2023 河南萌新联赛第（五）场

解题报告

郑州轻工业大学

2023 年 8 月 9 日

致谢

感谢河南工业大学 HaPpY1213，河南农业大学 cbyyx，周口师范学院 Ice_teapoy，广州大学 qfl_zzz 为本场比赛提供题面/数据建议。

难度分布

预估难度分布：

- Easy: DB
- Easy-Mideum: IJE
- Mideum: AGL
- Mideum-Hard: MHK
- Hard: CF

赛时过题分布：

- $800 \sim 1000$: DB
- $500 \sim 800$: IJG
- $100 \sim 500$: AE
- $20 \sim 100$: LM
- 20 以下: CFHK

Problem D - 01 分数规划

题目大意

- 给定一个含 0, 1, ? 的字符串，求出将所有 ? 替换为 0, 1 后，字符串的最长连续 1 或 0 的子串长度。
- 关键词：贪心

Problem D - 01 分数规划

题目大意

- 给定一个含 0, 1, ? 的字符串，求出将所有 ? 替换为 0, 1 后，字符串的最长连续 1 或 0 的子串长度。
- 关键词：贪心

- 由于原有的 1 或 0 都不能再改变，我们要尽可能延长连续的 1 或 0 。

Problem D - 01 分数规划

题目大意

- 给定一个含 0, 1, ? 的字符串，求出将所有 ? 替换为 0, 1 后，字符串的最长连续 1 或 0 的子串长度。
- 关键词：贪心

- 由于原有的 1 或 0 都不能再改变，我们要尽可能延长连续的 1 或 0 。
- 分别考虑这两种情况，贪心地想，将所有 ? 替换为 1 或全部替换为 0 后一定不会使答案变差，所以最终的答案对两种情况取 max 即可。

Problem B - 亚托莉 -我挚爱的时光-

题目大意

- 模拟 linux 操作系统，给你一些指令完成安装、卸载某软件或某软件的个人数据。再给定一些查询指令，查询是否安装该软件或是否有该软件的个人数据。
- 关键词：模拟

Problem B - 亚托莉 -我挚爱的时光-

题目大意

- 模拟 linux 操作系统，给你一些指令完成安装、卸载某软件或某软件的个人数据。再给定一些查询指令，查询是否安装该软件或是否有该软件的个人数据。
- 关键词：模拟

- 按照题目给定的命令模拟。通过读入命令的字符串判断接下来进行哪种操作。我们可以用 map 来存储安装的软件及个人数据，查询时也在 map 中查询即可。

Problem I - 双指针

题目大意

- 给定一个数组 a 和数组 b，问有多少对 i，j 满足 $1 \leq i < j \leq n$ 使得 $a_i \times a_j = b_i \times b_j$。
- 关键词：思维，STL

Problem I - 双指针

题目大意

- 给定一个数组 a 和数组 b，问有多少对 i，j 满足 $1 \leq i < j \leq n$ 使得 $a_i \times a_j = b_i \times b_j$。
- 关键词：思维，STL

- 将原式转化为 $a_i/b_i = b_j/a_j$，令 $c_i = a_i/b_i, d_j = b_j/a_j$，这样问题就转化成有多少对 i，j 满足 $1 \leq i < j \leq n$ 使得 $c_i = d_j$。

Problem I - 双指针

题目大意

- 给定一个数组 a 和数组 b，问有多少对 i，j 满足 $1 \le i < j \le n$ 使得 $a_i \times a_j = b_i \times b_j$。
- 关键词：思维，STL

- 将原式转化为 $a_i/b_i = b_j/a_j$，令 $c_i = a_i/b_i, d_j = b_j/a_j$，这样问题就转化成有多少对 i，j 满足 $1 \le i < j \le n$ 使得 $c_i = d_j$。
- 使用 map 分别统计数组 c 和数组 d 中每个数出现的次数并计算答案。注意由于 $i < j$ 当 $a_i/b_i = b_i/a_i$ 时处理不当答案可能会有冗余计算，注意避免这种情况。如果担心 double 精度问题可以用分数代替。注意答案开 long long。

