NOIP 2023 模拟赛　题解

2023 年 10 月 1 日

目录

A 枣树（tree）

考虑 T 的直径，设它是从 u 到 v 的。显然 u, v 均为叶节点。

由于 T^m 是 T^{m-1} 在叶子下面挂一棵树得到的，因此 T^m 的直径一定是将 T^{m-1} 的直径的两个端点分别在对应的 T 中向下走最深的路径得到的，故长度等于“T 的最深路径长度”+ “T^{m-1} 的直径”+ “T 的最深路径长度”。

于是, $\operatorname{diam}(T^m) = \operatorname{diam}(T) + 2(m-1)\left(1 + \max_{v\in T} \operatorname{dist}(1, v)\right)$.

简单 DP 一下即可，时间复杂度 $O(n)$ 。

B 对弈（gomoku）

说白了就是黑白双方下 k 子棋，构造平局。

先来寻找无解的条件:

当 $n=1$ 或 $m=1$ 时，不妨设 $n=1$ ，则 $k=1$ 时显然无解，$k\geq 2$ 只需黑白交错即可有解。

接下来考虑一般的情况。$k=1$ 时还是无解，考虑 $k=2$ 的情况。

由于 $\min\{n,m\}\geq 2$ ，因此一定存在一个 2×2 的子棋盘。如果这个子棋盘上有 2 个黑子，则一定存在两个黑子共线，否则，一定存在 2 个白子，故有两个白子共线。因此 $k=2$ 也是无解的。现在考虑 $k\geq 3$ 的情况，如果我们对 $k=3$ 给出了一个构造，即不存在 3 个同色棋子共线，那么这个构造对于所有 $k\geq 3$ 都是适用的。

这个具体的构造如下表所示，其中第 i 行第 j 列的数为 $\left(i+\left\lfloor\frac{j}{2}\right\rfloor\right) \bmod 2$:

$$
\begin{gathered}
10011001100\ldots\\
01100110011\ldots\\
10011001100\ldots\\
01100110011\ldots\\
10011001100\ldots
\end{gathered}
$$

容易证明，不存在 3 个 (同色) 棋子在横向、纵向、斜向共线，从而这个构造符合第一个要求。第二个要求即为，两种颜色的棋子的数量差不超过 1 (否则这个局面无法下成)。

如果 n 是偶数，则每一列包含 $\frac{n}{2}$ 个 0 和 $\frac{n}{2}$ 个 1，从而符合要求。如果 m 是偶数，则每一行包含 $\frac{m}{2}$ 个 0 和 $\frac{m}{2}$ 个 1，也满足要求。

如果 n,m 都是奇数，则前 $n-1$ 行的 0,1 可以相互配对 (两行两行配对)，最后一行 1 的个数前 $m-1$ 个数中也恰好包含 0,1 各一半，因此也是满足要求的。于是 $k\geq 3$ 的情况成功构造，总时间复杂度为 $O(nm)$ 。

C 括号（kacbret）

首先尝试发现贪心是不行的。有一个关键的 observation：如果删除一对 ()，那么其中的字符必须要全部删除，要不然字典序不会变小，所以只需要使用相邻的删除操作就可以得到最优解。这说明我们可以把问题转化成保留原序列的若干连续段，使得剩下的串字典序最小。

显然这是一个简单的子序列 dp 模型，考虑到字典序的特性我们从后往前 dp。设 f_i 表示操作后 i 个字符留下来的字典序最小的串，转移时可以在 f_{i+1} 的基础上直接添加，还可以找到和当前的（在原串上配对）的位置，然后删除这一整段（根据结论这是唯一其他需要考虑的），所以可以得到（设 nxt_i 表示配对字符的位置）：

$$f_i = \min\left(s_i + f_{i+1}, f_{nxt_{i+1}}\right)$$

剩下的问题变成了快速比较两个字符串的字典序。第一种方式是哈希求出最长公共前缀，可以主席树暴力维护。更好的方法是维护一个动态增加叶子的 trie 树，用倍增的方法跳最长公共前缀，如果哈希值相同就往上跳。

