2019/10/05 CSP 模拟赛题解

lcy

NFLS

2019/10/05

第一档部分分 $n \leq 10$

- 考虑全排列. 暴力枚举每一个可能的排列，判断是否合法.

第一档部分分 $n \le 10$

- 考虑全排列. 暴力枚举每一个可能的排列, 判断是否合法.
- 时间复杂度为 $O(n!)$.

第二档部分分 $n \le 20$

- 考虑状压 DP.

第二档部分分 $n \leq 20$

- 考虑状压 DP.
- 从左到右依次考虑每个位置上的数字，并且用二进制维护每一个数字有没有出现过，记录方案数.

第二档部分分 $n \le 20$

- 考虑状压 DP.
- 从左到右依次考虑每个位置上的数字，并且用二进制维护每一个数字有没有出现过，记录方案数.
- 时间复杂度为 $O(2^n n)$.

第三档部分分 $n \leq 2 \times 10^7$

- 我们发现，答案实际上是斐波那契数列 -1.

第三档部分分 $n \le 2 \times 10^7$

- 我们发现，答案实际上是斐波那契数列 -1.
- 证明：假设没有排列与原来的完全相同的限制，设 f_n 为 $1 \sim n$ 的方案数.
  - $p_n = n$，那么前 $n-1$ 个的方案数为 f_{n-1}.
  - $p_n = n-1$，那么必然 $p_{n-1} = p_n$，前 $n-2$ 个的方案数为 f_{n-2}.

第三档部分分 $n \le 2 \times 10^7$

- 我们发现，答案实际上是斐波那契数列 -1.
- 证明：假设没有排列与原来的完全相同的限制，设 f_n 为 $1 \sim n$ 的方案数.
  - $p_n = n$，那么前 $n-1$ 个的方案数为 f_{n-1}.
  - $p_n = n-1$，那么必然 $p_{n-1} = p_n$，前 $n-2$ 个的方案数为 f_{n-2}.
  - 综上，$f_n = f_{n-1} + f_{n-2}$.

第三档部分分 $n \le 2 \times 10^7$

- 我们发现，答案实际上是斐波那契数列 -1.
- 证明：假设没有排列与原来的完全相同的限制，设 f_n 为 $1 \sim n$ 的方案数.
  - $p_n = n$，那么前 $n-1$ 个的方案数为 f_{n-1}.
  - $p_n = n-1$，那么必然 $p_{n-1} = p_n$，前 $n-2$ 个的方案数为 f_{n-2}.
  - 综上，$f_n = f_{n-1} + f_{n-2}$.
- 斐波那契数列的第 n 项可以 $O(n)$ 计算. 时间复杂度为 $O(n)$.

满分

- 斐波那契数列的计算可以用矩阵乘法加速. 时间复杂度为 $O(\log n)$.

满分

- 斐波那契数列的计算可以用矩阵乘法加速. 时间复杂度为 $O(\log n)$.
- 还可以用特征方程解出斐波那契数列的通项公式, 再用 Cipolla 算法或者暴力枚举算出 5 在模 19491001 意义下的二次剩余的方法, 但是所需技巧应该是省选或者更高难度级别的.

Observation 1

- 首先，我们需要知道我们要采取什么样的策略.

Observation 1

- 首先，我们需要知道我们要采取什么样的策略.
- 我们现在只考虑 $m = 1$ 的情况.

Observation 1

- 首先，我们需要知道我们要采取什么样的策略.
- 我们现在只考虑 $m = 1$ 的情况.
- 我们考虑设置一个阈值 x. 代表一个假人至多打 x 下. 如果打完了没有倒下，那么就不再打他了.

Observation 1

- 首先，我们需要知道我们要采取什么样的策略.
- 我们现在只考虑 $m = 1$ 的情况.
- 我们考虑设置一个阈值 x. 代表一个假人至多打 x 下. 如果打完了没有倒下，那么就不再打他了.
- 最坏情况就是不断遇到最大的假人. 设有 y 个假人血量 $> x$, 答案就是 $x \cdot (y + 1)$（这里，很明显，我们可以将这个阈值设为一个假人的血量）.

