Summer2013 Day3 Solution

By Chancellor

Outline

- [网络流] B - Romantic Value
- [分块]C - Chameleon
- [贪心]D – Spaceport Management
- [DP]G – The Big Jubeat
- [Disjoint Set]O – Gears
- [模拟]W - Wonder

Problem	AC	WA	PE	RTE	FPE	SF	TLE	MLE	OLE	CE	Submit
B	12	36	0	0	0	2	7	1	0	4	62
C	1	4	0	0	0	0	8	0	0	0	13
D	1	2	0	0	0	0	0	0	0	0	3
G	8	26	0	0	0	6	12	4	0	0	56
O	9	19	0	3	0	16	14	0	0	2	63
W	21	40	0	0	3	0	2	0	0	3	69
Summary	52	127	0	3	3	24	43	5	0	9	266

B: Romantic Value

题目大意：给定一个无向图及其中的两点p、q，图中的每条边都有一个权值c。现要求删除某些边使得p、q之间不存在任何路径，在被破坏掉边的总权值最小的情况下，使得剩余边权值的总和与删除边数cnt的比值最大：

Maxmize{[sum(c)-rest(c)]/cnt}

分析

1. 显然是求最小割
2. 要让比值尽量大，那么最小割的边数要尽量小
3. 每条边的c值小于1000，那么p、q之间的最小割一定小于M*1000<=1000000；将每条边的c重新定为c*1000000+1，然后求maxFlow；则maxFlow/1000000为最小割，maxFlow%1000000为最少的割边数。代入求比值即可。
4. 注意使用long long，否则maxFlow容易越界。

Code

```
int sum = 0;
scanf("%d%d%d%d", &n, &m, &s, &t);
for(int i = 0; i < m; i++) {
        scanf("%d%d%d", &a, &b, &c);
        addEdge(a, b, c*1000000+1);
        addEdge(b, a, c*1000000+1);
        sum += c;
}
llg mm = maxFlow(s, t);
if(mm == 0)  printf("Inf\n");
else  printf("%.2f\n", (sum-mm/1000000)*1.0/(mm%1000000));
```

C: Chameleon

- 题目大意：给定G组，第i组包含m_i个整数，保证所有给定的整数**互不相同**。再给定Q个询问，每次问第i组和第j组的整数放一块按递增序sort以后，$GroupID_{i-1}$!= $GroupID_i$的数目
- $GroupID_i$定义为sort后第i个数字原本是属于哪一组的。

Example

- 第i组{1, 5, 8}
- 第j组{2, 6, 7}
- Merge以后{1, 2, 5, 6, 7, 8}
- GroupID = {i, j, i, j, j, i}
- Answer = 4

