浙江农林大学第二十五届程序设计竞赛题解

2023 年 10 月 15 日

启动!

- 图灵测试题，输出即可。（出题人不是 op）

nim again

- 本题为简单题，dp 计数套了 nim 游戏的壳。

nim again

- 本题为简单题，dp 计数套了 nim 游戏的壳。
- nim 游戏经典结论，先手必胜的条件为：各个石子堆的石子个数异或和大于 0。

nim again

- 本题为简单题，dp 计数套了 nim 游戏的壳。
- nim 游戏经典结论，先手必胜的条件为：各个石子堆的石子个数异或和大于 0。
- 本题为后手必胜，因此我们需要求出各个石子堆的石子个数异或和为 0 的不同序列数量。

nim again

- 考虑 dp 求合法的不同序列数量。

nim again

- 考虑 dp 求合法的不同序列数量。
- 令 $f(i,x)$ 为使用 i 个石子摆放成的石子堆序列的异或和为 x 时的方案数，则答案为 $f(n,0)$。

nim again

- 考虑 dp 求合法的不同序列数量。
- 令 $f(i,x)$ 为使用 i 个石子摆放成的石子堆序列的异或和为 x 时的方案数，则答案为 $f(n,0)$。
- 枚举最后一堆有多少个石头，则有转移：

$$f(i,x)=\sum_{j=0}^{i-1} f(j,x\oplus(i-j)).$$

nim again

- 考虑 dp 求合法的不同序列数量。
- 令 $f(i,x)$ 为使用 i 个石子摆放成的石子堆序列的异或和为 x 时的方案数，则答案为 $f(n,0)$。
- 枚举最后一堆有多少个石头，则有转移：

$$f(i,x)=\sum_{j=0}^{i-1} f(j,x\oplus(i-j)).$$

- 复杂度 $O(n^2 2^{\lceil \log n \rceil})$。

迷宫逃脱（easy）

- 简单题，可以使用 dfs 也可以并查集，这里介绍并查集做法。

迷宫逃脱（easy）

- 简单题，可以使用 dfs 也可以并查集，这里介绍并查集做法。
- 先将所有相邻的非陷阱节点合并，然后对每一个陷阱节点，将与之相连的非陷阱节点合并。

迷宫逃脱（easy）

- 简单题，可以使用 dfs 也可以并查集，这里介绍并查集做法。
- 先将所有相邻的非陷阱节点合并，然后对每一个陷阱节点，将与之相连的非陷阱节点合并。
- 对每一个询问判断下两个节点是否处在同一个集合中即可。

迷宫逃脱（easy）

- 简单题，可以使用 dfs 也可以并查集，这里介绍并查集做法。
- 先将所有相邻的非陷阱节点合并，然后对每一个陷阱节点，将与之相连的非陷阱节点合并。
- 对每一个询问判断下两个节点是否处在同一个集合中即可。
- 复杂度：$O((m+q)\alpha(n))$，其中 $\alpha(n)$ 为并查集复杂度。

迷宫逃脱（hard）

- 中档下位题，主要在于代码实现。

迷宫逃脱（hard）

- 中档下位题，主要在于代码实现。
- 先将所有非陷阱节点缩成一个点，然后对于每个询问看下两个连通块周围是否有一个相同的陷阱节点即可。

迷宫逃脱（hard）

- 中档下位题，主要在于代码实现。
- 先将所有非陷阱节点缩成一个点，然后对于每个询问看下两个连通块周围是否有一个相同的陷阱节点即可。
- 因为陷阱节点数量 $k \leq 10^3$，所以可以对于每个连通块开大小为 1000 的 bool 数组记录，询问时枚举判断即可。

迷宫逃脱（hard）

- 中档下位题，主要在于代码实现。
- 先将所有非陷阱节点缩成一个点，然后对于每个询问看下两个连通块周围是否有一个相同的陷阱节点即可。
- 因为陷阱节点数量 $k \leq 10^3$，所以可以对于每个连通块开大小为 1000 的 bool 数组记录，询问时枚举判断即可。
- 当然也可以使用 bitset 加速，但设置了不用也能过的时限。
- 复杂度：$O((n+q)k)$。

