

第四届上海理工大学程序设计全国挑战赛 题解

上海理工大学 ACM 集训队

2024 年 5 月 12 日

赛前难度预估

- easy: A, C, M
- easy-mid: I, K
- mid: E, G
- mid-hard: D, J, L
- hard: B, F, H

为了给新生带来带来良好的体验，把题目难度放在了 A 题样例中，希望对大家有所帮助。

M. 蔡光数组

题意

给定 4 个正整数，判断它们是不是 USST-蔡光数组。

Author: Cai_Guang

首次通过

校内组：上海理工大学 严子安 0:07 (+)

校外组：上海师范大学 陶天宇 0:13 (+)

牛客：华中科技大学 mhdh 0:01 (+)

题解

直接判断 $a_1 \neq a_2$ 且 $a_2 \neq a_4$ 且 $a_1 \neq a_4$ 且 $a_2 = a_3$ 即可。

时间复杂度: $\Theta(1)$

题意

给定一个长度为 n 的数组 a_i ，将其按 a_i 从小到大顺序输出序号。

Author: xhytom

首次通过

校内组：上海理工大学 孙乐彦 0:02 (+)

校外组：华东师范大学 顾F涵 0:03 (+)

牛客：东华大学 业余红石 0:01 (+)

A. 我是组题人

题解

使用一个数组记录数字，另一个数组记录每个位子的数原来所在的位置，冒泡排序即可完成，当然也可以使用其他排序方法实现。

时间复杂度： $\Theta(n^2)$

注：本题意在给出题目难度，优化选手参赛体验。

C. 小学题

题意

给定一个给定边长大的正方形和边长位于 $[l, r]$ 范围内整数的小正方形，求小正方形边长为何值时，阴影面积最大。

Author: xhytom

首次通过

校内组：上海理工大学 王一学 0:06 (+)

校外组：上海电力大学 朱凯年 0:08 (+)

牛客：清华大学 Aging1986 0:03 (+)

C. 小学题

题解

设大正方形边长为 n ，小正方形边长为 m ，可得面积

$$S = \frac{m^2}{2} + \frac{nm - m^2}{2} + \frac{n^2 - nm}{2}$$

化简可得

$$S = \frac{n^2}{2}$$

可以发现阴影部分面积与小正方形边长 m 无关，输出任意一个 $[l, r]$ 范围内的整数即可通过本题。

C. 小学题

注意

本题 n, l, r 可能会超过 `int` 类型存储上限 ($2^{31} - 1$), 需要使用 `long long` 来解决。

时间复杂度: $\Theta(T)$

K. 环形数组 (easy)

题意

给定矩阵的长宽，按要求输出矩阵。

Author: Cai_Guang

首次通过

校内组：上海理工大学 王一学 0:14 (+)

校外组：华东师范大学 冯顾炜 0:27 (+)

牛客：嘿嘿 QuQ_iteration 0:01 (+)

K. 环形数组 (easy)

题解

使用三个变量记录当前的位置 (x, y) 和方向 $target$ ，用一个二维数组记录答案矩阵，每次判断前方一格是否越界或已有数字，如果能填就往前走一格并放一个数，否则右转即可。

时间复杂度: $\Theta(nm)$

I. 纠缠之圆

题意

给定两个圆的坐标和半径，计算公切线数量。

Author: [QingTian222](#)

首次通过

校内组：上海理工大学 王一学 0:24 (+)

校外组：华东师范大学 陶雨捷 0:24 (+)

牛客：清华大学 Aging1986 0:09 (+)

I. 纠缠之圆

题解

可以将问题转化为两个圆之间的位置关系：

- 当两个圆重合时，有无数条切线。
- 当两个圆内含时，不存在切线。
- 当两个圆内切时，存在 1 条切线。
- 当两个圆相交时，存在 2 条切线。
- 当两个圆外切时，存在 3 条切线。
- 当两个圆外离时，存在 4 条切线。

分类讨论圆心距和两个圆的半径之间的关系即可通过。

I. 纠缠之圆

注意

计算机使用浮点数来存储小数，浮点数计算会有精度问题。因此采用距离的平方比大小以保证运算中只含有整数或引入 `eps`，避免精度问题。

时间复杂度： $\Theta(T)$

E. 昨日方舟

题意

给定若干条部署指令，问最终地图上的部署情况。

Author: xhytom

首次通过

校内组：上海理工大学 严子安 2:18 (+1)

校外组：华东师范大学 陶雨捷 0:58 (+)

牛客：清华大学 Aging1986 0:27 (+)