Observation 1

- 首先，我们需要知道我们要采取什么样的策略.
- 我们现在只考虑 $m = 1$ 的情况.
- 我们考虑设置一个阈值 x. 代表一个假人至多打 x 下. 如果打完了没有倒下，那么就不再打他了.
- 最坏情况就是不断遇到最大的假人. 设有 y 个假人血量 $> x$，答案就是 $x \cdot (y + 1)$（这里，很明显，我们可以将这个阈值设为一个假人的血量）.
- $m = 1$ 的情况下，我们可以枚举这个阈值，然后算答案，取最小的即可. 除去排序时间复杂度为 $O(n)$.

Observation 2

- 我们接下来考虑 $m > 1$ 的情况.

Observation 2

- 我们接下来考虑 $m > 1$ 的情况.
- 我们先将所有的假人按血量从大到小排序.

Observation 2

- 我们接下来考虑 $m > 1$ 的情况.
- 我们先将所有的假人按血量从大到小排序.
- 我们将所有的假人分成 m 组，每组打倒 1 个人，给每个组独立设置一个阈值. 对应的策略，就是不断将阈值最小的组打掉，但是同时会连带打不是这个组，所以接下来要打这个组的阈值 − 已经打掉的阈值，可以发现敲的次数还是这个阈值，没有变化.

Observation 2

- 我们接下来考虑 $m > 1$ 的情况.
- 我们先将所有的假人按血量从大到小排序.
- 我们将所有的假人分成 m 组，每组打倒 1 个人，给每个组独立设置一个阈值. 对应的策略，就是不断将阈值最小的组打掉，但是同时会连带打不是这个组，所以接下来要打这个组的阈值 − 已经打掉的阈值，可以发现敲的次数还是这个阈值，没有变化.
- 可以证明，这些组肯定是若干个连续段. 否则答案一定不会更优（考虑一个组的血量全部大于另外一个组，那么阈值也大于另外一个组，交换一对假人会带来另一个组的阈值的额外代价，所以肯定不优）.

Observation 2

- 我们接下来考虑 $m > 1$ 的情况.
- 我们先将所有的假人按血量从大到小排序.
- 我们将所有的假人分成 m 组，每组打倒 1 个人，给每个组独立设置一个阈值. 对应的策略，就是不断将阈值最小的组打掉，但是同时会连带打不是这个组，所以接下来要打这个组的阈值 − 已经打掉的阈值，可以发现敲的次数还是这个阈值，没有变化.
- 可以证明，这些组肯定是若干个连续段. 否则答案一定不会更优（考虑一个组的血量全部大于另外一个组，那么阈值也大于另外一个组，交换一对假人会带来另一个组的阈值的额外代价，所以肯定不优）.
- 因此，我们可以暴力枚举哪些假人作为阈值，然后剩下来的假人的阈值也就随之确定了，就是恰好比它小且作为阈值的那位假人的阈值. 可以 2^n 或者 $\binom{n}{m}$ 枚举.

DP

- 我们现在考虑 DP.

DP

- 我们现在考虑 DP.
- 设 $dp[i][j]$ 表示考虑了第 $i \sim n$ 位的假人（提醒一下，这里的假人是按血量从大到小排序的）打倒了 m 位假人所需要的最小次数.

DP

- 我们现在考虑 DP.
- 设 $dp[i][j]$ 表示考虑了第 $i \sim n$ 位的假人（提醒一下，这里的假人是按血量从大到小排序的）打倒了 m 位假人所需要的最小次数.
- 枚举这一组的阈值，可以得到如下的 DP 转移方程:

$$dp[i][j] = \min\{a_k(k - i + 1) + dp[k+1][j-1] \mid i \le k \le n\}.$$

DP

- 我们现在考虑 DP.
- 设 $dp[i][j]$ 表示考虑了第 $i \sim n$ 位的假人（提醒一下，这里的假人是按血量从大到小排序的）打倒了 m 位假人所需要的最小次数.
- 枚举这一组的阈值，可以得到如下的 DP 转移方程:

$$dp[i][j] = \min\{a_k(k-i+1) + dp[k+1][j-1] \mid i \le k \le n\}.$$

- 很明显这是一个 $O(n^3)$ DP.

DP 续

- 接下来我们考虑斜率优化.