Problem J - 树上 DP

题目大意

- 给定一棵根节点为 1 的树，可以交换任意次任意相邻节点的权值，求出该树的所有子树的节点的权值和的总和的最大值。
- 关键词：贪心

Problem J - 树上 DP

题目大意

- 给定一棵根节点为 1 的树，可以交换任意次任意相邻节点的权值，求出该树的所有子树的节点的权值和的总和的最大值。
- 关键词：贪心

- 每个节点有多少个祖先节点，表示该节点处在多少颗子树中，由于其本身也是一棵子树的根，意味着该节点对美丽值贡献次数为 (祖先的个数 + 1) 次。

Problem J - 树上 DP

题目大意

- 给定一棵根节点为 1 的树，可以交换任意次任意相邻节点的权值，求出该树的所有子树的节点的权值和的总和的最大值。
- 关键词：贪心

- 每个节点有多少个祖先节点，表示该节点处在多少颗子树中，由于其本身也是一棵子树的根，意味着该节点对美丽值贡献次数为 (祖先的个数 + 1) 次。
- 显然每个节点 u 对美丽值的贡献为 u 的深度次，即 $dep_u \times a_u$ (默认根节点深度为 1)。

Problem J - 树上 DP

- 由于可以交换任意次任意相邻节点的权值，所以这些权值可以在树上任意排列。

Problem J - 树上 DP

- 由于可以交换任意次任意相邻节点的权值，所以这些权值可以在树上任意排列。
- 根据贪心，可以尽量将较大的权值放在深度较大的节点，即将权值和深度按非降序排序然后一一相乘的总和就是最大美丽值。

Problem E - 换根 DP

题目大意

- 给定一棵边权只有 1 或 2 的树，两点间距离为路径上边权的 gcd 。定义 $f(t)$ 为节点 t 到树上其他节点的距离之和，求 $\min(f(1), f(2), \ldots, f(n))$。
- 关键词：并查集，贪心

Problem E - 换根 DP

题目大意

- 给定一棵边权只有 1 或 2 的树，两点间距离为路径上边权的 gcd 。定义 $f(t)$ 为节点 t 到树上其他节点的距离之和，求 $\min(f(1), f(2), \ldots, f(n))$。
- 关键词：并查集，贪心

- 注意到 gcd 的性质，我们只要经过一次边权为 1 的边，后续的距离就全部为 1。而对于不经过边权为 1 的边的情况下，两两点对的距离一定为 2。

Problem E - 换根 DP

题目大意

- 给定一棵边权只有 1 或 2 的树，两点间距离为路径上边权的 gcd 。定义 $f(t)$ 为节点 t 到树上其他节点的距离之和，求 $\min(f(1), f(2), \ldots, f(n))$。
- 关键词：并查集，贪心

- 注意到 gcd 的性质，我们只要经过一次边权为 1 的边，后续的距离就全部为 1。而对于不经过边权为 1 的边的情况下，两两点对的距离一定为 2。
- 使用并查集，将两两间距离为 2 的点合并为同一个连通块。则对于一个节点 t，t 与同一连通块内的点距离为 2，与块外其他点的距离均为 1。要使答案最小，我们只需要找到大小最小的连通块并计算答案即可。

Problem A - 买爱心气球

题目大意

- 给定 m 个大气球和 n 个小气球。Alice 和 Bob 依次操作。大气球每次可以拿 5 个, 2 个或 1 个，小气球可以拿取任意数量。Alice 先手操作，问谁先拿完气球？
- 关键词：博弈，SG 定理

Problem A - 买爱心气球

题目大意

- 给定 m 个大气球和 n 个小气球。Alice 和 Bob 依次操作。大气球每次可以拿 5 个, 2 个或 1 个，小气球可以拿取任意数量。Alice 先手操作，问谁先拿完气球？
- 关键词：博弈，SG 定理

- 如果大气球只能拿 1 个或 2 个，这就是经典的巴什博弈问题。即如果 m 是 3 的倍数，后手必胜，否则先手必胜。因为后手可以在先手拿取 x 个之后，对应拿 $3-x$ 个。

Problem A - 买爱心气球

题目大意

- 给定 m 个大气球和 n 个小气球。Alice 和 Bob 依次操作。大气球每次可以拿 5 个, 2 个或 1 个，小气球可以拿取任意数量。Alice 先手操作，问谁先拿完气球？
- 关键词：博弈，SG 定理

- 如果大气球只能拿 1 个或 2 个，这就是经典的巴什博弈问题。即如果 m 是 3 的倍数，后手必胜，否则先手必胜。因为后手可以在先手拿取 x 个之后，对应拿 $3-x$ 个。
- 如果大气球能拿 5 个，可以发现在模 3 意义下拿 5 个和拿 2 个没有本质不同，因此性质不变。

