T1

树边+前向边和返祖边数量是等价的，两者取最大即可称为 X 类边，极端情况是链，共 N*(N-1)/2 条

但横叉边与上面俩是互斥的，称为 Y 类边，极端情况是菊花，共(N-1)*(N-2)/2 条

考虑把菊花的一个叶子挪到某个叶子下面，发现 Y 类边少了一条，X 类边多了一条。链类似

于是可以判断无解的情况，即 X 类边+Y 类边>N*(N-1)/2

yy 横叉边有点奇怪，考虑构造 X 类边刚好的方案，那么之后能连得横叉边数量是最多的，一定满足

进一步观察，一个点能贡献 X 类边的数量之和它的深度有关，于是直接先搞条链，最后一个点深度刚好卡好，接下来全都是深度为 1 的叶子即可

有其他构造方案的同学可以上来交流一下

T2

取一次咋做，线段树维护一下

取多次而且不能香蕉咋办

取了一次之后，把线段树上这一段变成相反数

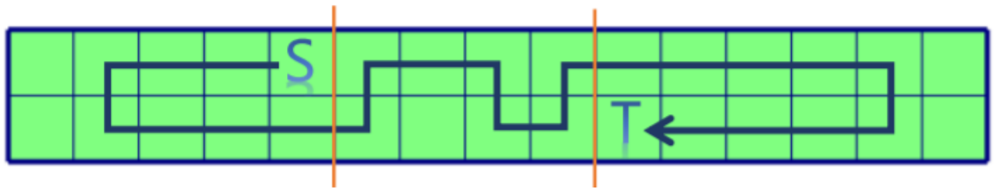
然后再贪心取和最大的。

重复以上操作，发现最后一定有对应的解，且根据贪心过程一定是最大的

线段树上维护区间和最大/小及位置，左/右连续最大/小及位置，取反标记

T3

可以发现不重复经过同一个格子的路径一定是形如这样的:



这个路径可以分成三段:

➢ 从 S 出发向左走一段再回来。

➢ 上下上下地往右走。

➢ 往右走一段再回到 T。

当然, S 和 T 的位置可以调换。

发现这个性质之后就可以直接 DP 了, 左右两段可以用字符串 Hash 做, Left[i][j][k] 表示匹配到第 i 行第 j 列的位置, 匹配了 k 个字符的方案, 那么 Left[i][j][k]的转移就是 Left[i][j][k]=Left[i][j-1][k]+1; Right[i][j][k]表示匹配到第 i 行第 j 列的位置, 匹配了 k 个 字符的方案, 那么 Right[i][j][k]的转移就是 Right[i][j][k]= Right[i][j+1][k]+1。接着中间的一段用简单的 DP 实现, 设 F[i][j][k]表示在第 i 行第 j 列的位置, 匹配到第 k 个字符的方 案, 把三段拼起来就好了。

这题就这么简单, 主要是细节处理上比较麻烦。

总复杂度是 $O(N^2)$。

CF613E