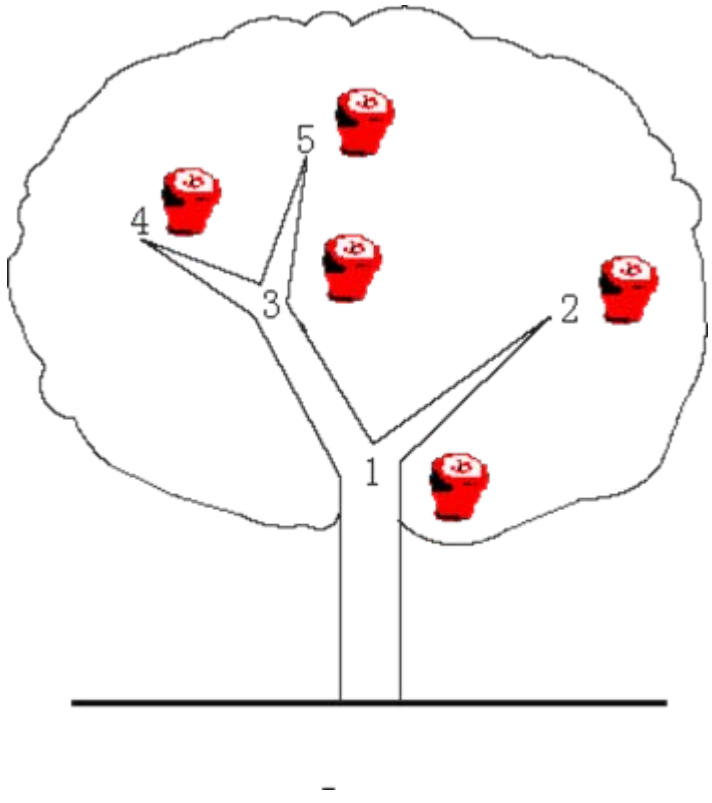
acm International Collegiate Programming Contest