Problem A - 买爱心气球

- 接下来考虑 n 个小气球，假设 $m \% 3 \neq n$，且 $n > m \% 3$，则可以通过将 n 拿至 $m \% 3$，然后与 Bob 进行对手操作即可（如果 Bob 拿大气球，Alice 也拿大气球并凑 3，如果 Bob 拿小气球，Alice 拿相同数量的大气球）。如果 $m \% 3 \neq n$ 且 $n < m \% 3$，可以通过拿大气球来达到 $m \% 3 = n$。此时均为先手必胜。
如果最初状态下 $m \% 3 = n$，相当于先后手对调，此时后手必胜。

Problem A - 买爱心气球

- 接下来考虑 n 个小气球，假设 $m \% 3 \neq n$，且 $n > m \% 3$，则可以通过将 n 拿至 $m \% 3$，然后与 Bob 进行对手操作即可（如果 Bob 拿大气球，Alice 也拿大气球并凑 3，如果 Bob 拿小气球，Alice 拿相同数量的大气球）。如果 $m \% 3 \neq n$ 且 $n < m \% 3$，可以通过拿大气球来达到 $m \% 3 = n$。此时均为先手必胜。
  如果最初状态下 $m \% 3 = n$，相当于先后手对调，此时后手必胜。
- 结论即 $m \% 3 \neq n$ 时，先手必胜，否则，后手必胜。其实大小气球可以分别视作两个独立游戏，根据 SG 定理也可以简单得到答案 $(m \% 3) \oplus n \neq 0$ 时，先手必胜，否则，后手必胜。

Problem G - Kruskal

题目大意

- 给定一张 n 个点的无向完全图，u, v 两点间距离为 $u \& v$，求最小生成树的边权和。
- 关键词：思维，位运算

Problem G - Kruskal

题目大意

- 给定一张 n 个点的无向完全图，u, v 两点间距离为 $u \& v$，求最小生成树的边权和。
- 关键词：思维，位运算

- 结论：当 n 可以被表示为 $2^k - 1$ 的形式时，答案为 1，否则，答案为 0。具体证明如下：

Problem G - Kruskal

题目大意

- 给定一张 n 个点的无向完全图，u, v 两点间距离为 $u \,\&\, v$，求最小生成树的边权和。
- 关键词：思维，位运算

- 结论：当 n 可以被表示为 $2^k - 1$ 的形式时，答案为 1，否则，答案为 0。具体证明如下：
- 既需要满足边权尽量小，还需要连成的是一棵树。因此可以先考虑连出一条链，再将其他点连向链上的节点。

Problem G - Kruskal

题目大意

- 给定一张 n 个点的无向完全图，u, v 两点间距离为 $u \,\&\, v$，求最小生成树的边权和。
- 关键词：思维，位运算

- 结论：当 n 可以被表示为 $2^k - 1$ 的形式时，答案为 1，否则，答案为 0。具体证明如下：
- 既需要满足边权尽量小，还需要连成的是一棵树。因此可以先考虑连出一条链，再将其他点连向链上的节点。
- 如何考虑链上节点的选择？

Problem G - Kruskal

- 设 k 是满足 $2^k > n$ 的最小正整数。可以考虑先将 $2^0, 2^1, \ldots, 2^{k-1}$ 依次连接为一条链，这条链上的边权均为 0。

Problem G - Kruskal

- 设 k 是满足 $2^k > n$ 的最小正整数。可以考虑先将 $2^0, 2^1, \ldots, 2^{k-1}$ 依次连接为一条链，这条链上的边权均为 0。
- 接下来考虑不在链上的其他节点：

Problem G - Kruskal

- 设 k 是满足 $2^k > n$ 的最小正整数。可以考虑先将 $2^0, 2^1, \ldots, 2^{k-1}$ 依次连接为一条链，这条链上的边权均为 0。
- 接下来考虑不在链上的其他节点：
  - 对于 $1 \sim n-1$ ，可以证明这些点的第 $1 \sim k$ 个二进制位中一定有一位 0。根据反证法，如果点 t 的第 $1 \sim k$ 个二进制位上均为 1，则一定存在 $2^k = t + 1 \leq \mathrm{n}$ ，不符合 k 的定义，得证。

Problem G - Kruskal

- 设 k 是满足 $2^k > n$ 的最小正整数。可以考虑先将 $2^0, 2^1, \ldots, 2^{k-1}$ 依次连接为一条链，这条链上的边权均为 0。
- 接下来考虑不在链上的其他节点:
  - 对于 $1 \sim n-1$ ，可以证明这些点的第 $1 \sim k$ 个二进制位中一定有一位 0。根据反证法，如果点 t 的第 $1 \sim k$ 个二进制位上均为 1，则一定存在 $2^k = t + 1 \leq \mathrm{n}$ ，不符合 k 的定义，得证。
  - 因此对于 $1 \leq t \leq n-1$，假设 t 的第 x 个二进制位上为 0，将其连接至链上的 2^{x-1} 即可，且边权为 0。

