

题解

n 条线段, 每条 $[l_i, r_i]$, 每个点被覆盖 k 次

卖瓜

- 一个简单的贪心是将线段按照右端点升序排序, 能选则选。
- 用线段树维护每个点被覆盖的次数即可。
- 如何证明贪心的正确性?

卖瓜

- 考虑调整，每次选择之后“左端点最ⁿ大”的进行调整即可。

假设给出的区间为 a_1, a_2, \dots, a_m

策略 $S = \{a_{s_1}, a_{s_2}, a_{s_3}, \dots, a_{s_k}\} \quad |S| = k$

下一个最优解 $S' = \{a_{s'_1}, a_{s'_2}, \dots, a_{s'_c}\} \quad |S'| = c$ 证明 $S' \rightarrow S$

假设 a_i 是 $a_i \in S'$ 且 a_i 是 S 中左端点最小的。

a_j 是 $a_j \in S$ 且 a_j 是 S' 中左端点最小的。

一定可以将 S' 中 a_i 替换为 a_j

$$R_{a_i} \geq R_{a_j}$$

$$\textcircled{1} L_{a_j} \geq L_{a_i} \quad \underbrace{\begin{matrix} a_j \\ \uparrow \\ a_i \end{matrix}}_{(V)}$$

$$\textcircled{2} L_{a_j} < L_{a_i} \quad \underbrace{\begin{matrix} a_j \\ \uparrow \\ a_i \end{matrix}}_{(V)}$$

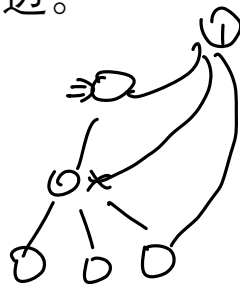
给定一棵树.

卖外卖

求至少加多少条边使 1 号点到所有点距离 ≤ 2

$\Leftrightarrow \forall x, \textcircled{1} - x \text{ or } \textcircled{1} - \text{fa}_x \text{ or } \textcircled{1} - \text{Son}_x$

- 容易发现对于一个点来说要么自己与 1 号点连边, 要么父亲与 1 号点连边, 要么有一个儿子与 1 号点连边。
- 于是记录一个 $f_{i,0/1/2}$ 树形 DP 即可。



根节点

$f_{i,0}$ \leftarrow i 与 1 号点连边, 且 i 未与 1 号点连边

$f_{i,1}$ \leftarrow i 与 1 号点连边, 且 i 未与 1 号点连边

$f_{i,2}$ \leftarrow i 与 1 号点连边

$f_{x,0} \rightarrow f_{\text{fa}_x,2}$

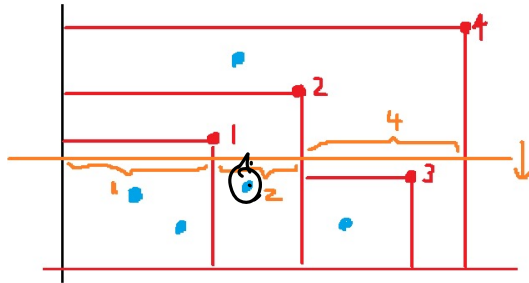
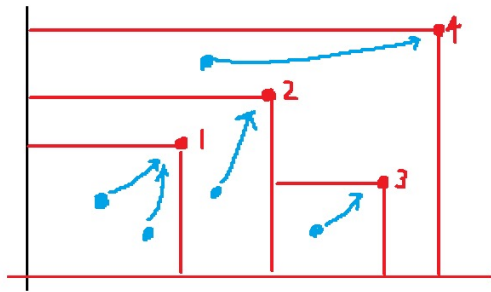
$f_{x,1} \rightarrow f_{\text{fa}_x,0} \text{ 或 } f_{\text{fa}_x,2} \text{ 或 } \underline{f_{\text{fa}_x,1}}$

$f_{x,2} \rightarrow \text{all}$

卖夹心糖

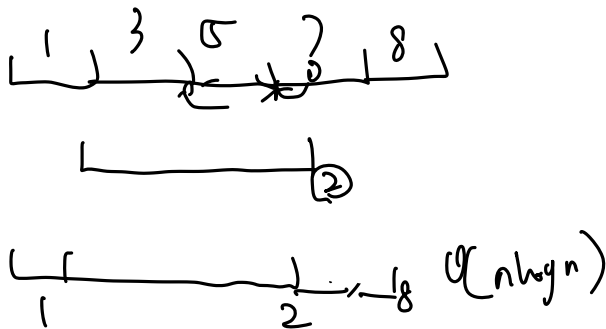
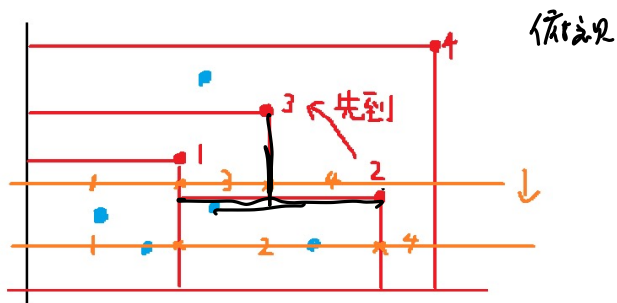
- 首先看看中间 30% 的部分分 (x' 递增)
- 必定不会出现一个夹心糖被多次看到的情况，因此想办法求每个夹心糖被谁占了就行
- 从上至下使用扫描线算法，用一个 set 记录当前每一段属于哪个区域，每次下移经过一个红点都产生一个新的区域