时间复杂度 $O(n \log n)$。

总结：

字典序问题贪心不是唯一解，倒序 dp 同样充分利用了字典序的性质。

复杂的问题可以借助性质转化成简单的 dp 模型，本题就是先证明只需要使用连续段，然后可以转成线性 dp。

D 染色（color）

这样的问题当然首先考虑当对于某一个黑方给定的初始节点，白方的最优策略是什么样的。不排除出现“每个点的最优策略不都好算，但是最劣点的最优策略很好算”的情况，但是先不担心这一点。

首先考虑什么情况下会被迫只扩张一个点。思考特殊性质（环，仙人掌）之后发现扩张过程大概率是被一个环卡住的，走到那里无路可走了。

进一步地，发现**对于一般无向图，卡住只可能发生在一个点双的边界上**，即接下来扩张的那一个点一定是该点双最后一个被染色的点。证明很容易：如果不是，那这个点将一个点双分割成两个互不相连的集合，矛盾。

于是感性理解一下，走得越远，遇到的点双数越多，就越不容易被卡住。

严谨来说，考虑第一次被卡住的时间，一定这之前的点都被染黑了。设卡在的点为 u，当前染了 d 次（不包括第一个点，下同），那么一共染了 $2d-1$ 个点，且 $\operatorname{dis}_u \le d$。

另一方面，显然 d 越大越优（窝在那里面干嘛）。所以令 c_d 表示距离 $\le d$ 的点数量，设 d_0 为最小的满足 $c_d = 2d-1$ 的点，那么第一个被卡住的点至少是距离 d_0 的这个点，并且等号显然可以取到。

不断进行以上操作，对于初始点 u，设沿途卡住 u 的点为 $a_1, a_2, \dots, a_k$，那么答案就是 $\lceil \frac{n-1+k}{2} \rceil$。令 $a = a_k$，那么可以发现 $k = 2\operatorname{dis}(u,a) - c_{\operatorname{dis}(u,a)}$。

再发现一点，就是如果 a 不是真正的 a_k（即 a 处没有被卡住，或者 a 不是最大的 a_k），那么上面的 $k = 2\operatorname{dis}(u,a) - c_{\operatorname{dis}(u,a)}$ 一定会**偏小**（即不合法者一定偏小）！由此知对于一个固定的初始点 u，答案为 $\max\limits_a \left\{2\operatorname{dis}(u,a) - c_{\operatorname{dis}(u,a)}\right\}$。

（上面这一步相当关键，做的时候被卡在这里了，因为一直在想如何按顺序从 a_1 一步步推进到 a_k。只考虑末状态而看淡过程的技巧再次体现。）

进一步地，我们注意到题目求的是所有 u 中答案的**最大值**，所以**整道题的答案即为** $\boldsymbol{\max\limits_u \max\limits_a \left\{2\operatorname{dis}(u,a) - c_{u,\operatorname{dis}(u,a)}\right\}}$。

（其实到这里对于一个固定的起始点 u 可以 $O(n)$ 算答案了，有 $O(n^2)$ 的部分分了）

接下来是优化。一个自然的想法当然还是枚举 u 去找 a，但是会遇到一些困难，因为最优的 a 虽然固定，但是却不显然。

重新审视上面的答案式子就会柳暗花明：由于式子求的是 max 的 max，所以**两个 max 的顺序可以交换**，即**可以枚举 a 去找 u**。这也是一个很厉害的技巧。

此时对于一个 a，最优的 u 可能不止一个，却都已经很明显了：删掉 a 之后图 G 会被分成若干个连通块，当 u 所在的连通块确定时，式子中的 $c_{u,\operatorname{dis}(u,a)}$ 也就确定了，所以此时 u 的最优解即为每个连通块中离 a 最远的点。

此时如果直接枚 a，复杂度不变，但是这个相对简单的形式意味着我们可以尝试去**统一处理多个 u 的答案**。

具体地，如果我们能够确定一个点 v，满足删去 a 之后，u 和 v 不在同一个连通块内，则对于 G 中以 v 为根的任意一棵 dfs 树，有 $\text{dis}(u,a)=\text{dis}(v,u)-\text{dis}(v,a)$，且删去 a 之后 u 所在的连通块大小即为其 dfs 树中的子树大小，可以直接枚举子树判断其 low 值是否 $\le \text{dfn}_a$，若否则取其子树中 dis 最大的点更新答案（注意不是 dfs 树中的深度）。