AsindE 爆金币咯

- 中档下位题，离散化加二分。

AsindE 爆金币咯

- 中档下位题，离散化加二分。
- 假设最终选择的区域的左上角和右下角坐标分别为 $(x_1, y_1), (x_2, y_2)$，那么 x_1, y_1, x_2, y_2 一定在给定的金币的坐标中出现过。

AsindE 爆金币咯

- 中档下位题，离散化加二分。
- 假设最终选择的区域的左上角和右下角坐标分别为 $(x_1, y_1), (x_2, y_2)$，那么 x_1, y_1, x_2, y_2 一定在给定的金币的坐标中出现过。
- 那么可以直接将金币坐标离散化，这样我们就有了一个 $2n \times 2n$ 的正方形区域。

AsindE 爆金币咯

- 中档下位题，离散化加二分。
- 假设最终选择的区域的左上角和右下角坐标分别为 $(x_1, y_1), (x_2, y_2)$，那么 x_1, y_1, x_2, y_2 一定在给定的金币的坐标中出现过。
- 那么可以直接将金币坐标离散化，这样我们就有了一个 $2n \times 2n$ 的正方形区域。
- 二分边长，枚举区域中的格子作为选定区域的右下角判断是否能收集到 c 个金币，这里使用矩阵前缀和判断。

AsindE 爆金币咯

- 中档下位题，离散化加二分。
- 假设最终选择的区域的左上角和右下角坐标分别为 $(x_1, y_1), (x_2, y_2)$，那么 x_1, y_1, x_2, y_2 一定在给定的金币的坐标中出现过。
- 那么可以直接将金币坐标离散化，这样我们就有了一个 $2n \times 2n$ 的正方形区域。
- 二分边长，枚举区域中的格子作为选定区域的右下角判断是否能收集到 c 个金币，这里使用矩阵前缀和判断。
- 二分检查需要注意实现细节，由于坐标被离散化，二分的边长可能并不在离散化的范围中，可以在每次二分时提前预处理。
- 复杂度：$O(4n^2 \log w)$，w 为值域。

AsindE の 数组

- 中档题，主要考察思维。

AsindE の 数组

- 中档题，主要考察思维。
- 题目中多次提示质数，可以想到用质数作为中转去交换其他数。但如果选最小的质数，则后面可能出现不与它互质的数。

AsindE の 数组

- 中档题，主要考察思维。
- 题目中多次提示质数，可以想到用质数作为中转去交换其他数。但如果选最小的质数，则后面可能出现不与它互质的数。
- 反向思考，倒着找出最大的质数即可。在本题范围中，在质数 p 和 $2p$ 之间一定还存在其他质数，所以找到的最大的质数与给定数据中除自己以外的数互质。

AsindE の 数组

- 中档题，主要考察思维。
- 题目中多次提示质数，可以想到用质数作为中转去交换其他数。但如果选最小的质数，则后面可能出现不与它互质的数。
- 反向思考，倒着找出最大的质数即可。在本题范围中，在质数 p 和 $2p$ 之间一定还存在其他质数，所以找到的最大的质数与给定数据中除自己以外的数互质。
- 复杂度：$O(n+\sqrt{nw})$，其中 $\sqrt{nw}$ 为找质数的复杂度，w 为值域。
- ps：本题还有一种选择最小质数加随机乱搞的做法，跑得比 std 还快，还卡不掉。

子树覆盖

- 中档题，带一点数据结构。

子树覆盖

- 中档题，带一点数据结构。
- 考虑在 dfs 过程中求出每个节点的覆盖数量。

子树覆盖

- 中档题，带一点数据结构。
- 考虑在 dfs 过程中求出每个节点的覆盖数量。
- 假设当前节点为 u，深度为 $depth_u$，覆盖距离为 d_u。在进入该 u 的子节点前先查询此时深度小于等于 $depth_u + d_u$ 的节点数量，然后将节点深度为 $depth_u$ 的节点数量加 1。

