题目来源：「CF1381」The Majestic Brown Tree Snake

设少女们的阵型的长度为 L（边的数量）。

记 p 为关键点，当且仅当从 p 出发有至少 3 条 边不相交 且 长度 $\geq L$ 的链（记为 p 的分支）。

显然，如果阵型的两端能到达某个关键点，阵型就能翻转。

1. 如果两端能到达某个关键点，则两端就能到达所有关键点。
2. 如果两端不能到达某个关键点，阵型就不能翻转。

结论 1 证明

设两个关键点分别为 p_1 和 p_2，阵型的一端为 p_1。

由定义可知 p_1 至少有一个分支不为阵型所在的分支，且不在 $p_1 \to p_2$ 的路径上，此时把阵型移动到这个分支中，再随着 $p_1 \to p_2$ 的路径移动就可以到达 p_2。

结论 2 证明

如果阵型不能到达直径（非两端的某个点也算），就可以把这条直径删去。

因此我们可以归纳成更小的树来证明。

此时阵型一定能到达直径，并且离不开直径。

反证法：如果阵型能离开直径（$s \to t$），记离开的那个点为 u，这说明 $dis(s,u) \geq L$ 且 $dis(t,u) \geq L$，又因为阵型在 u 的第三个分支上，所以 u 是关键点，与条件矛盾，所以原命题成立。

同时说明阵型中一定有至少一条边永远在直径上，这条边决定了阵型的方向，所以阵型不能翻转。

算法一

只判断是否存在关键点。

期望得分 20 分。

算法二

以某个关键点为根，交替使两个端点移动到子树中深度最深的那个结点，如果某时刻两端有祖先关系，说明可以到达关键点。

如果相邻的两次移动后两个端点的位置不变，说明不可能再扩展了，所以移动次数不会超过 n 次。

每次移动时可以用倍增实现，时间复杂度 $\mathcal{O}(n\log_2 n)$。

期望得分 100 分。

算法三

和算法二类似，以某个端点 s 为根，维护 s 和 t 可以向下/向上移动到的位置，如果 s 和 t 能移动到的位置有交集，说明答案为 `Yes`。

具体实现时，用双指针维护 s 和 t 分别能到达位置，比如 t 能移动到新的位置，就更新这个新的位置对 s 的贡献，时间复杂度 $\mathcal{O}(n)$。

期望得分 100 分。