

《封印》命题报告

题目大意

你是一名大魔法师，现在遇到了 n 只怪物，第 i 只怪物的出现时间为 $[l_i, r_i)$ ，有经验值 w_i 。对于怪物 i ，你可以选择一个实数 $k_i \in [l_i, r_i]$ ，并在 $[l_i, k_i)$ 时间内施展封印术控制这只怪物。特别地，如果 $k_i = l_i$ ，表示你没有对这种怪物施展封印术。由于人是有限的，在同一时刻，你最多对 K 个怪物施展法术， K 为给定常数。

由于你已经很久没有使用过封印术了，在 0 时刻你的熟练度为 0。对于怪物 i ，如果 $k_i = r_i$ ，那么你就成功封印了这只怪物，所以在 r_i 时刻你的熟练度就会增加 w_i ；如果 $k_i < r_i$ ，那么怪物就会在 k_i 时刻攻击你，使得熟练度重置为 0。

在任意时刻，设此时熟练度为 W ，你可以选择施展终极秘术，将时间线上的所有的 n 只怪物变成 W 枚金币，并带着它们离开。如果同一时刻发生多个事件（熟练度增加、熟练度重置、终极秘术），它们之间的生效顺序可以任意安排。

现在，请求出你最多能带着多少枚金币离开。

数据范围

对于所有数据， n, l_i, r_i, w_i, K 均为正整数， $1 \leq K \leq n \leq 3 \times 10^5, 1 \leq w_i \leq 10^9, 1 \leq l_i < r_i \leq 2n$ ，且保证 $l_1, l_2, \dots, l_n, r_1, r_2, \dots, r_n$ 构成 $1 \sim 2n$ 的排列。

各子任务特殊约束及分值如下：

子任务编号	$n \leq$	特殊性质	分值
1	20	-	5
2	2500	$w_i = 1$	15
3	3×10^5	$w_i = 1$	20
4	2500	-	15
5	10^5	$K \leq 30$	20
6	3×10^5	-	25

解题过程

算法一

下文“区间 i ”指怪物 i 或者区间 $[l_i, r_i)$ ，称 w_i 为区间的权值。

首先考虑指数级算法。枚举集合 S 表示最后权值计入答案的区间。设 $L = \min_{i \in S} \{r_i\}, R = \max_{i \in S} \{r_i\}$ 。为了使 S 合法，所有 $L < l_i < R$ 的区间在 $[L, R]$ 内的部分均需被封印，且所有 S 中的区间均需被完整封印，其余区间均可被完全放弃（即在 l_i 被放弃）。

进一步地，可以认为 $L < l_i < R < r_i$ 的区间也被完整封印，因为如果大于 R 的时刻出现超过 K 个这种区间，那么它们也会使 $R - 1$ 时刻不合法。

检查是否存在时刻被覆盖超过 K 次即可。

时间复杂度 $O(2^n n)$, 期望通过子任务 1。

算法二

先考虑所有区间权值均为 1 的情况。

上面的朴素算法提供了思路, 最后计入答案的区间集合数量是指数级的, 但集合对应的 $[L, R]$ 很少。

枚举 $[L, R]$, 区间可被分为六类:

1. I 类: $l_i \leq L \leq r_i \leq R$;
2. II 类: $l_i \leq L \leq R < r_i$;
3. III 类: $L < l_i < r_i \leq R$;
4. IV 类: $L < l_i < R < r_i$;
5. V 类: $l_i \geq R$;
6. VI 类: $r_i < L$ 。

其中, III, IV 类区间一定被完整封印, 同时 III 类区间必须计入答案。除此之外, 可以完整封印一些 I 类区间以使答案更大。

定义 $[L, R]$ 合法当且仅当所有 III, IV 类区间没有使得一个时刻被覆盖超过 K 次, 容易发现对于固定的 L , 合法的 R 一定是 L 开始的若干连续正整数。进一步, 如果 $[L, R_1]$ 和 $[L, R_2]$ 均合法且 $R_1 < R_2$, 那么 $[L, R_2]$ 对应答案一定不劣。证明如下: 在 $[L, R_1]$ 最优答案中的 I 类区间集合 S , 那么 $[L, R_2]$ 相对于 $[L, R_1]$ 多出的 III, IV 类区间都有 $l_i \geq R_1$, 所以它们和 S 中的区间都不交, 故在 $[L, R_2]$ 时选择 S 中的所有区间也合法。

现在要考虑的 $[L, R]$ 只有 $O(n)$ 个, 注意到 R 随 L 增大单调不减, 故可以用尺取法加上线段树来 $O(n \log n)$ 求出这些 $[L, R]$ 。