算法

- 令$n = \sum m_i$
- 把Group分成$m_i < \sqrt{n}(1)$和$m_i \geq \sqrt{n}(2)$两种

对于每个询问：

- 如果两组都是第(1)种，那么直接像归并排序那样线性merge, 然后枚举统计。
  $O(\sqrt{n})\ per\ query.$

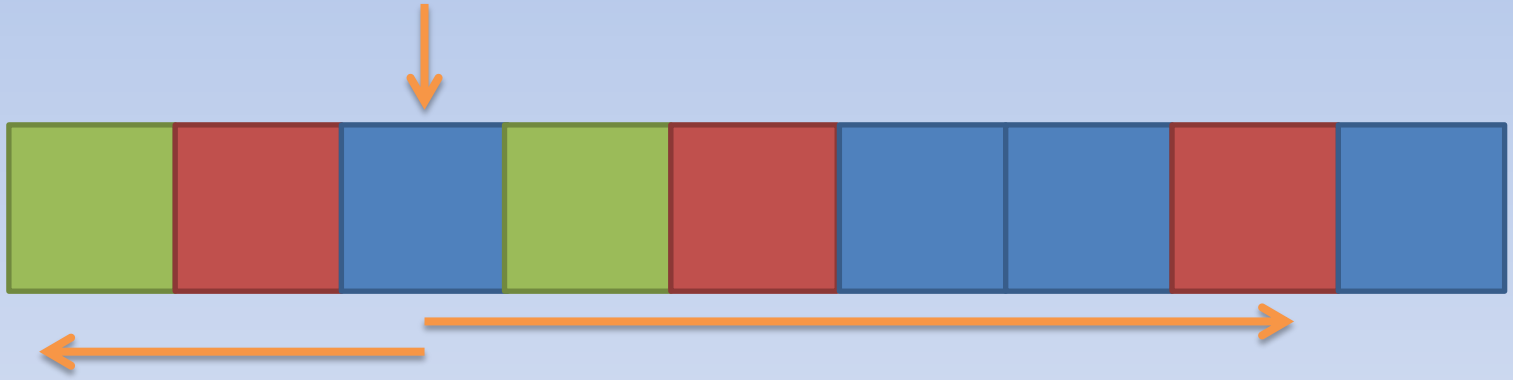
算法

考虑询问包含第(2)种的：

- 因为这时$m_i \geq \sqrt{n}$，所以这样的组不会超过$\sqrt{n}$个。
- 如果可以对每个这样的组在O(n)的复杂度内处理完所有和自己相关的询问，那么就可以做到均摊的$O(n\sqrt{n})$。

算法

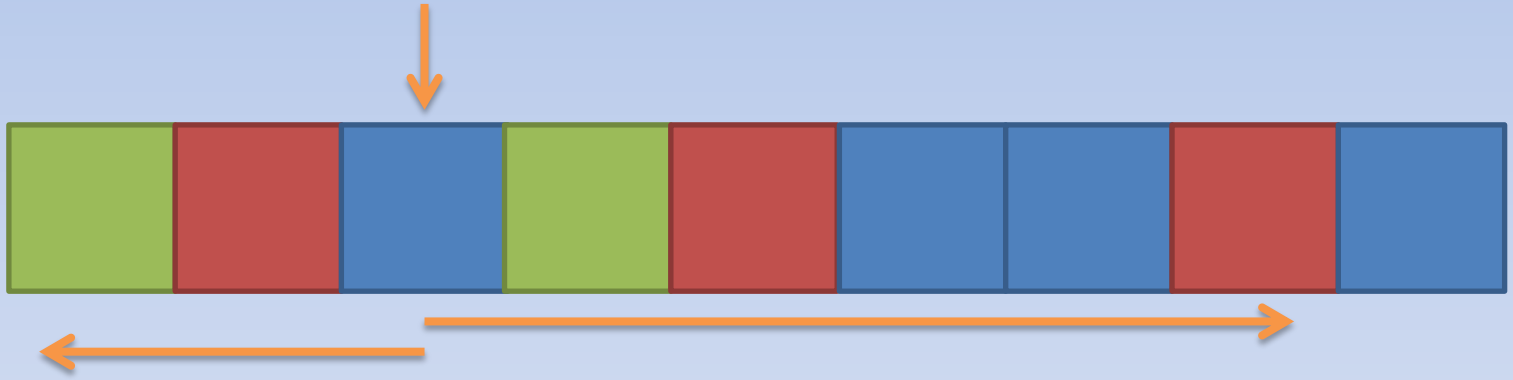
- 把所有组的数字都放在一起sort一遍，从左往右扫
- 现在假设focus在某一组上(此例中的红色)



