HNOI2018 模拟赛 Solution

ztzshiwo

2018 年 2 月 22 日

1 line

Source : hihocoder#1315 Reachable Permutations
官方题解: hihoCoder 挑战赛 21 C

排列找不到什么很好的性质, 只能爆搜. 或许有什么神乎其技的优化能拿到高分.

既然如此我们考虑转化, 如果不是排列而是一个 01 序列的话, 显然序列 A 可达序列 B 当且仅当序列 A 中的第 i 个 1 在序列 B 中第 i 个 1 的左边(或位置相同).

对于排列 A 中的每一个数 a_i, 将 $\geq a_i$ 的数赋为 1,$< a_i$ 的数赋为 0, 得到 n 个 01 序列.

可以证明排列 A 可达排列 B 当且仅当排列 A 生成的每一个 01 序列可达对应的排列 B 生成的 01 序列

.

必要性: 如果任何一个 01 序列不满足条件, 若第 i 个 1 不满足条件, 可知前面的 1 都不能移动, 而后面的 1 无论怎样都不能移动到前面的位置, 必要性得证.

充分性: 只需构造一种方案使得对于任意满足条件的排列都可达. 从大到小考虑每一个数, 对于大于当前数的位置肯定都固定了, 不用考虑. 假设当前数需要从 a 到 b, 每次找到 $(a, b]$ 中最大的数位置 i, 然后交换 a 和 i, 对于小于当前数的数, 可知这样只会让大的数到小的数前面, 因此将当前数移到目标位置后的排列生成的 01 序列仍然满足条件.

据此我们可以得到一个 $O(n2^n)$ 的做法: 通过状压 dp 求出从全 0 序列开始每次放入一个 1, 最后变成全 1 序列, 且过程中每一个 01 序列都满足条件的方案数. 状态 x 表示 01 序列为 x, 能到达当前状态的方案数, 然后直接枚举下一个 1 的位置转移就行了.

2 porcelain

Source :

hihocoder#1381 Little Y's Tree

UOJ#194 思考熊

部分思路:

虚树上找最远点的处理 多个联通块的处理

部分分的做法:

如果所有的断边都相同的话, 对于每一个联通块, 判断一下关键点的情况(在当前联通块的哪个方向), 可以暴力算出答案.

如果所有的关键点都相同的话, 以关键点为根, 按 DFS 序建立深度的线段树, 拿个栈维护每个区间属于哪个联通块. 直接区间信息合并就可以了.

如果认真看完这两道题的题解的话, 这道题也就迎刃而解了. 但是题解好像很长, 而且充斥着无聊的内容.

我就大概讲一下做法. 首先可以考虑如果边权非负, 那么一个点到树上的最远距离肯定是直径两个点中的一个. 我们切 k 条边, 会导致树变成 $k+1$ 个联通块, 在 DFS 序上分成的区间也一定是 $O(k)$ 的.

可以直接线段树维护区间内点的直径, 然后用栈维护每一个区间属于哪个联通块, 询问时查询直径中哪个点最远, 把答案取 MAX 输出. 时间复杂度 $O(nlog^2n)$(倍增或链剖) 或者 $O(nlogn)$(RMQ).

但是如果边权可以为负就不能这么做了... 反例可以自己随便找找.

这样的话就需要一些奇奇怪怪的做法了... 我直接附个文件吧... 其实就是我以前讲课的课件 (有人有印象吗?)

通过那个奇奇怪怪的做法, 我们就可以用 $O(nlog^2n)$ 的时间复杂度解决这道题了.

有人想要吐槽这道题吗?

3 tower

Source : hihocoder#1456 Rikka with Lattice

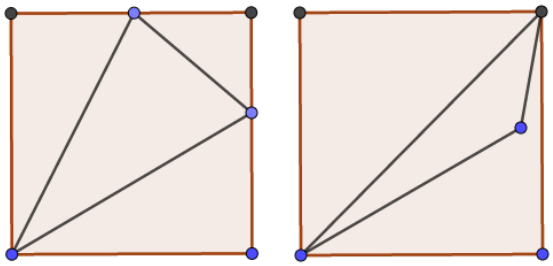
官方题解: hihoCoder Challenge 25 D

其实我这道杜教筛是给后面某 lxswkrsdrtsbrz 的强者做铺垫的.

做他的题的时候你们自然会感受到.

这道题是我见过的少数几道看上去不那么杜教筛的杜教筛...

我们把三角形限制在一个矩形范围内, 可以发现只有两种情况:



考虑第一种情况可以得到 $ab = nm + 1$, 而 $a! = n, b! = m$, 显然不存在合法解.

只需要考虑第二种情况. 计算面积代换公式可以知道 $|am - nb| = 1$, 不在边界上的点可以在上方或下方. 每一个方程当且仅当 $(n, m) = 1$ 时有且仅有一个合法解.

因此对于一个 $(n, m) = 1$ 的 $n * m$ 矩阵, 一定会有 4 个合法的三角形, 可以列出式子.

$$Ans = \sum_{i=1}^{n}\sum_{j=1}^{m} 4[(i, j) = 1](n - i)(m - j)$$

然后就可以用杜教筛求解了. 时间复杂度 $O(n^{\frac{2}{3}})$ 或 $O(n^{\frac{2}{3}}logn)$.

我的题解很简陋, 可以去看官方题解. 如果官方题解还是看不懂的话可以来问我.