对于每个 $[L, R]$, 显然取的 I 类区间按 r_i 排序后一定是一个前缀。所以对于这 $O(n)$ 个 $[L, R]$, 分别将对应的 I 类区间按 r_i 从小到大尝试加入即可。

时间复杂度 $O(n^2 \log n)$, 期望通过子任务 2。

算法三

算法二的瓶颈在于每次重新尝试加入 I 类区间, 现在考虑在尺取过程中维护。尺取可以拆成 L 右移和 R 右移, 下面分开考虑会对区间类别产生什么影响。

L 右移: (a) I \rightarrow VI; (b) III \rightarrow I; (c) IV \rightarrow II。

R 右移: (d) II \rightarrow I; (e) IV \rightarrow III; (f) V \rightarrow IV。

记答案中选取的 I 类区间集合为 S , 所有 III, IV 类区间集合为 T 。

(a) 中如果被移除的 I 类区间如果在 S 中, 移除后显然最多再加入一个 I 类区间到 S 中。用小根堆维护所有不在 S 中的 I 类区间, 尝试加入 r_i 最小的那个即可。

(b)(c) 都可以看作是先删除了一个 $l_i = L$ 的 T 中区间。这其实和删除一个 S 中的区间产生的影响类似, 可以说明此时也只会再加入一个区间到 S 中。对于 (b), 还要再加入一个 I 类区间。此时如果它的 r_i 比 S 中最大的 r 小, 将它加入 S , 然后若产生不合法则删去 S 中 r 最大的区间, 这部分可以用大根堆维护 S 中区间; 否则加入小根堆, 并尝试将堆顶加入 S 。

(d) 由于这个发生改变的区间满足 $r_i = R$, 所以不会将 S 中的区间“顶替掉”。只要把它加入小根堆, 并尝试加入当前 r_i 最小的 I 类区间即可。

(e) 不会使 S, T 产生任何改变只需要把这个区间权值计入答案即可。

(f) 会向 T 中加入一个区间，上面算法二已经说明了不会影响 S 。

以上单步维护均为 $O(\log n)$ ，总时间复杂度 $O(n \log n)$ ，期望通过子任务 2,3。

算法四

现在考虑 w_i 没有特殊限制的情况，尝试扩展算法二。

尺取的正确性显然，区别在于现在不能按照 r_i 排序了。一个自然的想法是按照 w_i 从大到小贪心加入，下面证明这是对的。

固定 $[L, R]$ ，设 I 类区间为 $(l_1, r_1, w_1), (l_2, r_2, w_2), \dots, (l_m, r_m, w_m) (w_1 \geq w_2 \geq \dots \geq w_m)$ 。注意到这个结论和 Kruskal 最小生成树算法很类似（事实上，这本质上是因为二者都是拟阵），考虑类似的归纳证明：对于 $1 \leq k \leq m$ ，贪心加入前 k 个区间后得到的集合 A_k 一定是某组最优解 S_k 的子集。

1. $k = 0$ 时 $A_k = \emptyset$ 显然成立；
2. 假设 $k = i$ 时成立，考虑向 A_i 中加入 $(l_{i+1}, r_{i+1}, w_{i+1})$ 得到 A_{i+1} ，如果加入失败或加入成功且这个区间在 S_i 中，令 $S_{i+1} = S_i$ 即可说明此时命题成立。否则，考虑将第 $i + 1$ 个区间加入 S_i 得到 S_{i+1} ，此时 S_{i+1} 必然不合法。由于只保留 S_{i+1} 中编号不超过 $i + 1$ 的区间合法（这是因为 $i + 1$ 在贪心中成功加入了），而且此时被覆盖次数最多的时刻一定被覆盖恰好 $K + 1$ 次，所以一定可以通过删除 S_{i+1} 中恰好一个编号大于 $i + 1$ 的区间，使得 S_{i+1} 合法。由删除的区间的权值不超过 w_{i+1} 知删除后 S_{i+1} 的权值和不会比 S_i 小，故 S_{i+1} 也为一组最优解。

有了这个贪心，就可以类似算法二做到 $O(n^2 \log n)$ ，期望通过子任务 1,2,4。

这个贪心也可以说明在 $w_i = 1$ 时，将 I 类区间以任意顺序尝试加入都是正确的。进一步，这也说明所有权值和最大的合法 S 的区间个数相同且是所有合法方案中区间个数的最大值。