E. 昨日方舟

题解

按题目要求模拟所有的操作即可。

时间复杂度： $\Theta(nm + k)$

注：考虑到新生的代码能力较弱，所以出题人给本题设了个较高的难度分。

G. 石子游戏

题意

初始有一堆有 n 个石子的石子堆，设每轮操作前有 x 个石子，则此时可以往石子堆中放 $[1, x]$ 个石子，石子堆中不能超过 k 个石子，双方轮流操作，不能操作则失败。询问获胜方。

Author: xhytom

首次通过

校内组：上海理工大学 檀天 1:49 (+1)

校外组：上海师范大学 蒋文康 0:35 (+)

牛客：华东师范大学 Auroraaaaa 0:06 (+)

G. 石子游戏

题解

从大到小考虑，如果当前石子堆有 k 个石子，那么就无法操作，即 k 为必败态。

那么 $[\lceil \frac{k}{2} \rceil, k-1]$ 均可通过一次操作转到状态 k ,

那么 $[\lceil \frac{k}{2} \rceil, k-1]$ 均为必胜态。于是 $\lfloor \frac{k-1}{2} \rfloor$ 为下个必败态。

如此反复迭代下去，就可以求出所有的必胜态，最后判断初始 n 是必胜或必败态即可。

时间复杂度: $\Theta(T \log k)$

D. 方块游戏

题意

给定 3 种不同方块，画布由若干经过镜像，旋转操作后的上述方块，问画布中每种方块有多少个。

Author: xhytom

首次通过

校内组：上海理工大学 陈述季 2:25 (+)

校外组：同济大学 赵思铭 2:02 (+)

牛客：江西师范大学 Lisens 0:22 (+)

D. 方块游戏

题解

由于存在由几个方块可以拼出一个颜色与 3 号方块相同的区域，导致搜索难度过大。注意到三种方块的三个颜色是线性无关的。设三种方块各有 x, y, z 个，画布中红，蓝，黄的格子各有 R, B, Y 个。我们可以通过三种颜色列出方程组。

$$\begin{cases} 2x + y + z = R \\ x + y + 2z = B \\ x + 2y + 2z = Y \end{cases}$$

D. 方块游戏

题解

解方程组得

$$\begin{cases} x = \frac{(2*R-Y)}{3} \\ y = Y - B \\ z = B - \frac{R+Y}{3} \end{cases}$$

输出方程解 x, y, z 即可。

时间复杂度: $\Theta(nm)$

题意

Author: xhytom

校内组：-

校外组：华东师范大学 冯顾炜 2:18 (+)

牛客：小学 牛客 288141082 号 0:11 (+)

J. 上学

题解

通过容斥原理，可以将题目转化为求有多少个三元集合 (x, y, z) 满足： x, y, z 节点在一条从根节点出发的链上。

我们可以枚举树中的每个点，枚举到的点认为是三元集合中最深的点，之后从所有该点的祖先中选择两个点与该点形成一个三元集合，记该点的深度为 d ，则对答案贡献为 $\frac{d(d-1)}{2}$ 。

最后，我们用任选 3 个点的方案数减去上述计算得出的数量，就是满足题目要求的三元集合的数量。

时间复杂度： $\Theta(n)$

J. 上学

验题人给出的另一种做法

对于第 i 号节点，计算根结点到其路径上不合法答案为 $\binom{3}{dep_i}$ 。考虑容斥原理，对于第 j 号节点，若有 c 个子节点，即相当于 $1 \rightarrow j$ 的答案被减过 c 次，将多减的答案加上即可。

```
int ans = n * (n - 1) * (n - 2) / 6;
std::function<void(int, int)> dfs = [&](int x, int fx) {
    dep[x] = dep[fx] + 1;
    for (auto y : G[x]) {
        if (y == fx) continue;
        dfs(y, x);
    }
    // std::cout << x << ' ' << G[x].size() << ' ' << dep[x] << ' ' << ans << '\n';
    if (G[x].size() == 1) {
        ans -= dep[x] * (dep[x] - 1) * (dep[x] - 2) / 6;
    } else {
        ans += dep[x] * (dep[x] - 1) * (dep[x] - 2) / 6 * (G[x].size() - 2);
    }
};

dfs(1, 0);
```

此外本题也有树形 dp 的做法，在此不再赘述。

L. 环形数组 (hard)

题意

给定环形数组的大小 n, m , 问数字为 x 的格子位置。

Author: Cai_Guang

首次通过

校内组：上海理工大学 孙乐彦 2:25 (+)

校外组：华东师范大学 刘俊毅 1:15 (+)

牛客：湖南大学 sishenzhimu 1:01 (+)