DP 续

- 接下来我们考虑斜率优化.
- 对于当前的 $dp[i][j]$，假设 k 比 k' 优（假设 $k < k'$），可以得到

$$a_k(k-i+1)+dp[k+1][j-1] < a_{k'}(k'-i+1)+dp[k'+1][j-1].$$

展开，移项，化简得到

$$\frac{(a_k k + dp[k+1][j-1]) - (a_{k'} k' + dp[k'+1][j-1])}{a_k - a_{k'}} > i-1.$$

DP 续

- 接下来我们考虑斜率优化.
- 对于当前的 $dp[i][j]$，假设 k 比 k' 优（假设 $k < k'$），可以得到

  $$a_k(k-i+1)+dp[k+1][j-1] < a_{k'}(k'-i+1)+dp[k'+1][j-1].$$

  展开，移项，化简得到

  $$\frac{(a_k k + dp[k+1][j-1]) - (a_{k'}k' + dp[k'+1][j-1])}{a_k - a_{k'}} > i-1.$$

- 可以维护凸包，然后在凸包上二分，可以做到 $O(nm \log n)$. 可以通过全部测试数据. 也可以进一步用双指针去掉 $\log n$.

DP 续

- 接下来我们考虑斜率优化.
- 对于当前的 $dp[i][j]$，假设 k 比 k' 优（假设 $k < k'$），可以得到

$$a_k(k-i+1)+dp[k+1][j-1] < a_{k'}(k'-i+1)+dp[k'+1][j-1].$$

展开，移项，化简得到

$$\frac{(a_k k + dp[k+1][j-1]) - (a_{k'} k' + dp[k'+1][j-1])}{a_k - a_{k'}} > i-1.$$

- 可以维护凸包，然后在凸包上二分，可以做到 $O(nm\log n)$. 可以通过全部测试数据. 也可以进一步用双指针去掉 $\log n$.
- P.S. 暴力遍历凸包也可以过，因为我构造不出来能卡掉的数据.

关于本题

- 本题来源 CodeChef CCC - Hit the Coconuts.

关于本题

- 本题来源 CodeChef CCC - Hit the Coconuts.
- 本题可能存在诸多其他做法. 也有可能因为数据过弱，存在错误的做法过了的情况. 我已经把我想到的和其他人想到的错误的做法能卡的都卡了. 如果有人有其他做法，可以尝试在 codechef 上提交一下.

第一档部分分

- 显然，所有的区间都是合法的. 输出 $\frac{(r-l+2)(r-l+1)}{2}$ 即可.

第二档部分分

- 只有刚好跨过那两个位置中恰好一个的区间是不合法的. 分类讨论即可.

第三、四、五档部分分

- 将询问离线下来.

第三、四、五档部分分

- 将询问离线下来.
- 考虑右端点 r 从 1 到 n 枚举. 用 $ans[i]$ 记录 $[i, r]$ 中有多少区间是合法的.

第三、四、五档部分分

- 将询问离线下来.
- 考虑右端点 r 从 1 到 n 枚举. 用 $ans[i]$ 记录 $[i,r]$ 中有多少区间是合法的.
- 左端点 l 从 r 到 1 枚举. 用 cur 记录当前有多少个合法的右端点点为 r 的区间. 令 $ans[i] \leftarrow ans[i] + cur$.

第三、四、五档部分分

- 将询问离线下来.
- 考虑右端点 r 从 1 到 n 枚举. 用 $ans[i]$ 记录 $[i, r]$ 中有多少区间是合法的.
- 左端点 l 从 r 到 1 枚举. 用 cur 记录当前有多少个合法的右端点点为 r 的区间. 令 $ans[i] \leftarrow ans[i] + cur$.
- 判断区间合法可以用线段树, ST 表等数据结构, 维护区间最值. 但实际上, 可以在 l 向左扫的时候维护区间最值, 明显是最高效的. 时间复杂度为 $O(n^2)$.

第六档部分分

- 这档部分分是给常数较大的正确做法，或者根号做法的.

第六档部分分

- 这档部分分是给常数较大的正确做法，或者根号做法的.
- 官方题解给出了一个分块做法，但是难度非常大.

满分

- 考虑一个区间，将数字作为点，将差 1 的数字连边，那么一个区间合法 $\Leftrightarrow |V| - |E| = 1$.

满分

- 考虑一个区间，将数字作为点，将差 1 的数字连边，那么一个区间合法 $\Leftrightarrow |V| - |E| = 1$.
- 依然考虑离线，右端点 r 从左往右扫. 现在考虑对于每一个 l，维护 $[l, r]$ 的 $|V| - |E|$. 这个可以用线段树维护.