- 现在扫描到了一个蓝色的组，往左看。
- 若是和前一个红色之间没有别的蓝色，ans[蓝色]++

算法

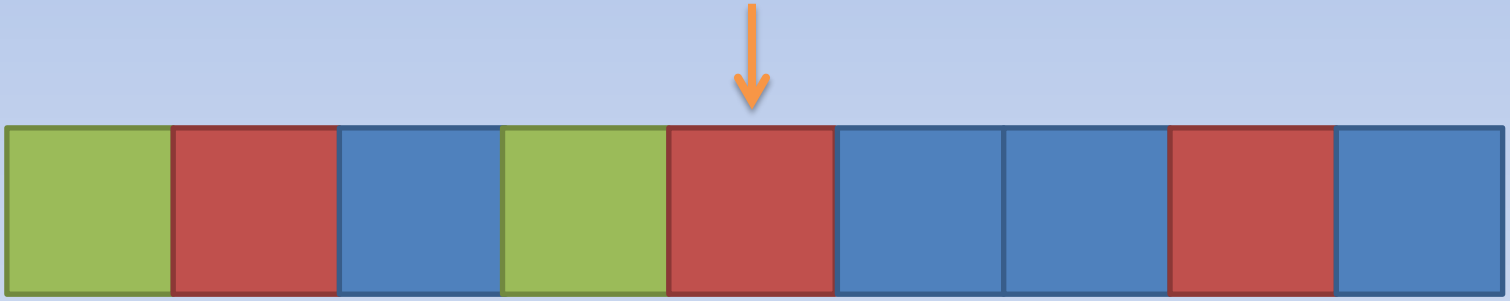
- 把所有组的数字都放在一起sort一遍，从左往右扫
- 现在假设focus在某一组上(此例中的红色)



- 现在扫描到了一个蓝色的组，往右看。
- 若是和后一个红色之间没有别的蓝色，ans[蓝色]++

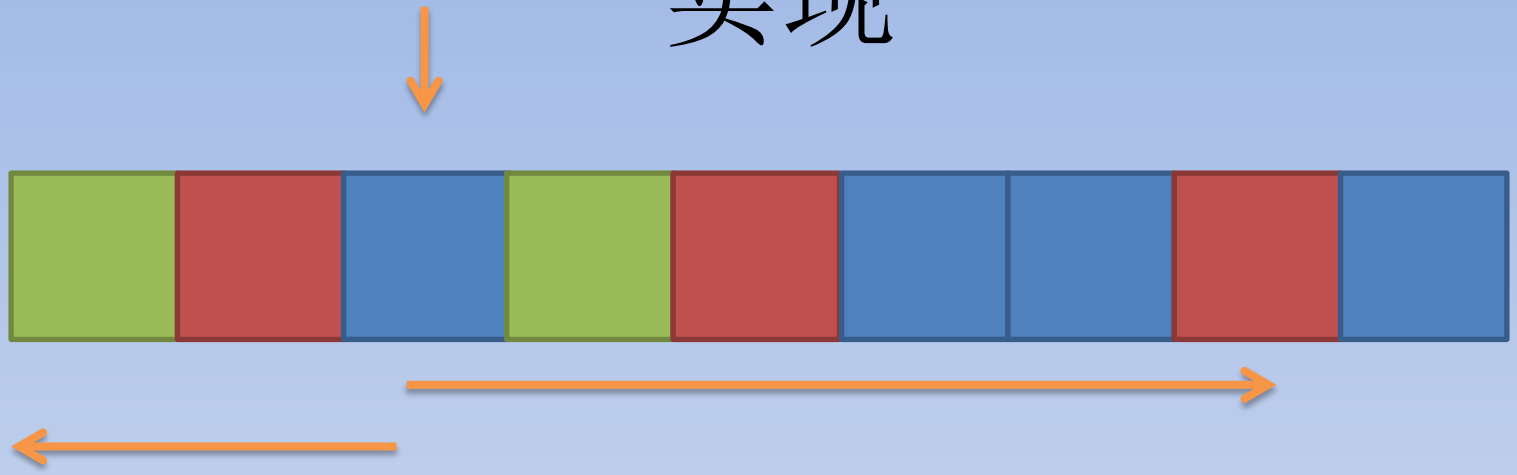
算法

- 把所有组的数字都放在一起sort一遍
- 现在假设focus在某一组上(此例中的红色)



- 若是扫描到红色，不更新答案。

实现



- nxt[i]表示下一个和i同色的块的下标
- pre[i]表示上一个和i同色的块的下标
- cur表示最后一次扫描到的红色的块的下标
- 然后就可以直接判定上面提到的条件