算法五

考虑 K 比较小的情况，尝试扩展算法四。

注意到所有 I 区间，都覆盖 L 或者右端点与 L 重合，故算法四中计入答案的 I 类区间只有 $O(K)$ 个。如果能找出这些区间的同时不遍历其它区间，就可以得到更优的复杂度。

设 lim 为最小的被覆盖了 K 次的时刻，容易发现此时可以加入的 I 类区间即为所有 $r_i \leq lim$ 的 I 类区间，这些区间中 w_i 最大的即为下一个加入答案的区间。

lim 和所有 I 类区间容易用线段树维护，时间复杂度 $O(nK \log n)$ ，期望通过子任务 1,2,4,5。

算法六

尝试结合算法三和算法四。还是考虑算法三中提到的六种情况。不妨设所有区间的权值互不相同。

(a) 即从 S 中移除一个区间 i 。设 j 为 w_j 最大的能加入 $S \setminus \{i\}$ 的区间（显然 $w_j < w_i$ ），下面说明 $S \setminus \{i\} \cup \{j\}$ 即为新的最优解：

考虑对比删去 i 前后上面的贪心过程：

1. 权值大于 w_i 的区间是否加入没有变化；
2. 对于权值在 (w_j, w_i) 中的区间，假设此时原来不在 S 中的区间 k 可以加入，设 S 中所有权值小于 w_j 的区间集合为 A ，那么由于 $j, k \notin S$ ，所以 $\forall a \in A, r_a < r_j, r_a < r_k$ ，那么 j, k 对 A 中区间的限制等效，故 k 也可加入 $S \setminus \{i\}$ ，与 j 的最大性矛盾；

3. 对于权值小于 w_j 的区间, 由于删去的 i 满足 $r_i = L - 1$ 是现在所有 I 类区间中最小的, 故将 i 替换成 j 后限制更严, 但由于 j 能加入 $S \setminus \{i\}$, 所以这部分区间是否加入也没有变化。

为了找到 j 可以先找到最小的 lim 满足时刻 lim 被覆盖了 K 次, 那么 j 即为 $r_j \leq lim$ 且 w_j 最大的 j , 容易用线段树维护 (同算法五)。

(b)(c) 看作是先删除了一个 $l_i = L$ 的 T 中区间。这其实和删除一个 S 中的区间产生的影响类似 (可以看作删除 S 中一个权值无穷大的区间), 具体同上。对于 (b), 还要再加入一个 I 类区间。设 j 为 w_j 最小的从 $S \cup \{i\}$ 删除后合法的区间 (显然 $w_j \leq w_i$), 那么下面类似 (a) 中说明 $S \cup \{i\} \setminus \{j\}$ 即为新的最优解:

考虑对比加入 i 前后上面的贪心过程:

1. 权值大于 w_i 的区间是否加入没有变化;
2. 如果 $j \neq i$, 则 w_j 成功加入;
3. 对于权值在 (w_j, w_i) 中的区间, 原来在 S 中的由 $S \cup \{i\} \setminus \{j\}$ 合法知它们仍然会加入, 原来不在 S 中的现在在前面多了区间 i , 更不可能加入;
4. 对于区间 j , 由最小性知每一个 $k \in S$ 且 $w_k < w_j$ 都要满足 $r_k < r_j$, 且设 $mx = \max_{k \in S \wedge w_k < w_j} \{r_k\}$, 那么如果所选 I 类区间为 $S \cup \{i\}$, 则 $[mx, r_j)$ 中有一个时刻被覆盖 $K + 1$ 次, 故可知此时 j 不会加入;
5. 对于权值小于 w_j 的区间, 其中原来不在 S 中的 k , 由 j 最小性知 $r_k < r_j, r_k < r_i$, 否则 $S \cup \{i\} \setminus \{k\}$ 合法。所以 i, j 对 k 的限制等效, 故这样的 k 肯定也不会加入。对于原来在 S 中的 k , 由 $S \cup \{i\} \setminus \{j\}$ 合法知它们仍然会加入。

找 j 同样可以先找到最大的 lim 满足时刻 lim 被覆盖了 $K + 1$ 次, 那么 j 即为 $r_j > lim$ 且 w_j 最小的 j , 容易用线段树维护 (类似算法五)。

(d) 同 (b) 的加入。

(e) 和算法三中没有区别。

(f) 和算法三中没有区别。

所有线段树操作均为 $O(\log n)$, 每次 L, R 移动需要 $O(1)$ 次线段树操作, 总复杂度 $O(n \log n)$, 期望通过所有子任务。

参考资料

这是一道原创题, 没有参考资料。

致谢

感谢卢暮野、董向恒、夏敬和同学参与本题的讨论和验题工作。