

车窗外 这夜色 流光溢彩

首先如果你要走向 1 节点就是每条有用的边都要走两遍。有用就是子树内有关键点。这个答案叫做 ans 。

然后你不用走回去你只要找一个点停了就行。找到深度最大的点 $mxdep$ 。

最后答案就是 $ans - mxdep$ 。

时间复杂度 $\mathcal{O}(n)$ 。

别忘了 闭上眼 才算醒来

不妨假设 $a_i < a_j$ ，固定 i ，那么合法的 j 一定是 i 之后第一个值比 i 大的。也就是能找到 j 等价于 $[i + 1, n]$ 最大值比 a_i 大。

然后 $a_i > a_j$ 也是一样的，枚举 j 即可。

取模就是诈骗。

时间复杂度 $\mathcal{O}(n)$ 。

你参演 这场戏 变换姿态

首先最大的 m 是很简单的。按 r 排序能选就选即可。

然后要求值最大就相当于从大到小能选就选。当然这个能选指的是选了之后答案不会变小。

选了一场演出相当于把一段大区间分成了两段小区间。然后能选就相当于两端小区间的答案 +1 和原来的大区间相等。

于是我们只要求一段区间内最多选多少场演出就行了。

怎么求 $[L, R]$ 内有多少演出：按 r 排序，找到第一场 $l \geq L$ 的演出，之后能选就选直到 $> R$ 。

暴力实现就是 $\mathcal{O}(n^2)$ 。

优化这个东西，选了一场演出之后下一场选的演出也确定了。这个是一个树形结构。使用倍增优化这个往上跳的过程即可。时间复杂度 $\mathcal{O}(n \log n)$ 。

谜底 结局 我该怎么猜

怎么求一个序列 a_1, \dots, a_n 经过 x 轮冒泡后的结果？

首先不难观察出后 x 个肯定是前 x 大的值，于是秒了子任务 3，也就是前 v 小的数的和。

考虑 a'_1 是 a_1, \dots, a_{x+1} 中的最小值， a'_2 是 a_1, \dots, a_{x+2} 中去掉 a'_1 后的最小值， a'_i 是 $a_1, \dots, a_{\min(n, i+x)}$ 中去掉 a'_1, \dots, a'_{i-1} 的最小值。

所以在子任务 4 中， $l = 1, r = n, x, u = 1, v$ 的答案和 $l' = 1, r' = \min(n, v + x), x' = x, u' = 1, v' = v$ 的答案是一样的，也就是直接求出 $[l', r']$ 内前 v 小的数的和就行了。

然后 $u \neq 1$ 差分成两次询问。问题变成查询一个区间前 v 小的数的和。这个用可持久化线段树就能做了。时间复杂度 $\mathcal{O}(n \log n)$ 。