红点与蓝点一起



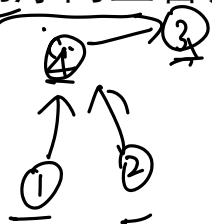
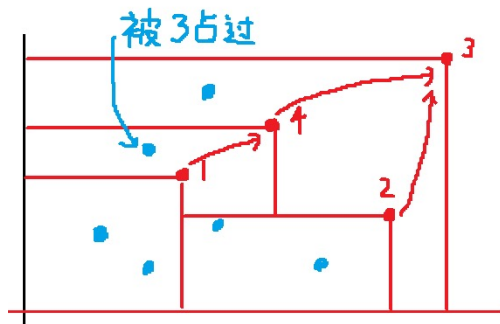
卖夹心糖

- 使用扫描线同样可以对 100% 的情况计算每个蓝点是被哪个红点直接控制的
- 需要注意的是这里可能出现下面的红点比上面先到，导致扫描线过程中区域变少的情况

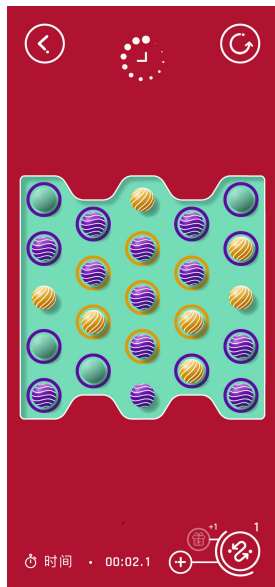


卖夹心糖

- 同样的，对红点也求直接控制他的红点，会形成一个树形结构
- 可以发现，如果一个子节点比父节点后到，那么自己占领的蓝点之前是被父节点占领过的，并且可以一直向上溯源
- 那么使用一个并查集，按时间倒序向上合并就可以求出真正的答案

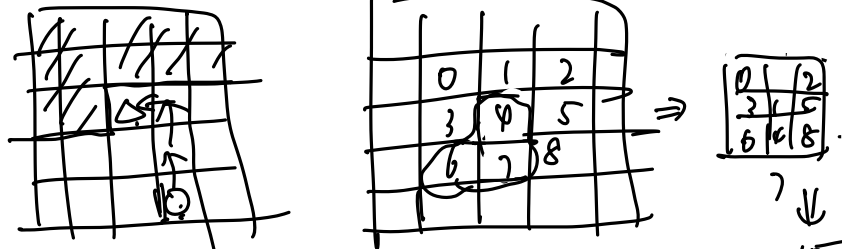


卖棋盘



Dotcraft

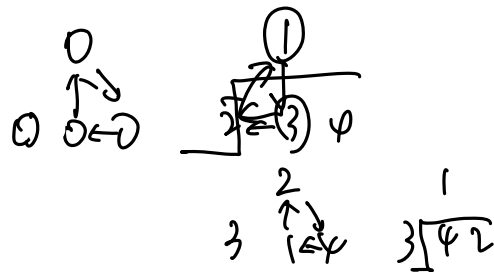
卖棋盘



- 我们一行一行地去填数，先考虑在前 $i - 1$ 行填好的情况下如何填第 i 行。由于一些特殊情况，我们先不考虑最后一行怎么填。
- 考虑对于一个位置 (i, j) 我们怎么将 $(i - 1)n + j$ 移到 (i, j) 上。
- 假设现在这个数在 (x, y) 处，我们考虑先向上移再横向移动。
- 首先，如果在同一行，我们可以通过 $D(y), R(x+1), U(y), L(x+1)$ 将其向左移动一格（且不改变其他上面的格子）
- 如果不在同一行，我们可以通过 $R(x), D(j), L(x), U(j)$ 将其向上移动一格（不改变其他上面的格子）



卖棋盘



- 现在只剩下最后一行还没有排好序了
- 考虑对于一个位置 x ，我们可以通过 $R(n-1); D(x); L(n-1); U(x)$ 使得以 x 为右下角的三角形的三个格子逆时针循环移位一格，同样的也可以构造出将其顺时针移位一格。于是将两种方法重叠可以让它变成最后一行的相邻三个的循环位移。
- 最后至多剩下两个位置不能处理，用一次 S 操作就可以了。