高级数据结构课后习题讲解

唐陈兴

POJ3321 Apple Tree

- 输入一棵树，其中节点1固定为根，对于每个查询x，以x为根的子树具有的苹果的个数。



Sample Input

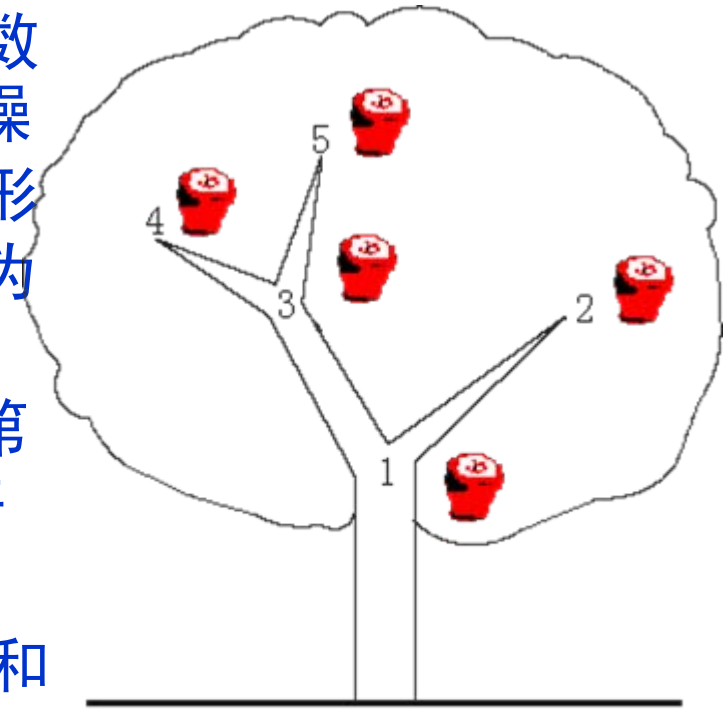
```
3
1 2
1 3
3
Q 1
C 2
Q 1
```

Sample Output

```
3
2
```

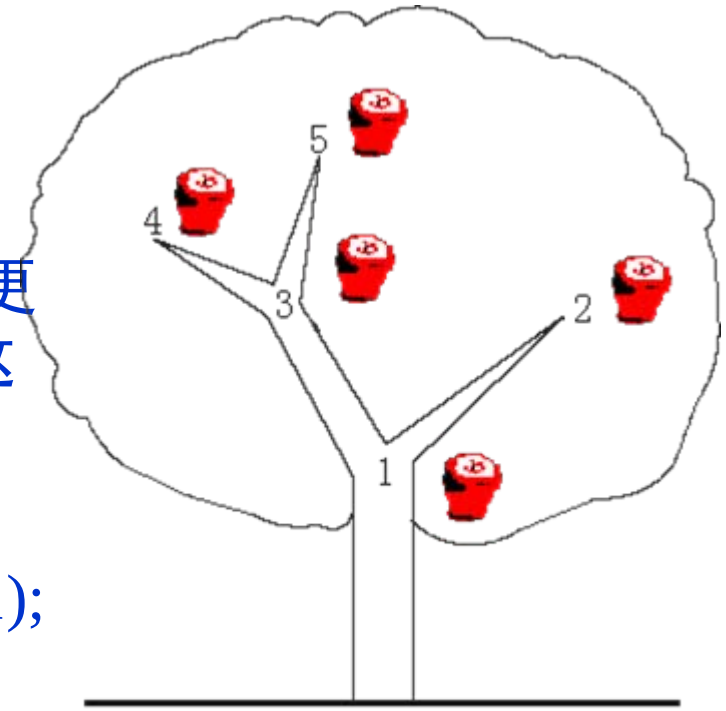
POJ3321 Apple Tree

- 解题说明
  - 树状数组本质是建立一个辅助数组c，对原数组a进行快速求和操作。树状数组不能直接处理树形数据，所以必须把树结构转化为连续的序列。
  - 通过对树进行一次DFS，得到第一次进入Reach和最后一次离开Out的某个节点的编号。
  - 节点x的子树，必然在Reach[x]和Out[x]之间。



POJ3321 Apple Tree

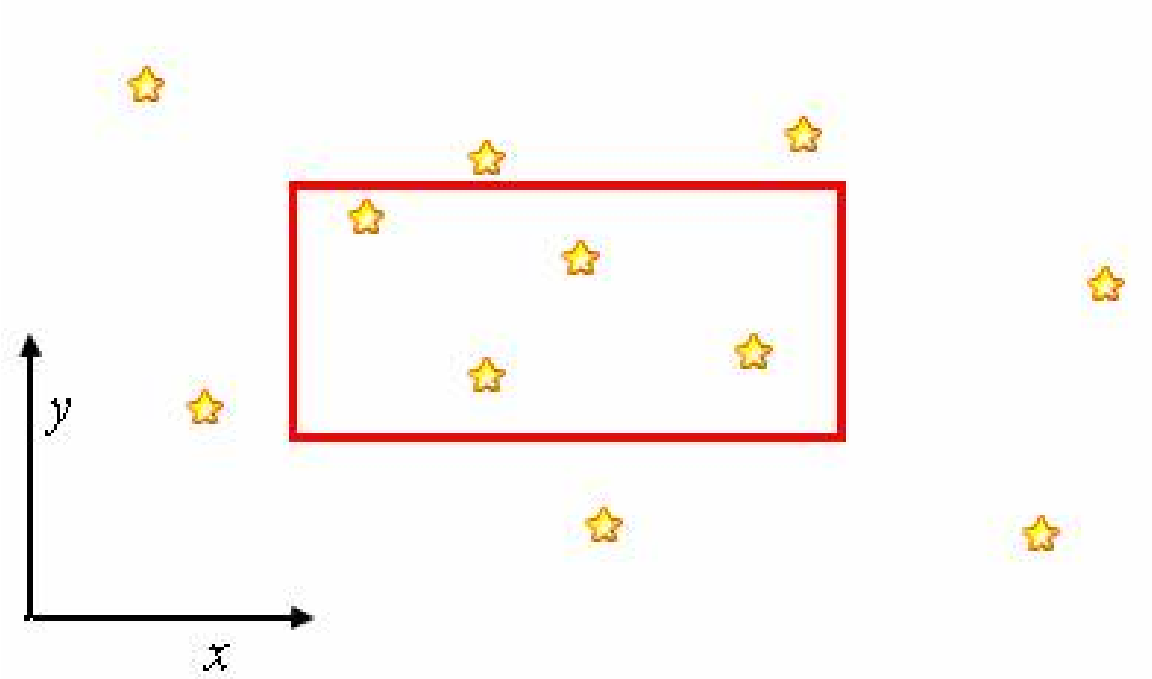
- 解题说明
  - 1<=Reach[x], Out[x]<=2*n-1
  - 1<=x<=n
  - Reach[x]和Out[x]之间的编号，某个点可能存在多次，在实际更新的时候，只要更新Reach[x]这个编号的节点即可。
  - 查询时，以x为根的子树就是Query(Out[x])-Query(Reach[x]-1);



