T1 - string

首先要让 t 跟 s 前面相同的部分尽可能多。

如果 s 倒数第 k 位是 1，可以让 t 的最后 k 位和 s 都不一样。这样满足条件。

如果不是的话，为了前面相同部分尽量多，要修改的 1 就是倒数第 k 位之前的第一个了。剩下的位其实就可以任意填了，先把所有零变成一，再尽量少的把一变成零就好了。

T2 - maimai

题目里的一次次出行其实就是进行了一种链剖分。dfs 一下，先算出来所有子树的信息，现在只要决定当前点归属子树里的哪条链就好了。因为要最大化最小值又最小化最大值，所以选最短的那条链一定正确。

T3 - bug

对于 bug 所在的行，写出 dp 方程：$f_i = \min\limits_{0 \le j < i} f_j + (i-j)^4 + a_i$。

观察到可以一直选一个，这样的话 $f_i \le n^2$。所以当 $(i-j)^4 \ge n^2$ 时就不用枚举了，只用枚举 j 的一个后缀即可。最多就是 $O(\sqrt{n})$ 个。写成 std 的样子复杂度也是对的。

T4 - canteen

考虑预处理出每个点最近的电动车距离 d_i，这个可以 bfs 也可以两遍 dfs 算子树里的和子树外的。

接下来就只用考虑路径上的了，如果在 p 位置出走寻找电动车，答案就是 $2dis(x,p) + 3d_p + dis(p,y)$。这个东西可以直接数据结构维护了。

刚开始写的是树剖加线段树，不过后来发现可以树上倍增。找到 lca 之后对两边分别求就好了，维护的就是一个 $3d_i + deep_i$ 之类的最小值。