的记录
- 方式一就是二维树状数组的add，只要把对应包含[x,y]项的元素更新即可
- 方式二就是二维数组的查询,但是注意这里的查询下标不是[0,0]，因此我们应该通过四次查询相减求的所需的区间的和
- （同时注意本题目的下标是从0开始，但是一般树状数组的应用时从1开始的，因此所有下标在录入的时候都+1）

- 核心代码:
- Add(x+1,y+1).....
- //因为输入时从0 0 算起
- //树状数组时从1，1算起
- sum(x2+1,y2+1)-sum(x2+1,y1)-sum(x1,y2+1)+sum(x1,y1)
- //画个图就很明显了

THANK YOU