L. 环形数组 (hard)

题解

定义数组的第 x 层为，距离数组四个方向边界距离最小值为 x 的数的集合。

例如下图是一个 5×5 的环形数组，根据定义，蓝色格子为第 1 层，黄色格子为第二层，绿色格子为第 3 层

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

L. 环形数组 (hard)

题解

那么我们可以先考虑数字为 x 的格子位于数组的第几层。
注意到每层公差为 -8 ，则第 i 层中有 $2n + 2m + 4 - 8i$ 个数字。
则前 i 层中共有 $(2n + 2m - 4i) \times i$ 个数字。
通过解不等式或者二分，找到数字为 x 的格子位于的层数。
接下来我们分类讨论 x 的具体位置即可。

L. 环形数组 (hard)

可供参考的分类讨论方式

对于一个大小为 $n \times m$ 的最外层，我们可以将其分成 4 块考虑。其中每块大小分别为 $m-1, n-1, m-1, n-1$ 。



时间复杂度: $\Theta(T \log(\max\{n, m\}))$

B. 俄罗斯方块

题意

给定一个 $n \times n$ 的矩阵，初始有一块 1×1 的方块在其中，你需要将若干给定矩形放入矩阵，使得每个格子都被至少一个矩形覆盖。

Author: xhytom

首次通过

校内组：上海理工大学 卢景天 3:38 (+2)

校外组：上海电力大学 朱凯年 2:07 (+)

牛客：小学 牛客 288141082 号 1:44 (+)

B. 俄罗斯方块

题解

通过计算可以发现，所有矩形加上已经放置的 1×1 的矩形的面积之和恰好等于 n^2 ，所以矩形数量正好够不重叠地覆盖每个方格。

本题做法有许多，这里提供出题人的做法作为参考。

H. 十六进制的疑惑

题意

蔡光在计算正确答案为负数的减法时会得到错误答案，找到有多少对下标不同的蔡光答案（可以是正确的）与正确答案的有序对。

Author: Cai_Guang

首次通过

校内组：-

校外组：-

牛客：小学 牛客 288141082 号 1:25 (+)

H. 十六进制的疑惑

题解

观察可以发现只有负数可以作为错误答案，其余都为正确答案。对所有的非负数答案，蔡光本身计算就是正确的。可以用 `std::map` 和字符串哈希来计算这部分答案。

对于所有的负数，蔡光计算错误原因是最高位借的一个 1 没有还，此时错误答案与正确答案相加为十六进制下形如 $-100...000$ ，所以我们可以对数组中每个负数，枚举和正确答案相加后后缀有多少个 0，通过 $-100...000$ 与错误答案相减后，就能得到对应正确的答案，使用 `std::map` 和字符串哈希来统计这个正确答案的个数。

时间复杂度: $\Theta(nk \log n)$, k 为十六进制最大长度, 本题中为 100。

注：本题中如果直接用 `std::map` 存储 `std::string`，时间复杂度为 $\Theta(nk^2 \log n)$ ，但实际在构造的极限数据下跑的很快，因此常数不大的各种写法都可以通过本题。

F. 十六进制的异或

题意

定义十六进制下的异或为：对于第 i 位 $x_i \oplus y_i = (x_i + y_i) \bmod 16$ 。

对每个十进制数 x 询问数组中十六进制数 a_i 的下标 i ，使得 $x \oplus a_i$ 最大化，保证数组中数字互不相同。

Author: [Cai_Guang](#)

首次通过

校内组：-

校外组：上海电力大学 朱凯年 3:13 (+)

牛客：上海理工大学 上海理工大学郭嘉 1:44 (+)

F. 十六进制的异或

题解

容易发现十六进制下从高到低位考虑使得答案最优，对于当前位 c ，以 $(15 - c) \rightarrow 0, 15 \rightarrow c$ 的优先级考虑，因此使用字典树 (Trie) 从高到低位处理即可。

由于十进制下 10^{18} 在十六进制下长度仍不超过 20，因此在本题中将所有数字转成 20 位的 16 进制即可。

时间复杂度: $\Theta(nk + 16 \times qk) = \Theta((n + q)k)$, k 为十六进制最大长度, 本题中为 20。

注：本题意在考察 Trie。而对于新生选手，也可以依据从高位到低位考虑的性质来构造类似数据结构，从而通过本题。

感谢各位对题面书写提出的改进意见以及对难度的评估。

鸣谢

感谢以下出题人：（排名不分先后）

Cai_Guang, xhytom, LBRhhhh, QingTian222, GammaRays。

感谢以下线下志愿者：（排名不分先后）

fsl123, ocharin, Huangyf, shstyle, CN_Amuzi, sehikonig,
 Dreameclipse, LBRhhhh, xhytom, Cai_Guang, Weigt,
 GammaRays, QingTian222。

感谢 Weigt, LBRhhhh 对线下比赛的技术支持。

感谢牛客竞赛为线上比赛提供技术支持。

感谢教练 vito0210 为我们提供这次举办比赛的机会。

感谢各位参赛选手!