树状数组的扩展

- 二维树状数组 m*n
  - 动态求子阵和
  - logm * logn 的查询复杂度
    - 经典例题：POJ2155 Matrix
    - 一般来说，树状数组查询的是一个区域，更新的是一个点。而这道题查询的是一个点，更新的是一个区域，所以需要巧妙的转化。
- 三维树状数组 m*n*L
  - 动态求三维长方体体积
  - logm * logn * logL的查询复杂度
    - 三维树状数组练习
    - **经典例题：URAL1470 UFOs**
    - http://acm.timus.ru/problem.aspx?space=1&num=1470

POJ2482 Stars in Your Window



- 1<= stars <=10000
- 1<=W，H<=1000000,
- 0<=x，y<2^31
- 用一个W×H的矩形框在平面夜空中

POJ2482 Stars in Your Window

- 数值范围很大，首先必须离散化。
- 二维转化为一维
  - 向右平移两条间隔为w平行线，右边的点不断加入线段树中，平行线左边的点从线段树中删掉。
- 一维的线段，拆成两个点 insert(x, v), insert(x+h,-v) 转化为求前k项最大和问题。
- 细节问题：
  - 输入数据在边框上不算是看得到，而矩形框也不一定要放在整数点上。
  - 解决方法是：x*=2，y*=2，w=w*2-1,，h=h*2-1, 乘2后要用int64.

POJ3171 Cleaning Shifts

- FJ要打扫区间[M..E]， 有N 头奶牛，他们打扫区间[T1,T2]的费用S。
  - 1 <=N<=10,000，0 <=M<=E<=86,399，0<=S<=500,000
- 问最少要花费多少钱才能打扫完区间[M..E]，如果无法打扫完，就输出-1.

POJ3171 Cleaning Shifts

- FJ要打扫区间[M..E]， 有N 头奶牛，他们打扫区间[T1,T2]的费用S。
  - 1 <=N<=10,000，0 <=M<=E<=86,399，0<=S<=500,000
- 问最少要花费多少钱才能打扫完区间[M..E]，如果无法打扫完，就输出-1.

- DP[i]表示打扫区间[M..i] 的最小费用。
- DP[i]=min(DP[a[j].t1-1]..DP[i-1])+a[j].s　iff a[j].t2=I
- 可以用线段树，胜者树来维护最小值。