子树覆盖

- 中档题，带一点数据结构。
- 考虑在 dfs 过程中求出每个节点的覆盖数量。
- 假设当前节点为 u，深度为 $depth_u$，覆盖距离为 d_u。在进入该 u 的子节点前先查询此时深度小于等于 $depth_u + d_u$ 的节点数量，然后将节点深度为 $depth_u$ 的节点数量加 1。
- 在对所有子节点 dfs 完后再查询一次对应节点数量，将两次查询的结果做差就是该节点的覆盖数量，答案取最大值即可。

子树覆盖

- 中档题，带一点数据结构。
- 考虑在 dfs 过程中求出每个节点的覆盖数量。
- 假设当前节点为 u，深度为 $depth_u$，覆盖距离为 d_u。在进入该 u 的子节点前先查询此时深度小于等于 $depth_u + d_u$ 的节点数量，然后将节点深度为 $depth_u$ 的节点数量加 1。
- 在对所有子节点 dfs 完后再查询一次对应节点数量，将两次查询的结果做差就是该节点的覆盖数量，答案取最大值即可。
- 维护小于等于某个数的数的数量可以用平衡树等数据结构，当然本题可以直接离散化后用树状数组维护。
- 复杂度：$O(n \log n)$。

yy 能量

- 本题为中档题，需要将平方和相关信息分开维护。

yy 能量

- 本题为中档题，需要将平方和相关信息分开维护。
- 先考虑一条路径，设此时从末尾开始往前有一段长为 x 的 “y” 段，且当前的连续 “y” 段长度的平方和为 sum。如果末尾新增一个 “y”，则平方和变为：$sum + 2x + 1$。

yy 能量

- 本题为中档题，需要将平方和相关信息分开维护。
- 先考虑一条路径，设此时从末尾开始往前有一段长为 x 的 “y” 段，且当前的连续 “y” 段长度的平方和为 sum。如果末尾新增一个 “y”，则平方和变为： $sum + 2x + 1$。
- 假设现在有 n 条路径，末尾 “y” 段的长度分别为 $a_1, a_2, \cdots, a_n$，所有路径连续 “y” 段长度平方总和为 sum。现在给每条路径末尾新增一个 “y”，则平方和变为： $sum + (2 \times \sum a_i) + n$。

yy 能量

- 于是可以考虑分别维护 $(x+1)^2 = x^2 + 2x + 1$ 的一次项和常数项。

yy 能量

- 于是可以考虑分别维护 $(x+1)^2 = x^2 + 2x + 1$ 的一次项和常数项。
- 则对于单元格 (i, j)，维护以下信息：

yy 能量

- 于是可以考虑分别维护 $(x+1)^2 = x^2 + 2x + 1$ 的一次项和常数项。
- 则对于单元格 (i,j)，维护以下信息：
  - $F(0,i,j)$ 表示从原点到 (i,j) 的满足 (i,j) 处为 “y” 的路径数量，即前面描述的 n。
  - $F(1,i,j)$ 表示从原点到 (i,j) 的满足 (i,j) 处为 “y” 的的路径的最后一段 “y” 段的长度和，即前面描述的 $\sum a_i$。
  - $F(2,i,j)$ 表示从原点到 (i,j) 的所有路径的连续 “y” 段长度的平方和。
  - $P(i,j)$ 表示从原点到 (i,j) 的路径数量。

yy 能量

- 对于 $P(i,j)$，有如下转移：

$$P(i,j) = P(i-1,j) + P(i,j-1).$$

yy 能量

- 对于 $P(i,j)$，有如下转移:

$$P(i,j) = P(i-1,j) + P(i,j-1).$$

- (i,j) 不为 “y” 时，对 $F(0/1/2,i,j)$ 有如下转移:

$$F(0/1,i,j) = 0.$$
$$F(2,i,j) = F(2,i-1,j) + F(2,i,j-1).$$

yy 能量

- 接下来为当 (i, j) 处为 "y" 时的情况：

$$
\begin{aligned}
F(0,i,j) &= P(i,j). \\
F(1,i,j) &= 2 \times F(1,i-1,j) + F(0,i-1,j) \\
&\quad + 2 \times F(1,i,j-1) + F(0,i,j-1). \\
F(2,i,j) &= F(2,i-1,j) + F(2,i,j-1) \\
&\quad + 2 \times F(1,i-1,j) + F(0,i-1,j) \\
&\quad + 2 \times F(1,i,j-1) + F(0,i,j-1).
\end{aligned}
$$

yy 能量

- 接下来为当 (i,j) 处为 “y” 时的情况：

$$
\begin{aligned}
F(0,i,j) &= P(i,j).\\
F(1,i,j) &= 2\times F(1,i-1,j)+F(0,i-1,j)\\
&\quad +2\times F(1,i,j-1)+F(0,i,j-1).\\
F(2,i,j) &= F(2,i-1,j)+F(2,i,j-1)\\
&\quad +2\times F(1,i-1,j)+F(0,i-1,j)\\
&\quad +2\times F(1,i,j-1)+F(0,i,j-1).
\end{aligned}
$$

- 当 $(i-1,j)$ 处不为 “y” 时，还需要给 $F(0/1/2,i,j)$ 加上 $P(i-1,j)$，$(i,j-1)$ 同理，注意边界处理。

yy 能量

- 接下来为当 (i,j) 处为 "y" 时的情况:

$$
\begin{aligned}
F(0,i,j) &= P(i,j). \\
F(1,i,j) &= 2\times F(1,i-1,j) + F(0,i-1,j) \\
&\quad + 2\times F(1,i,j-1) + F(0,i,j-1). \\
F(2,i,j) &= F(2,i-1,j) + F(2,i,j-1) \\
&\quad + 2\times F(1,i-1,j) + F(0,i-1,j) \\
&\quad + 2\times F(1,i,j-1) + F(0,i,j-1).
\end{aligned}
$$

- 当 $(i-1,j)$ 处不为 "y" 时，还需要给 $F(0/1/2,i,j)$ 加上 $P(i-1,j)$，$(i,j-1)$ 同理，注意边界处理。
- 答案为 $F(2,r,c)$，复杂度为 $O(rc)$。

Azusa 的回家路

- 本题为中档题，想到了就能写，码量可能有点大。

Azusa 的回家路

- 本题为中档题，想到了就能写，码量可能有点大。
- 如果一个地块能使用，那么它一定存在一条路径从一个边界走向对面的边界（对穿）。

Azusa 的回家路

- 本题为中档题，想到了就能写，码量可能有点大。
- 如果一个地块能使用，那么它一定存在一条路径从一个边界走向对面的边界（对穿）。
- 地块边长为 m，如果有 $m+1$ 个这样的地块，那么一定存在两个地块，它们拼起来能满足条件，因为两个地块的中间拼接连通的的位置最多 m 个。

Azusa 的回家路

- 本题为中档题，想到了就能写，码量可能有点大。
- 如果一个地块能使用，那么它一定存在一条路径从一个边界走向对面的边界（对穿）。
- 地块边长为 m，如果有 $m+1$ 个这样的地块，那么一定存在两个地块，它们拼起来能满足条件，因为两个地块的中间拼接连通的的位置最多 m 个。
- 所以我们最多找 $m+1$ 个地块，然后对它们两两枚举即可。复杂度：$O(nm^2+8^2m^4)$，其中 8 为一个地块变化的所有可能数量。

AsindE 的二叉树（special）

- 中档上位题，本题改自 2023 年 5 月校赛中一道交互题的 checker。

AsindE 的二叉树（special）

- 中档上位题，本题改自 2023 年 5 月校赛中一道交互题的 checker。
- 考虑维护删除节点后该节点到根节点的所有节点的节点数量。

AsindE 的二叉树（special）

- 中档上位题，本题改自 2023 年 5 月校赛中一道交互题的 checker。
- 考虑维护删除节点后该节点到根节点的所有节点的节点数量。
- 当删除一个节点 a_i 时，就往上跳父亲（$a_i = \lfloor \frac{a_i}{2} \rfloor$），每次减去对应的儿子数量，如果碰到已被删除的节点就停下来。这样最多会跳 qk 次。