Summary

- 对于$m_i, m_j \leq \sqrt{n}$ 的询问　$O(\sqrt{n})\ per\ query$
- 否则，通过第二种方法，均摊$O(n\sqrt{n})$处理出所有和大块相关的询问。
- 总时间复杂度$O(n\sqrt{n} + Q\sqrt{n})$
- 空间复杂度$O(n)$

D: Spaceport Management

- 题意：管理一个有m容量的地方，共n 个时间段
- 如果第i个时间段开始有ci，可以在开始的时候改变成[0,m]中的任意整数（花费1的代价），或者不改变（0代价）
- 结束后变成$c_{i+1} = c_i + a_i$
- 当然c_{i+1}必须满足$0<=c_{i+1}<=m$

Solution

- 类似贪心
- 策略：能不管理就先不管理，因为后管理可以变到任何一个可能的情况，肯定优于先管理。
- 对于某次管理后数值的确定：可以暂时先待定，记下可能的范围[0, m]，往后处理，变化a_i后，变成[ai, m + ai]，其中越界的部分就变成了无效的，修正一下会变成一个区间[l, r]。
- 只要[l, r]区间还存在可以取的值，在当初待定的那次管理处就可以取一个对应的值来保证一直到现在都不会越界。
- 直到某次更新后发现[l, r]==空集，这次就必须管理了，用同样方式处理。
- 初始情况可以等价于从$[c_0, c_0]$这个区间开始。

G: The Big Jubeat

在一个4*4方格上有多个人，每个方格上最多只能站一个人，有n个时间段，每个时间段最多可以将1个人移动到相邻格子

在每个时间段，在一些格子上会有分数，如果那个时间段有分数的格子上有人，则可以得到那个格子的1分。

问在得到分数最多的情况下移动的最小步数

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

显然是状态压缩DP
记F[i][mask]为第i个时间段后人的位置为mask的（最大分数, -最小步数）

F[i][mask] = max(
F[i-1][mask] + (c(mask & p),0)
F[i-1][mask2] + (c(mask & p),-1)
)
mask2 为 可以通过移动1个人到相邻格子变为mask的状态
c(n) 表示n的二进制下1的个数

可以通过滚动数组来减少内存的使用

O：Gears

【题目大意】

将齿轮分成若干组，每组的齿轮有两种属性：顺时针旋转或逆时针旋转。

要支持以下操作：

1）将两个齿轮所在的组合并，两个组之间确定旋转关系

2）将一个齿轮分离出原来的组，成为一个新组。

3）询问两个齿轮的转向关系。

4）询问齿轮所在的组的齿轮个数。

【解法】

题目的操作普遍是关于关系的处理，可以用并查集来处理。

每个齿轮组转向相同的齿轮放到同一个集合，并且统计个数。

对于同组但转向不同的齿轮用一个数组映射到另一个集合。

如果两个齿轮在同一集合，则同向；

如果一个齿轮能映射到另一个齿轮，则反向；

否则Unknown。

【从并查集里面删除一个点】

直接删除的困难:

如果要删除的结点为根结点, 则修复并查集的指向要用 O(节点数)的时间。

解决办法:

初始化时,每个结点加入一个虚根, 使得每个原结点在并查集表示的树上面均为叶结点。

齿轮组的大小统计：根节点保存该子树的大小，在连接的时候，将子树的大小累加到新根上，输出时将该齿轮所在的根节点和不同转向的映射的根节点大小之和输出即可。

所有操作均在disjoint set
的时间复杂度内完成

空间复杂度O(N+M)

By Kotomi

Wonder

题目大意：

1、N位玩家进行回合游戏， 每回合按照1-N的顺序依次行动

2、每位玩家在自己当前回合结束后，总产能将会累加上自己该回合产能。

3、每当玩家在第i回合派遣一组工人砍伐树林，在第i+2回合他的行动阶段时将会获得20产能的额外加成

4、任一玩家率先完成奇迹的建造后，游戏结束

按照所给条件直接模拟即可

Some tips:

存在所有城市产能均为0的情况

存在产能为0的城市通过砍树完成奇迹的情况

存在一位玩家在同一回合内多次派遣工人的情况