POJ2985 The k-th Largest Group

- 有n只猫，编号为1..n，初始的时候，放置在1..n组中。有两个操作，1)合并x, y所在组的猫，成为1组，2) 计算当前第k大组的大小。

POJ2985 The k-th Largest Group

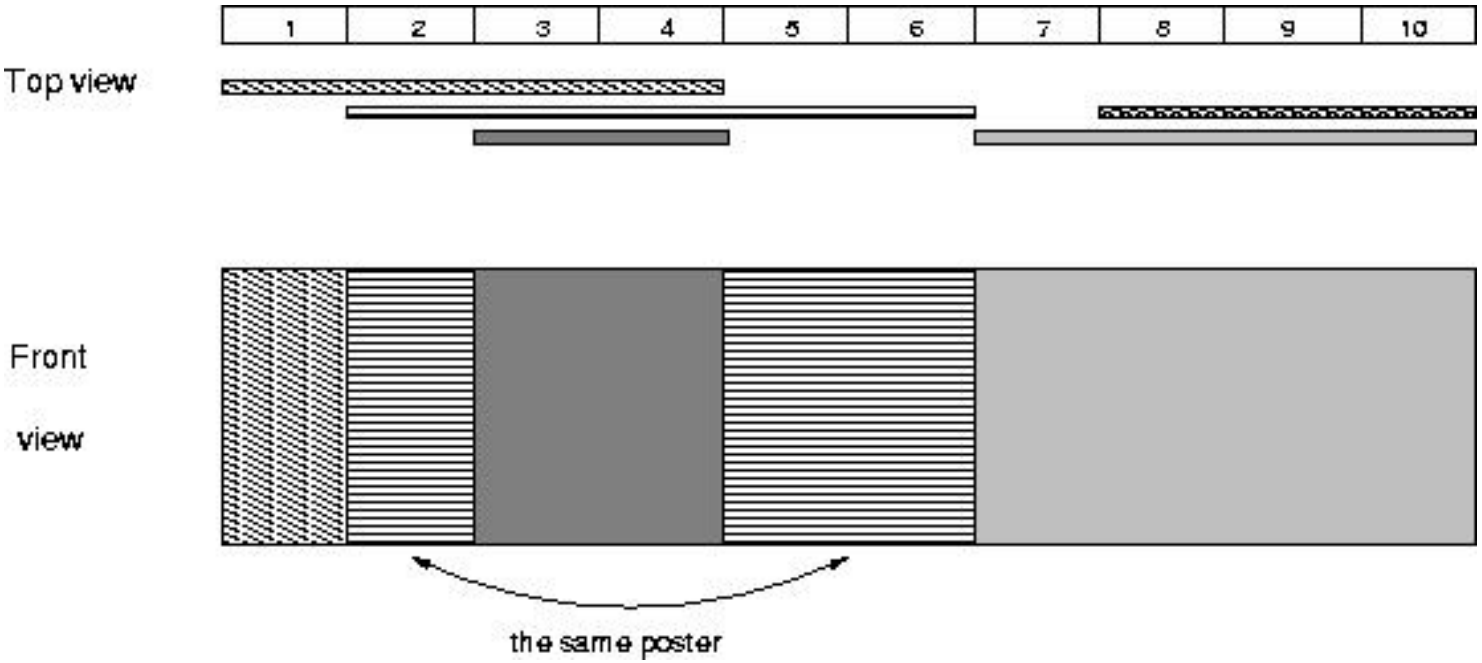
- 有n只猫，编号为1..n，初始的时候，放置在1..n组中。有两个操作，1)合并x, y所在组的猫，成为1组，2) 计算当前第k大组的大小。

- 用并查集合并两个组，计算大小。

- 用线段树查找当前第k大的组的大小。
  - 由于猫最多n只，所以每组的大小最多就是n。
  - Struct record{
  - int L,R;
  - int cnt; //表示大小为L到R的组共有多少个。
  - }

POJ2528 Mayor's posters

- 在一个长10000000的走廊上依次贴n张海报，每张长度为1 <= li <= ri <= 10000000，问最后能看到几张。



POJ2528 Mayor's posters

- 反向处理
  - 这样可以保证后面的海报的优先级比较高
  - 每次用线段树判断当前海报所分解的线段是不是都已经被完全覆盖了
  - Struct record{
  - int L,R;
  - bool flag;
  - };

POJ2528 Mayor's posters

核心代码

```
int update(int left,int right,int t){
        int ret;
        if (a[t].cnt==a[t].R-a[t].L+1) return 0;
        if (left==a[t].L && right==a[t].R){
                a[t].cnt=right-left+1;
                return 1;
        }
        int mid=(a[t].L+a[t].R)/2;
        a[t].cnt=0;
        if (mid>=right) ret=update(left,right,t*2);
        else if (left>=mid+1) ret=update(left,right,t*2+1);
        else ret=update(left,mid,t*2)+update(mid+1,right,t*2+1);
        a[t].cnt=a[t*2].cnt+a[t*2+1].cnt;
        return ret>0?1:0;
}
```