AsindE 的二叉树（special）

- 中档上位题，本题改自 2023 年 5 月校赛中一道交互题的 checker。
- 考虑维护删除节点后该节点到根节点的所有节点的节点数量。
- 当删除一个节点 a_i 时，就往上跳父亲（$a_i = \lfloor \frac{a_i}{2} \rfloor$），每次减去对应的儿子数量，如果碰到已被删除的节点就停下来。这样最多会跳 qk 次。
- 查询时从要查询的节点 c 开始往上跳到最后一个没被删除的父亲后输出即可。

AsindE 的二叉树（special）

- 如果用 map 直接来维护跳过的点，则复杂度应该是 $O(qk\log(qk))$ 的，但这样常数较大，较难通过，需要再进行一些优化，这里给出一种优化方式。

AsindE 的二叉树（special）

- 如果用 map 直接来维护跳过的点，则复杂度应该是 $O(qk\log(qk))$ 的，但这样常数较大，较难通过，需要再进行一些优化，这里给出一种优化方式。
- 事实上在往上跳的过程中，除了 c 的所有父亲以外，如果该点没有在询问中出现过，说明之后的查询不会用到这个点，不需要把它加入 map 维护。

AsindE 的二叉树（special）

- 如果用 map 直接来维护跳过的点，则复杂度应该是 $O(qk\log(qk))$ 的，但这样常数较大，较难通过，需要再进行一些优化，这里给出一种优化方式。
- 事实上在往上跳的过程中，除了 c 的所有父亲以外，如果该点没有在询问中出现过，说明之后的查询不会用到这个点，不需要把它加入 map 维护。
- 最终做法是先离线所有询问，对每一个询问往上跳的时候，根据上面的条件判断是否加入 map，最终 map 里只有 $q+k$ 个点，复杂度为：$O(qk\log(q+k))$。

yyjj 打电动

- 本题为中档题，最短路，dp。

yyjj 打电动

- 本题为中档题，最短路，dp。
- 减少一条花费时间最多的边的时间可以转化为：再能跳过一条边的情况下到一个点的最短路。

yyjj 打电动

- 本题为中档题，最短路，dp。
- 减少一条花费时间最多的边的时间可以转化为：再能跳过一条边的情况下到一个点的最短路。
- 令 $f(i, j, 1/0)$ 为：从起点出发到达 i 点经过了 j 个点，并且有无跳过一条边的最小花费。对每个询问枚举 j 来求出花费的时间的最小值。

yyjj 打电动

- 本题为中档题，最短路，dp。
- 减少一条花费时间最多的边的时间可以转化为：再能跳过一条边的情况下到一个点的最短路。
- 令 $f(i, j, 1/0)$ 为：从起点出发到达 i 点经过了 j 个点，并且有无跳过一条边的最小花费。对每个询问枚举 j 来求出花费的时间的最小值。
- 在实现上可以选择 dp 或者最短路，dp 会更快一点。
- dp：$O(n^3)$，dijkstra：$O((n2+nm)\log(nm))$。

倪佬与无趣的暑假

- 困难题，暴力转移时加上排序和记忆化就行。

倪佬与无趣的暑假

- 困难题，暴力转移时加上排序和记忆化就行。
- 对于每个回合，我们可以分两步考虑：
  1. 该回合结束后存活的人数；
  2. 该回合结束后每个队存活的人数。

倪佬与无趣的暑假

- 先考虑预处理两个信息。

倪佬与无趣的暑假

- 先考虑预处理两个信息。
- $E(i,j)$ 表示当前有 i 个人存活且当前按行动顺序为第 j 个人在行动的概率，显然 $E(i,1)=1$。转移方程如下：

$$\begin{aligned} i \times E(i,j) &= E(i+1,j) \times (j-1) \\ &\quad + E(i+1,j-1) \times (i-j+2). \end{aligned}$$

