画立方体

模拟，基本功

旅行游戏

考虑点权最大的那个结点（设为 u），显然不管从其他哪个结点开始跳一次，都可以到达 u，且不能越过 u，到达 u 后也不能再跳向其他结点。于是结点 u 一定是最优方案中最后一个到达的点。我们将结点 u 从树上去掉（相连的边也去掉），对剩下的每个连通块都递归进行上述过程即可得到答案。

但要实现上述递归过程并不容易，我们转而考虑反向进行这个过程：按点权从小到大枚举结点 u，并将 u 作为所有与 u 相连的（已经枚举过的）连通块的根。从而建立一棵新树，每个结点的深度（根的深度为 1）即是答案。

时间复杂度 $O(n\log n)$。

CSDN @CofDoria

ICPC 网络赛里的一个题。我很喜欢。虽然稍显套路，但是在大数据结构横飞的世界中可以算为一股清流。

最小公倍数

设$A=a+x\ge a,B=b+y\ge b$，题中条件转化为$\frac{AB}{\gcd(A,B)}=c$，再设$d=\gcd(A,B),A=dA',B=dB'(\gcd(A',B')=1)$，上式转化为$dA'B'=c(\gcd(A',B')=1)$，要求$d(A'+B')-a-b$最大值。

a,b为定值，代掉d，转化为求$c(\frac{1}{A'}+\frac{1}{B'})$最大值，条件为$A'\le\frac{c}{b},B'\le\frac{c}{a},A'B'|c,\gcd(A',B')=1$。

看一眼数据，发现c一共只有不超过18个质因子，于是考虑暴搜，直接暴力枚举哪个因子分配给谁是$O(3^n)$的，看上去很能过但是由于$int128$比int多了4倍常数，根本过不了。

其实到这里答案已经近在眼前了，搜索过程中搜出A',B'使用质因子集合s所能达到的合法的最大值fa，fb，然后只需要一个非常naive的递推求出$fb[s]$表示s即s的子集所能达到的B'的最大值即可。

这样的话搜索的复杂度相当于枚举每个质因子的每一次是否需要选择，为$O(2^n)$，瓶颈在递推，为$O(n2^n)$。

极大上升子序列

可以很自然地写出dp：设$f(i)$表示以i结尾的子序列的数量，$f(i)$对$f(j)$产生贡献（$i < j$）的条件是$a_i < a_j$且(i,j)之间没有k使得$a_i < a_k < a_j$。

为了方便计算，可以取$a_{n+1} = n+1$，这样最后答案就为$f(n+1)$

这种dp一般都应该要考虑分治，现在考虑$[l, mid]$的数如何对$[mid+1, r]$中的f产生贡献，我们发现对$f(i)$产生贡献的数其实是一个单调下降的数列，这个数列恰好对应以权值为下标做一个单调栈的过程中，完成了所有比a_i小的数的单调栈的一段区间。

不妨按照权值从小到大考虑，同时维护一个左边的以权值为下标（从小到大），位置单调下降（从大到小）的单调栈，可以维护一个以权值为下标的树状数组，以查询当前单调栈中的某一段区间和。

接下来考虑怎么找这段区间，对于右边的一个a_i，其实就是要找到$[mid+1, i-1]$这个区间中最小的一个数，因为我们是按权值从小到大考虑，如果我们对右边也维护一个权值递增、位置递减的单调栈，这样的位置其实就是当前这个单调栈中比它位置靠左的数的最大值。

时间复杂度$O(nlog^2n)$。