Problem G - Kruskal

- 设 k 是满足 $2^k > n$ 的最小正整数。可以考虑先将 $2^0, 2^1, \ldots, 2^{k-1}$ 依次连接为一条链，这条链上的边权均为 0。
- 接下来考虑不在链上的其他节点：
  - 对于 $1 \sim n-1$ ，可以证明这些点的第 $1 \sim k$ 个二进制位中一定有一位 0。根据反证法，如果点 t 的第 $1 \sim k$ 个二进制位上均为 1，则一定存在 $2^k = t + 1 \leq \text{n}$ ，不符合 k 的定义，得证。
  - 因此对于 $1 \leq t \leq n-1$，假设 t 的第 x 个二进制位上为 0，将其连接至链上的 2^{x-1} 即可，且边权为 0。
  - 对于 n，如果 $n < 2^k - 1$，仍然可以使用如上方式，以 0 的代价连接 n。

Problem G - Kruskal

- 设 k 是满足 $2^k > n$ 的最小正整数。可以考虑先将 $2^0, 2^1, \ldots, 2^{k-1}$ 依次连接为一条链，这条链上的边权均为 0。
- 接下来考虑不在链上的其他节点：
  - 对于 $1 \sim n-1$ ，可以证明这些点的第 $1 \sim k$ 个二进制位中一定有一位 0。根据反证法，如果点 t 的第 $1 \sim k$ 个二进制位上均为 1，则一定存在 $2^k = t + 1 \leq \mathrm{n}$ ，不符合 k 的定义，得证。
  - 因此对于 $1 \leq t \leq n-1$，假设 t 的第 x 个二进制位上为 0，将其连接至链上的 2^{x-1} 即可，且边权为 0。
  - 对于 n，如果 $n < 2^k - 1$，仍然可以使用如上方式，以 0 的代价连接 n。
  - 如果 $n = 2^k - 1$，显然 n 不论与哪个点连接，均会产生贡献。此时我们令 n 与 1 连接，边权为 1。

Problem L - 异或与

题目大意

- 给定一个长度为 n 的序列 a，找到两个下标 $i, j\,(i \neq j)$，满足 $a_i \oplus a_j = 2 \times (a_i \,\&\, a_j)$。
- 关键词：模拟，构造，位运算

Problem L - 异或与

题目大意

- 给定一个长度为 n 的序列 a，找到两个下标 $i, j\ (i \neq j)$，满足 $a_i \oplus a_j = 2 \times (a_i \,\&\, a_j)$。
- 关键词：模拟，构造，位运算

- $2 \times (x \,\&\, y)$ 等价于 $x \,\&\, y$ 左移 1 位。

Problem L - 异或与

题目大意

- 给定一个长度为 n 的序列 a，找到两个下标 $i, j\ (i \neq j)$，满足 $a_i \oplus a_j = 2 \times (a_i \,\&\, a_j)$。
- 关键词：模拟，构造，位运算

- $2 \times (x \,\&\, y)$ 等价于 $x \,\&\, y$ 左移 1 位。
- 定义 x_i 为 x 的二进制上第 i 位上的数，则有以下推论：。
  - $x_i \oplus y_i = x_{i-1} \,\&\, y_{i-1} \Leftrightarrow y_i = (x_{i-1} \,\&\, y_{i-1}) \oplus x_i$。
  - $x_1 \oplus y_1 = (2 \times (x \,\&\, y)) \,\&\, 1 = 0 \Rightarrow y_1 = x_1 \oplus 0$

Problem L - 异或与

题目大意

- 给定一个长度为 n 的序列 a，找到两个下标 $i, j\ (i \neq j)$，满足 $a_i \oplus a_j = 2 \times (a_i \,\&\, a_j)$。
- 关键词：模拟，构造，位运算

- $2 \times (x \,\&\, y)$ 等价于 $x \,\&\, y$ 左移 1 位。
- 定义 x_i 为 x 的二进制上第 i 位上的数，则有以下推论：。
  - $x_i \oplus y_i = x_{i-1} \,\&\, y_{i-1} \Leftrightarrow y_i = (x_{i-1} \,\&\, y_{i-1}) \oplus x_i$。
  - $x_1 \oplus y_1 = (2 \times (x \,\&\, y)) \,\&\, 1 = 0 \Rightarrow y_1 = x_1 \oplus 0$
- 根据以上推论发现只要 x 确定，y 就只有唯一值，那么只需要枚举 a_i，查询对应的数是是否在序列里即可。

