

NOIP 2019 模拟赛 Solution

diamond_duke

题目名称	序列	积木	保镖
可执行文件名	seq	brick	guard
输入文件名	标准输入	标准输入	标准输入
输出文件名	标准输出	标准输出	标准输出
时间限制	1s	1s	1s
内存限制	512MB	512MB	512MB
子任务个数	4	5	6
题目类型	传统型	传统型	传统型

请注意： 评测时开启 O2 优化和 C++11 编译选项，栈空间限制同空间限制。

1 序列

考虑每次求出并集中的下一个元素，则重复这一过程 n 次即可得到答案。

注意到序列中的下一个元素一定是某个序列没有被访问过的元素中，最小的那个。

因此我们可以使用一个小根堆维护目前所有序列的最小值，则每次取出堆顶元素即为并集的下一个元素。在取出堆顶元素后，我们需要将这个序列的最小值更新为下一个元素，计算后将其加入堆即可。

时间复杂度： $\Theta(mk + nk \log_2 m)$ 。

2 积木

注意到在任意时刻最下一行的砖块一定不会新掉落，因此我们可以按照从下到上的顺序依次考虑每一行，并求出这一行中哪些砖块掉落了。

考虑如何求出这一结果。注意到一段位于 $[l, r]$ 的连续砖块段会一起掉落，当且仅当下面一行的 $[l - 1, r + 1]$ 的连续砖块也一起掉落了。

那么如果我们使用 `std::set` 维护每一行掉落的砖块连续段，即可得到这一行的答案。同时更新 `std::set` 中的内容即可继续求得上一行的答案了。

考虑计算这一过程的时间复杂度。注意到每上升一行，`std::set` 中所有区间的长度都会减小一个砖块。注意到总的砖块数为 n 。因此，只要跳过 `set` 为空的情况，即可做到总遍历次数为 $\Theta(n)$ 级别。

时间复杂度： $\Theta(n \log_2 n)$ 。

3 保镖

考虑合法等价于什么。考虑所有选出的人，他们的总工作时间为 $\sum a$ ，而每个保镖的工作周期为每 $(a + b)$ 小时可以工作 a 小时。

因此，只要所有选出的保镖均满足 $a + b \leq \sum a$ ，那么我们可以以 $\sum a$ 为周期依次排班，则下一次轮到同样的保镖时间隔了 $\sum a - a \geq b$ ，因此这一定是一种合法的方案。

否则，我们考虑使用这个不满足条件的保镖的某种方案，不妨设他是第一个值班的。我们考虑从他第一次值班到第二次值班中间那一段时间，则因为这段时间长度和超过 $\sum a - a$ ，故某个保镖一定被使用了至少两次，故这不是一种合法方案。

因此，合法方案一定满足条件 $\max\{a + b\} \leq \sum a$ 。

考虑枚举 $(a + b)$ 的最大值。则此时问题变为选出尽量少的保镖，使得 $\sum a \geq \max\{a + b\}$ 。于是我们在枚举的过程中使用线段树维护每个 a 的出现次数，然后线段树上二分即可得到这一最小值。最后将所有最小值取最小值即可得到答案了。

时间复杂度： $\Theta(n \log_2 n)$ 。