NOip easy version solution

Foreseeable97

July 27, 2015

Contents

1 problem 1:数

1.1 回顾题面

现在有一个数x。
我们对其做k次变换，假设第i次变换前x的值为x_i。
那么第i次变换就是将x赋值为大于等于x_i的最小的i的倍数。
第i次变换更加形式化的说法就是:
$x = \lceil \frac{x}{i} \rceil i.$
现在告诉你x初始等于n，问对它做k次变换后，x是多少。

1.2 30%的数据

暴力模拟，没有拿到这部分分的人心态不对。

1.3 100%的数据

我们定义f_i为变换了i次之后的x。
显然有递推式：$f_i = \lceil \frac{f_{i-1}}{i} \rceil i$
定义g_i为$\frac{f_i}{i}$
那么显然:$g_i = \lceil \frac{(i-1)g_{i-1}}{i} \rceil$
容易发现，当$g_{i-1} < i$的时候，$\lceil \frac{(i-1)g_{i-1}}{i} \rceil$就等于$g_{i-1}$
证明请自行证明。
也就是说当$g_{i-1} < i$的时候，之后的$g_x(x \geq i)$都满足$g_x = g_{i-1}$。
那么现在的问题是需要多少次变换，才能使得$g_{i-1} < i$。
可以证明只需要$O(\sqrt{n})$次就可以了。
f_i最多比f_{i-1}多i。
这个很显然。
那么假设变换了m次之后，$g_m < m$(变换失效)
那么就有$f_m < m^2$
考虑$f_m \leq n + m(m-1)/2$
所以当$n + m(m-1)/2 < m^2$的时候。
f_m必定小于m^2。
所以$m = O(\sqrt{n})$的时候，变换失效,$g_i = g_{i-1}$。

2 problem 2:图

2.1 回顾题面

略。这题是noi2014的某题的弱化版。

2.2 30%的数据

都是一棵树了，1到n的简单路径一定唯一。(重复经过一个点的路径显然不够优，答案一定可以是简单路径)

找出来就行了。

dfs或bfs。

2.3 100%的数据

假设我们知道了答案路径上经过的边中b的最大值是B。

那么我们可以把图里面的所有边的b属性大于B的边删掉。

然后我们的任务就是找一条经过的最大a最小的路径，这就是答案路径了。

但是我们不知道答案路径经过的边中b的最大值。

解决方法是枚举.

我们枚举m条边的每一条边的b权，这一部分复杂度是$O(m)$的。

另一个子问题是找一条经过的最大a最小的1 到n的路径。

我们可以想象出来，如果把这些边，按a值从小到大加到图里，第一个使得1和n在同一个连通块的边，一定就是a的最小值了。

这一部分的做法加上排序是$O(mlgm)$的。

所以总复杂度是$O(m^2lgm)$。

因为这是在维护最小生成树。。所以实际上还可以优化到$O(m^2)$。但作为noip模拟题就算了。

实际上还有用lct的$O(nlgn)$做法。

3 problem 3: 括号序列

3.1 回顾题面

定义“四面体”为以下整点组成的集合：
$T(n) = \{(x, y, z) | 1 \le z \le y \le x \le n\}$
$T(n)$表示边长为n的四面体内的点集。
我们定义“子四面体”为以下整点组成的集合：
$sT(x, y, z, a) = \{(x+i, y+j, z+k) | 0 \le k \le j \le i < a\}$
一开始，你有一个四面体$T(N)$，每一个$T(N)$里的点的权值都是0.
然后你需要处理M个操作：(Mxi, Myi, Mzi, Mai)，表示你需要将子四面体$st(Mxi, Myi, Mzi, Mai)$中的所有点的权值+1。
然后你需要回答Q个询问：(Qxi, Qyi, Qzi, Qai)，表示你需要输出子四面体$st(Qxi, Qyi, Qzi, Qai)$中的所有点的权值之和。

3.2 30%的数据

每次询问修改都$O(n^3)$暴力，总复杂度$O((m+q)n^3)$就能过了。

3.3 100%的数据

首先考虑一维下的版本。（数列，区间加，区间和）
针对本题先修改再询问的特点，我们可以在修改时只保存差分后的数组。
在处理询问前求前缀和还原原数列。之后再求一次前缀和就可以做到O(1)回答询问了。
现在试图把它推广到二维。（三角形，子三角形加，子三角形和）
如果朴素地对每一行应用一维的差分法，最后得到的标记会像下面一样：
O
+1 -1
+1 O -1
+1 O O -1
+1 O O O -1
其中+1标记和-1标记都是连续的一段。
标记也是一种值，所以可以对标记打标记！
于是我们将所有的+1标记按从右上到左下做差分，将所有的-1标记按从左上到右下做差分。
这样每个操作实际上只需要在两个数组中修改四个值，最后利用这两个数组还原原来的标记，利用标记还原原数阵。前缀和的计算也是类似，只不过第二维的前缀和要从两个方向各算一遍。
类似地，只需要4个数组就可以表示三维情况下的所有标记，经过三轮不同方向上的求前缀和就可以得到原四面体，前缀和的计算也需要4个数组。这样单操作/单询问要拆成8个，是$O(1)$的，复杂度是$O(N^3 + M + Q)$。