Problem M - 扫描线

题目大意

- 在二维平面上给定一个凸包，任意旋转，求凸包上所有点纵坐标最大值与最小值之差的最小值。
- 关键词：计算几何

Problem M - 扫描线

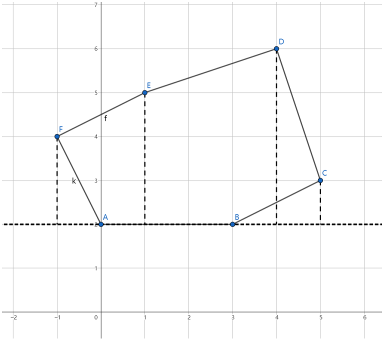
题目大意

- 在二维平面上给定一个凸包，任意旋转，求凸包上所有点纵坐标最大值与最小值之差的最小值。
- 关键词：计算几何

- 不难发现答案一定在凸包一边平行于 x 轴时取到。

Problem M - 扫描线

- 考虑枚举一条平行于 x 轴的边，将该边当作底边，再考虑取哪一个点时到该边距离最大。
  如图所示，顺序地看，每个点到该边的距离是先增后减的，那么我们就可以二分或三分在 log 时间内找到距离最大值。



Problem M - 扫描线

- 对每条边都这样考虑，最终的答案就是每条边的距离最大值取 min，总复杂度 $\mathcal{O}(n \log n)$。

Problem M - 扫描线

- 对每条边都这样考虑，最终的答案就是每条边的距离最大值取 min，总复杂度 $\mathcal{O}(n \log n)$。
- 当然这也是一道旋转卡壳板子题，有兴趣的可以自行了解。

Problem H - 三维偏序

题目大意

- 给定一个长度为 n 的数组 a ，你可以通过 $j-i$ 的花费交换 a_i, a_j 两个元素。对于所有位置 $1 \leq i \leq n$，求出在 a_i 不与其他元素发生交换的前提下，使得 $a_1, \ldots, a_n \leq a_i$ 的最小花费。如果无法交换成如上形式，输出 -1。
- 关键词：树状数组，线段树

Problem H - 三维偏序

题目大意

- 给定一个长度为 n 的数组 a ，你可以通过 $j-i$ 的花费交换 a_i, a_j 两个元素。对于所有位置 $1 \le i \le n$，求出在 a_i 不与其他元素发生交换的前提下，使得 $a_1, \ldots, a_n \le a_i$ 的最小花费。如果无法交换成如上形式，输出 -1。
- 关键词：树状数组，线段树

- 哪些元素需要移动？如何移动花费最小？元素的移动有规律吗？

Problem H - 三维偏序

- 原有的 $1 \sim i-1$ 中的小于等于 a_i 的元素一定不用再移动。因此我们需要再从 $i+1 \sim n$ 中找一些小于等于 a_i 的元素与 $1 \sim i-1$ 中大于 a_i 的元素进行交换。

Problem H - 三维偏序

- 原有的 $1 \sim i-1$ 中的小于等于 a_i 的元素一定不用再移动。因此我们需要再从 $i+1 \sim n$ 中找一些小于等于 a_i 的元素与 $1 \sim i-1$ 中大于 a_i 的元素进行交换。
- 为了花费最小，我们从 $i+1 \sim n$ 中选择的元素应该尽量靠左。即从 $i+1$ 开始从左往右选择 k 个小于等于 a_i 的元素，k 是 $1 \sim i-1$ 中大于 a_i 的元素个数。
  为了方便计算，我们可以假设 $1 \sim i$ 中小于等于 a_i 的元素花费 0 的代价与自己交换。

Problem H - 三维偏序

- 原有的 $1 \sim i-1$ 中的小于等于 a_i 的元素一定不用再移动。因此我们需要再从 $i+1 \sim n$ 中找一些小于等于 a_i 的元素与 $1 \sim i-1$ 中大于 a_i 的元素进行交换。
- 为了花费最小，我们从 $i+1 \sim n$ 中选择的元素应该尽量靠左。即从 $i+1$ 开始从左往右选择 k 个小于等于 a_i 的元素，k 是 $1 \sim i-1$ 中大于 a_i 的元素个数。
  为了方便计算，我们可以假设 $1 \sim i$ 中小于等于 a_i 的元素花费 0 的代价与自己交换。
- 因此原数组中从左到右前 i 个小于等于 a_i 的数一定向左移动或不移动，且最后它们一定在 $1 \sim i$ 中。