POJ3667 Hotel

- 旅馆有n个连续的房间，初始的时候全为空。
- 两个操作
  - 申请最左边k间连续的房间入住。输出最左边的编号。
  - 从房间x开始的d间房间的客人，都离店。

POJ3667 Hotel

- 旅馆有n个连续的房间，初始的时候全为空。
- 两个操作
  - 申请最左边k间连续的房间入住。输出最左边的编号。
  - 从房间x开始的d间房间的客人，都离店。

- 将问题转化为01, 寻找最左边连续的0
  - **struct** rec{
  - int L,R;
  - int cnt, //表示该线段中最长的连续0
  - lcnt, //表示该线段左边最长的连续0
  - rcnt; //表示该线段右边最长的连续0
  - }a[N];

可能存在的问题，线段更新的时候，该的孩子的孩子没有值。

POJ3667 Hotel

核心代码

```
void update(int left,int right,int val,int t){
        if (a[t].L==left && a[t].R==right){
                if (val==0) a[t].cnt=a[t].lcnt=a[t].rcnt=a[t].R-a[t].L+1;
                else a[t].cnt=a[t].lcnt=a[t].rcnt=0;
                return;
        }
        if (a[t].cnt==0){
                a[t*2].cnt=a[t*2].lcnt=a[t*2].rcnt=0;
                a[t*2+1].cnt=a[t*2+1].lcnt=a[t*2+1].rcnt=0;
        }
        if (a[t].cnt==a[t].R-a[t].L+1){
                a[t*2].cnt=a[t*2].lcnt=a[t*2].rcnt=a[t*2].R-a[t*2].L+1;
                a[t*2+1].cnt=a[t*2+1].lcnt=a[t*2+1].rcnt=a[t*2+1].R-a[t*2+1].L+1;
        }
        int mid=(a[t].L+a[t].R)/2;
        if (right<=mid) update(left,right,val,t*2);
        else if (left>mid) update(left,right,val,t*2+1);
        else{
                update(left,mid,val,t*2);  update(mid+1,right,val,t*2+1);
        }
        a[t].cnt=max(a[t*2].cnt,a[t*2+1].cnt,a[t*2].rcnt+a[t*2+1].lcnt);
        if (a[t*2].lcnt==a[t*2].R-a[t*2].L+1)  a[t].lcnt=a[t*2].lcnt+a[t*2+1].lcnt;
        else a[t].lcnt=a[t*2].lcnt;
        if (a[t*2+1].rcnt==a[t*2+1].R-a[t*2+1].L+1) a[t].rcnt=a[t*2].rcnt+a[t*2+1].rcnt;
        else a[t].rcnt=a[t*2+1].rcnt;
}
```

POJ3667 Hotel

- 核心代码
- int query(int d,int t){
- if (a[t].cnt==a[t].R-a[t].L+1) return a[t].L;
- // if (a[t].L==a[t].R) return a[t].L;
- int mid=(a[t].L+a[t].R)/2;
- if (a[t*2].cnt>=d) return query(d,t*2);
- if (a[t*2].rcnt+a[t*2+1].lcnt>=d) return mid-a[t*2].rcnt+1;
- return query(d,t*2+1);
- }

由此观之，线段树常用来统计数目。

FOJ1375 Josephus Again

A group of N FZU students decide to play the following intelligent game. They first choose three numbers A, B, and P, and then they gather together in a circle. Each FZU student wears a t-shirt with a distinct number from 1 to N on it. The students arrange themselves in a circle so that the numbers on their t-shirts are in consecutive order.

The game has N count-and-eliminate rounds, starting with round 1. In round i, the **(1+((A*i+B) mod P))-th** student is eliminated from the circle, with the counting starting from the student with the smallest number on his/her t-shirt.

The last student to be eliminated from the circle is the winner of the game.

FOJ1375 Josephus Again

- 块状数组模拟计数。

- 其实，本题完全也可以用线段树来计数。
  - Struct record{
  - int L,R;
  - int cnt; //表示区间L..R中还有多少个人活着
  - };

The end

Q & A