（这大概是属于多个答案统一计算的思想吧）

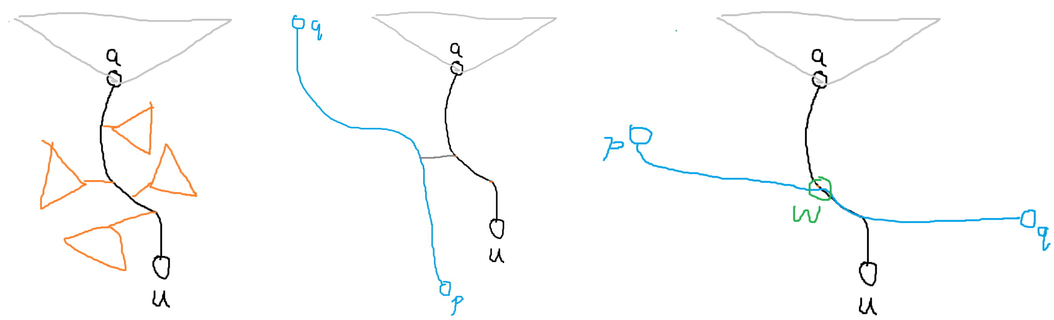
于是我们可以多选几个 v，只要保证一定有某个 v 使得最优解中的 a 能割开 u 和 v 即可。

所以现在问题归结为：找到若干个 v，使得一定有某个 v 使得最优解中的 a 能割开 u 和 v。

重新回去，设 $F(u,a)$ 为选择 u,a 的答案。由于我们刚才屡次提到的点双，所以建圆方树应该是一个自然的想法。

进一步地，我们发现 $F(u,a)$ 有一个上界，就是 u 走到 a 经过的点双数量（因为至多卡住这么多次），对应在圆方树上就是这两个点之间距离的 $1/2$。

优化这个上界，可以变成 $\text{dis}(u,a)/2-x$，其中 x 为路上叉出去的圆点个数，如图 1 所示，x 即橙色子树中的圆点个数：（这里的 dis 指的是圆方树上的距离）



(a) 图 1　　(b) 图 2　　(c) 图 3

于是我们可以理解到这个 $\text{dis}(u,a)$ 应该要很长才能抵掉这一车橙色子树。

树上，很长的链？想到直径！

于是我们猜想取 v 的集合为直径的两个端点 p,q 即可~~（真的这么好猜到吗）~~。下面证明这样取是对的。

用反证法，假设 p,q,u 都在 a 的同一侧。

第一种情况，如果路径 (p,q) 和路径 (u,a) 不相交。应该很显然，(p,q) 这条路径都比 (u,a) 还长了（直径定义），那 $\text{dis}(u,a)\le\text{dis}(p,q)$，而 $x\ge\text{dis}(p,q)/2$，则直接有 $F(u,a)\le 0$，如上面图 2 所示。

第二种情况，如果路径 (p,q) 和路径 (u,a) 相交。此时如上面图 3 所示，令 w 为路径 (u,a) 上靠近 a 的与路径 (p,q) 的交点。不妨设 w 靠近点 p，则由直径的性质我们有

$\text{dis}(w,a) \le \text{dis}(w,p)$，所以把 a 改成 w（若 w 为方点则改成重合部分中下一个圆点，区别不大），损失至多为 $\text{dis}(w,a)/2$，而 x 至少减少 $\text{dis}(w,p)/2$，仔细算算会发现不劣。

所以就证完了。代码很简短，就建出圆方树跑个直径，然后从直径两个端点各 bfs 一遍求 dis 然后再跑一遍 Tarjan 状物算答案即可。

时间复杂度 $O(n+m)$。

整道题中值得学习的三个 trick：

1. 将两个求 max 的顺序交换，可以使得组合意义更加明显，方便做题
2. 当需要对多个初始点求答案的时候，想办法通过一次（或常数次）统一的计算把它们的答案都求出来（up and down 也算是这样的 trick 吧）
3. 树和距离想到直径。