Problem H - 三维偏序

- 原有的 $1 \sim i-1$ 中的小于等于 a_i 的元素一定不用再移动。因此我们需要再从 $i+1 \sim n$ 中找一些小于等于 a_i 的元素与 $1 \sim i-1$ 中大于 a_i 的元素进行交换。
- 为了花费最小，我们从 $i+1 \sim n$ 中选择的元素应该尽量靠左。即从 $i+1$ 开始从左往右选择 k 个小于等于 a_i 的元素，k 是 $1 \sim i-1$ 中大于 a_i 的元素个数。
  为了方便计算，我们可以假设 $1 \sim i$ 中小于等于 a_i 的元素花费 0 的代价与自己交换。
- 因此原数组中从左到右前 i 个小于等于 a_i 的数一定向左移动或不移动，且最后它们一定在 $1 \sim i$ 中。
- 设原数组中前 i 个小于等于 a_i 的元素的下标之和为 w，由于这些元素都是向左移动或不移动，所以最后的代价就是 $w - \sum_{j=1}^{i} j$。

Problem H - 三维偏序

- 现在，原题的题意转化为求前 i 个小于等于 a_i 的元素的下标之和。

Problem H - 三维偏序

- 现在，原题的题意转化为求前 i 个小于等于 a_i 的元素的下标之和。
- 由于本题没有修改操作，因此考虑离线处理。我们可以将 a_i 排序，然后从小到大枚举。分别开两个树状数组（或者用一个树状数组维护两个信息）分别维护下标和与数量。每次枚举到一个位置，就将等于该数的所有下标添加在树状数组上。而对于查询答案，由于离线处理，树状数组中只存在小于等于 a_i 的元素，我们只需要用二分查询树状数组中前 i 小的下标之和，时间复杂度 $\mathcal{O}(\text{nlognlogn})$，也可以使用树状数组上二分，时间复杂度 $\mathcal{O}(\text{nlogn})$。

Problem H - 三维偏序

- 现在，原题的题意转化为求前 i 个小于等于 a_i 的元素的下标之和。
- 由于本题没有修改操作，因此考虑离线处理。我们可以将 a_i 排序，然后从小到大枚举。分别开两个树状数组（或者用一个树状数组维护两个信息）分别维护下标和与数量。每次枚举到一个位置，就将等于该数的所有下标添加在树状数组上。而对于查询答案，由于离线处理，树状数组中只存在小于等于 a_i 的元素，我们只需要用二分查询树状数组中前 i 小的下标之和，时间复杂度 $\mathcal{O}(\text{nlognlogn})$，也可以使用树状数组上二分，时间复杂度 $\mathcal{O}(\text{nlogn})$。
- 如果树状数组中的元素数量小于 i，则答案为 -1。

Problem K - 01BFS

题目大意

- 给定一个 $n \times m$ 的矩阵 G，每个单元有权值 $\mathcal{G}_{\rangle,|}$，选定其中一个单元为起点，再选定一个起始方向（上下左右），可以多次跳到该方向上权值严格大于当前单元权值的单元上，如果最开始选择的为左右方向，可换一次方向（上或下）继续跳，如果最开始选择的为上下方向，可换一次方向（左或右）继续跳，也可以不换方向，求最多能跳多少次？
- 关键词：动态规划，枚举

Problem K - 01BFS

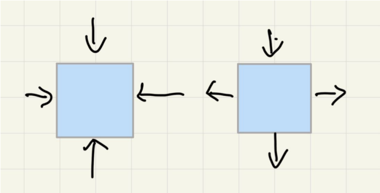
- 考虑当前单元格，从某个方向最多跳多少个单元格可以跳到当前单元格，以及从当前单元格跳出最多能跳多少个单元格。

Problem K - 01BFS

- 考虑当前单元格，从某个方向最多跳多少个单元格可以跳到当前单元格，以及从当前单元格跳出最多能跳多少个单元格。
- 可以通过对每一行每一列的每个方向上都求一次最长递增子序列和一次最长递减子序列，预处理出对于每个单元格四个方向上最多跳多少个单元格可以跳到当前单元格，以及从当前单元格跳出最多能跳多少个单元格。