满分

- 考虑一个区间，将数字作为点，将差 1 的数字连边，那么一个区间合法 $\Leftrightarrow |V| - |E| = 1$.
- 依然考虑离线，右端点 r 从左往右扫. 现在考虑对于每一个 l，维护 $[l, r]$ 的 $|V| - |E|$. 这个可以用线段树维护.
- 新加了一个数 x，那么 $|V| \to |V| + 1$. 然后再考虑 $y = x \pm 1$. 如果 y 出现在 x 前面，且位置为 p，那么前 p 个位置的 $|E| \to |E| + 1$. 通过线段树的区间操作来维护 $|V| - |E|$.

满分 • 续

- 现在问题在于统计答案. 我们考虑用同一棵线段树，来维护对于每一个 l, $[l,l],[l,l+1],\ldots,[l,r]$ 有多少个区间是合法的.

满分 • 续

- 现在问题在于统计答案. 我们考虑用同一棵线段树，来维护对于每一个 l, $[l,l],[l,l+1],\ldots,[l,r]$ 有多少个区间是合法的.
- 现在对于当前的 r，要将所有 $|V|-|E|=1$ 的答案 $+1$. 我们要对线段树的每个节点不光要记 $|V|-|E|$ 的最小值，还要记 $|V|-|E|$ 取最小值的个数，记 ans 还有 ans 的 tag.

满分 • 续

- 现在问题在于统计答案. 我们考虑用同一棵线段树，来维护对于每一个 l，$[l, l], [l, l+1], \ldots, [l, r]$ 有多少个区间是合法的.
- 现在对于当前的 r，要将所有 $|V| - |E| = 1$ 的答案 $+1$. 我们要对线段树的每个节点不光要记 $|V| - |E|$ 的最小值，还要记 $|V| - |E|$ 取最小值的个数，记 ans 还有 ans 的 tag.
- 我们在 modify ans 的时候，遇到了一个节点，如果 $|V| - |E| > 1$ 则直接退出；否则，考虑它是否完全被 $[1, r]$ 包含. 如果是的话，直接 update，即将 ans 加上 $|V| - |E| = 1$ 取最小值的个数，ans_tag++，不然，就继续往下递归.

满分・续・续

- 现在有一个标记下传的问题. 首先，我们要明确一点，这里打的标记，是在上一次标记下传之后，到这一次标记下传之前，累计的标记. 因为标记下传之后就清空了. 此时该节点内部在相对的意义上没有任何变化，即内部相对大小关系不变. 这个时候，我们考虑儿子会被打上多少的标记. 我们肯定要求儿子的最小值等于父亲的最小值（我们之前已经讲到了，在这段时间内节点内部相对大小关系不变，所以此时等于最小值，之前也一直等于最小值)，这样儿子就能够继承父亲的 tag，也就是儿子的 ans 加上 $tag \cdot \#(|V| - |E|$ 取最小值的个数)，ans_tag 加上 tag.

满分 • 续 • 续

- 现在有一个标记下传的问题. 首先, 我们要明确一点, 这里打的标记, 是在上一次标记下传之后, 到这一次标记下传之前, 累计的标记. 因为标记下传之后就清空了. 此时该节点内部在相对的意义上没有任何变化, 即内部相对大小关系不变. 这个时候, 我们考虑儿子会被打上多少的标记. 我们肯定要求儿子的最小值等于父亲的最小值 (我们之前已经讲到了, 在这段时间内节点内部相对大小关系不变, 所以此时等于最小值, 之前也一直等于最小值), 这样儿子就能够继承父亲的 tag, 也就是儿子的 ans 加上 $tag \cdot \#(|V| - |E|$ 取最小值的个数), ans_tag 加上 tag.
- 答案即线段树维护的 $[l, r]$ 的和.

满分 • 续 • 续

- 现在有一个标记下传的问题. 首先，我们要明确一点，这里打的标记，是在上一次标记下传之后，到这一次标记下传之前，累计的标记. 因为标记下传之后就清空了. 此时该节点内部在相对的意义上没有任何变化，即内部相对大小关系不变. 这个时候，我们考虑儿子会被打上多少的标记. 我们肯定要求儿子的最小值等于父亲的最小值（我们之前已经讲到了，在这段时间内节点内部相对大小关系不变，所以此时等于最小值，之前也一直等于最小值)，这样儿子就能够继承父亲的 tag，也就是儿子的 ans 加上 $tag \cdot \#(|V| - |E|$ 取最小值的个数)，ans_tag 加上 tag.
- 答案即线段树维护的 $[l, r]$ 的和.
- 时间复杂度为 $O(n \log n)$.

关于本题

- 本题来源 Codeforces 997E - Good Subsegments.