倪佬与无趣的暑假

- 先考虑预处理两个信息。
- $E(i,j)$ 表示当前有 i 个人存活且当前按行动顺序为第 j 个人在行动的概率，显然 $E(i,1)=1$。转移方程如下：

$$\begin{aligned} i\times E(i,j) &= E(i+1,j)\times(j-1)\\ &\quad + E(i+1,j-1)\times(i-j+2). \end{aligned}$$

- $dp(k,i)$ 表示 k 个人在经过一个回合后剩下 i 个人的概率，则有转移方程：

$$i\times dp(k,i) = E(i+1,i+1)+E(i+1,i).$$

倪佬与无趣的暑假

- 对于每一个回合，用数组表示队伍的状态：

倪佬与无趣的暑假

- 对于每一个回合，用数组表示队伍的状态：
  - $A = \{a_i\}$ 为初始队伍状态，$S_A = \sum a_i$.
  - $B = \{b_i\}$ 为结束队伍状态，$S_B = \sum b_i$.

倪佬与无趣的暑假

- 对于每一个回合，用数组表示队伍的状态：
  - $A=\{a_i\}$ 为初始队伍状态，$S_A=\sum a_i$.
  - $B=\{b_i\}$ 为结束队伍状态，$S_B=\sum b_i$.
- 令 $P(A,B)$ 为一轮后状态 A 变成状态 B 的概率，则有：

$$P(A,B)=\binom{S_A}{S_B}^{-1}\times dp(S_A,S_B)\times\prod\binom{a_i}{b_i}.$$

倪佬与无趣的暑假

- 最后直接从给定的初始状态 $A = \{a_i\}$ 开始 dfs，枚举该状态的所有子状态 B_i 继续 dfs。

倪佬与无趣的暑假

- 最后直接从给定的初始状态 $A = \{a_i\}$ 开始 dfs，枚举该状态的所有子状态 B_i 继续 dfs。
- 令 $f(X)$ 为从状态 X 开始到结束的期望回合数，则状态 A 的期望结束回合数为：

$$f(A) = 1 + \sum (P(A, B_i) \times f(B_i)).$$

倪佬与无趣的暑假

- 最后直接从给定的初始状态 $A=\{a_i\}$ 开始 dfs，枚举该状态的所有子状态 B_i 继续 dfs。
- 令 $f(X)$ 为从状态 X 开始到结束的期望回合数，则状态 A 的期望结束回合数为：

$$f(A)=1+\sum(P(A,B_i)\times f(B_i)).$$

- $f(X)$ 可以搞成递归形式，注意边界情况。

倪佬与无趣的暑假

- 最后直接从给定的初始状态 $A=\{a_i\}$ 开始 dfs，枚举该状态的所有子状态 B_i 继续 dfs。
- 令 $f(X)$ 为从状态 X 开始到结束的期望回合数，则状态 A 的期望结束回合数为：

$$f(A)=1+\sum(P(A,B_i)\times f(B_i)).$$

- $f(X)$ 可以搞成递归形式，注意边界情况。
- 总状态数为：$A\text{的子集数}\times\sum a_i$，排序后子集数约为 3000。
- 复杂度：$O((A\text{的子集数})^2\times\sum a_i)$

倪佬与无趣的暑假

- 最后直接从给定的初始状态 $A=\{a_i\}$ 开始 dfs，枚举该状态的所有子状态 B_i 继续 dfs。
- 令 $f(X)$ 为从状态 X 开始到结束的期望回合数，则状态 A 的期望结束回合数为：

$$f(A)=1+\sum(P(A,B_i)\times f(B_i)).$$

- $f(X)$ 可以搞成递归形式，注意边界情况。
- 总状态数为：$A\text{的子集数}\times\sum a_i$，排序后子集数约为 3000。
- 复杂度：$O((A\text{的子集数})^2\times\sum a_i)$
- ps：由于评测机的性能过于强劲，赛中不用记忆化也能过。

结束

感谢大家聆听！