Problem K - 01BFS

- 枚举每个单元格的情况，显然对于每个单元格都有四个方向可以到达该单元格，而对于每个方向到达单元格后只有三个方向可以走出该单元格，如下图。



Problem K - 01BFS

- 对于每个单元格会有 12 种情况（四个方向跳到当前单元格，三个方向跳出当前单元格），枚举每个单元格的 12 种情况，取最大值就是答案。

Problem C - 数位 DP

题目大意

- 给定一个长度为 2×10^5 的正整数，我们可以任意次删去其中一个数位，问可能会产生多少种不同的数 n，使得 n 不是 3 的倍数。
- 关键词：线性 DP

Problem C - 数位 DP

- 首先不考虑不是 3 的倍数这一限制。我们该如何找到删去一些位后产生不同数的数量？

Problem C - 数位 DP

- 首先不考虑不是 3 的倍数这一限制。我们该如何找到删去一些位后产生不同数的数量？
- 当处理形如 $abbc$ 的数字时，用 b_1 表示前一个 b，b_2 表示后一个 b。从 a 转移到 b_1，再转移到 c 是一个合法方案，从 a 转移到 b_2，再转移到 c 也是一个合法方案。但两者构成的数字相同，重复由此产生。

Problem C - 数位 DP

- 首先不考虑不是 3 的倍数这一限制。我们该如何找到删去一些位后产生不同数的数量？
- 当处理形如 $abbc$ 的数字时，用 b_1 表示前一个 b，b_2 表示后一个 b。从 a 转移到 b_1，再转移到 c 是一个合法方案，从 a 转移到 b_2，再转移到 c 也是一个合法方案。但两者构成的数字相同，重复由此产生。
- 由上可得，在当前状态转移至相同数字的后继状态时，会产生重复。因此我们可以只从当前位置转移至其后第一次出现的数字处，即可避免重复。

Problem C - 数位 DP

- 首先不考虑不是 3 的倍数这一限制。我们该如何找到删去一些位后产生不同数的数量？
- 当处理形如 $abbc$ 的数字时，用 b_1 表示前一个 b，b_2 表示后一个 b。从 a 转移到 b_1，再转移到 c 是一个合法方案，从 a 转移到 b_2，再转移到 c 也是一个合法方案。但两者构成的数字相同，重复由此产生。
- 由上可得，在当前状态转移至相同数字的后继状态时，会产生重复。因此我们可以只从当前位置转移至其后第一次出现的数字处，即可避免重复。
- 因为字符集很小，我们可以预处理出位置 i 后第一次出现 ~ 9 的下标位置，设为 $nxt_{i,0\sim 9}$。用 dp_i 表示处理了前 i 位后产生的不同数字的数量。
  状态转移方程：$dp_i \to dp_{nxt_{i,0\sim 9}}$。
  注意 $i=0$ 时不能转移至 0 字符处，因为会产生前导 0。

Problem C - 数位 DP

- 接下来处理 3 的限制。对于一个数 $abcd$，可以将其拆分为 $a \times 10^3 + b \times 10^2 + c \times 10 + d$。

Problem C - 数位 DP

- 接下来处理 3 的限制。对于一个数 $abcd$，可以将其拆分为 $a \times 10^3 + b \times 10^2 + c \times 10 + d$。
- 我们要求 $abcd$ 不是 3 的倍数，就是要求 $(a \times 10^3 + b \times 10^2 + c \times 10 + d)\%3 \neq 0$ 。因为对于任意非负整数 x，都有 $10^x \ \% \ 3 = 1$，所以只需要 $(a + b + c + d)\%3 \neq 0$ 即可。

Problem C - 数位 DP

- 接下来处理 3 的限制。对于一个数 $abcd$，可以将其拆分为 $a\times10^3+b\times10^2+c\times10+d$。
- 我们要求 $abcd$ 不是 3 的倍数，就是要求 $(a\times10^3+b\times10^2+c\times10+d)\%3\neq0$ 。因为对于任意非负整数 x，都有 $10^x \ \% \ 3=1$，所以只需要 $(a+b+c+d)\%3\neq0$ 即可。
- 为了处理这个限制，我们只需要在 dp 状态上再加一维，用 $dp_{i,0\sim2}$ 表示处理了前 i 位，且选择的数位之和余 3 为 $0\sim2$ 的方案数，后继位置上数字为 x 时，第二维状态也转移至 $(j+x)\%3$ 即可。时间复杂度 $O(30n)$。

Problem F - 布鲁特佛斯

题目大意

- 给定一个边长为 n 的正方形方框，求出用任意个边长为 $2\sim50$ 的正方形拼图恰好填满这个正方形方框的任意合法方案；若不存在构造方案，则输出 -1。
- 关键词：思维，构造

Problem F - 布鲁特佛斯

题目大意

- 给定一个边长为 n 的正方形方框，求出用任意个边长为 $2 \sim 50$ 的正方形拼图恰好填满这个正方形方框的任意合法方案；若不存在构造方案，则输出 -1。
- 关键词：思维，构造

- F 题的构造方式有很多，这里仅提供一种
- 对于边长为 $2 \sim 50$ 的正方形方框，可以直接选用边长为 $2 \sim 50$ 的正方形拼图恰好填满。

Problem F - 布鲁特佛斯

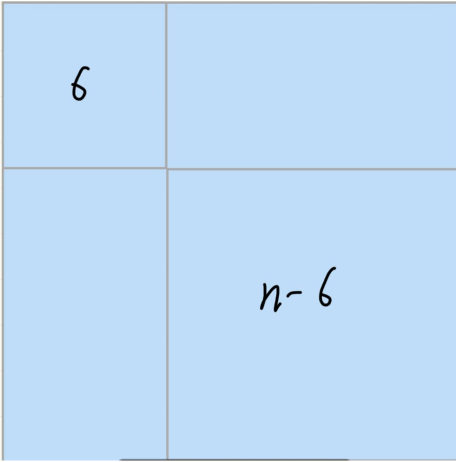
题目大意

- 给定一个边长为 n 的正方形方框，求出用任意个边长为 $2 \sim 50$ 的正方形拼图恰好填满这个正方形方框的任意合法方案；若不存在构造方案，则输出 -1。
- 关键词：思维，构造

- F 题的构造方式有很多，这里仅提供一种
- 对于边长为 $2 \sim 50$ 的正方形方框，可以直接选用边长为 $2 \sim 50$ 的正方形拼图恰好填满。
- 对于边长大于 50 的正方形方框，先考虑边长为偶数的情况，长和宽都为偶数是可以全部用边长为 2 的正方形拼图恰好填满。

Problem F - 布鲁特佛斯

- 现在考虑边长为奇数的情况，可以先构造出下图中的情况。

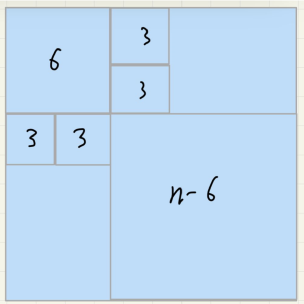


Problem F - 布鲁特佛斯

- 这时会发现有两个长为 $n-6$，宽为 6 的矩形，由于 n 为奇数，所以 $n-6$ 也为奇数，依旧无法构造出长和宽都为偶数的矩形。

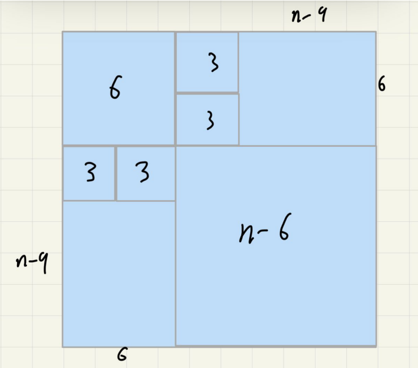
Problem F - 布鲁特佛斯

- 这时会发现有两个长为 $n-6$，宽为 6 的矩形，由于 n 为奇数，所以 $n-6$ 也为奇数，依旧无法构造出长和宽都为偶数的矩形。
- 接下来我们在两个矩形中个分别添加两个边长为 3 的正方形拼图形式如下图。



Problem F - 布鲁特佛斯

- 这时会发现就剩下两个长为 $n-9$，宽为 6 的矩形，由于 n 为奇数，所以 $n-9$ 为偶数，构造出了两个长和宽都为偶数的矩形，可以用边长为 2 的正方形拼图恰好填满。



Problem F - 布鲁特佛斯

- 对于剩下的边长为 $n-6$ 的正方形，我们可以递归处理，用上述方式进行构造，当递归到边长小于等于 50 的正方形时直接用已有的正方形拼图填充即可。

Problem F - 布鲁特佛斯

- 对于剩下的边长为 $n-6$ 的正方形，我们可以递归处理，用上述方式进行构造，当递归到边长小于等于 50 的正方形时直接用已有的正方形拼图填充即可。
- 题目里原本有个小背景故事，本题的思路其实是出题队打牛客时读假题写的，不过开局就被两名选手查到我们队队名和赛时代码交上去了，因此直接把背景故事给删掉了。不过还好影响不大，过题数量在预期范围之内（逃。