

# “华为杯” 2024 年广东工业大学 ACM 新生程序设计竞赛（决赛）题解

23 级出题组

2024 年 12 月 01 日

## 比赛赛况（封榜前）

赛时共有 127 支队伍进行了有效提交，本场比赛共 14 题，其中 14 题被提交，12 题通过，2 题无人通过。

赛时一共收到了 1536 份提交，其中 633 (41%) 份正确。

## 题目难度分布

题目难度预测分布：

- 签到题: A、N
- Easy: I、E、C、F
- Medium: J、D、K、G
- Hard: B、H、M
- Very-Hard: L

实际通过人数排序：

- A (123/164)、N (120/151)、E (104/143)、I (77/153)、F (70/221)、C (60/338)、K (29/138)、D (19/75)、J (17/29)、G (8/107)、M (5/5)、B (1/5)、H (0/4)、L (0/3)

## A. 演奏春日影

### 简明题意

给定  $n$  个字符串，原样输出的前提下判断是否为目标字符串，如果是则额外插入输出一个字符串。

签到题，按题意模拟即可。

具体的说，按行读入字符串，并原样输出。同时判断读到的字符串是否是“Tomori”，如果是就额外输出一行“Haruhikage”。

时间复杂度  $O(n)$ 。

## N. 哥伦比亚大选

### 简明题意

给定  $n$  个字符串和数值，输出所有数值中最大者对应的字符串。

签到题，按题意模拟即可。

设置初始的最大票数为 0，枚举所有的候选人，如果当前候选人的票数大于之前的票数，那么将答案更新为当前候选人的名字，并更新最大票数，最后输出答案即可。

时间复杂度  $O(n)$ 。

## I. 小 P 爱折跃

### 简明题意

给一个排列，问最少需要交换多少次数才能使得整个排列形成一个大的环。

可以先通过 DFS 或者并查集求出所有的联通块，答案即是联通块个数减一。

时间复杂度  $O(n)$ 。

## E. 黑塔的奇物

### 简明题意

用 1 到  $n$  的数字各  $n$  个，构造一个  $n \times n$  的矩阵，使得矩阵每一行和每一列的异或值的最大值最小。

给出最小值和构造方案。

可以通过打表或手算发现，对于正奇数  $n$ ，有： $n \oplus (n-1) = 1$ 。因此归纳可得：

$$1 \oplus 2 \oplus \cdots \oplus n = \begin{cases} 0, & n \bmod 4 = 3 \\ 1, & n \bmod 4 = 1 \end{cases}$$

容易看出， $n \bmod 4 = 3$  时，让每一列每一行都有 1 到  $n$  的每个数即可获得  $m = 0$  的最小阈值。

## E. 黑塔的奇物

而  $n \bmod 4 = 1$  时，由于 1 到  $n$ ，的数中二进制下最低位是 1 的数的数量是奇数，也就是无论如何摆放这  $n^2$  个数，都至少有一行或一列异或后二进制下最低位是 1。因此最小阈值  $m = 1$ ，具体方案同上。

时间复杂度  $O(n^2)$ 。



## C. 交通要塞

### 简明题意

给定  $n$  个二维平面的水平可行区间，区间每单位时刻向负方向整体移动一个单位。

初始时在  $(0, 0)$ ，每个单位时刻可以选择停留或前进。

问需要至少多少单位时刻才能穿过  $n$  个区间而不接触非法区间，或者报告不可能。

考虑贪心。我们尽可能的向前走，走不了再停住，等到能走后立刻前进。这样显然做一定是最优的。

考虑如何计算最短的路径。首先简化模型，考虑固定所有的车，让车不动，人动。那么每单位时间的操作将转化为  $(x, y) \rightarrow (x, y + 1)$  或者  $(x, y) \rightarrow (x + 1, y + 1)$ 。

## C. 交通要塞

进一步简化模型，我们可以先将所有的  $l_i$  和  $r_i$  都减去  $i$ ，得到了一个等价的变换。这样我们的前进就变成了直线的前进，即每单位时间的操作转化为  $(x, y) \rightarrow (x, y + 1)$  或者  $(x, y) \rightarrow (x + 1, y)$ 。

最后我们直接模拟移动，从原点出发扫过去即可。可以从 1 到  $n$  枚举，枚举到  $x$  时，对  $y$  进行维护，其中  $y$  代表上一步移动到的  $y$  坐标。如果  $l_x > r_{x-1}$  或者  $r_x < y$ ，则无法到达终点；如果  $l_x \leq r_{x-1}$  且  $l_x > y$ ，则需要多等待  $l_x - y$  个单位的时间。最后前进一步，消耗一个单位的时间。

时间复杂度  $O(n)$ 。

## F. 字符串缩写太多了！

### 简明题意

给定  $n$  个字符串，问对于所有可能的英文短语缩写可能含有的不同含义的个数的总和是多少。

简单计数题。观察本题所求答案的本质，发现本题完全不需要在乎字符串的缩写是什么，亦不用在乎字符串的内容是什么，其本质是不同长度的所有字符串的排列方案数之和，即答案为：

$$\sum_{i=1}^n {}^nP_i = \sum_{i=1}^n \frac{n!}{(n-i)!}$$

一种简便的计算方法是，初始时答案为 0，从 1 枚举到  $n$ ，枚举到  $i$  时将答案先自增然后乘上  $i$ 。

时间复杂度  $O(n)$ 。

## J. 好得不能再好了！泰拉投资大师课

### 简明题意

进行若干次独立的实验，每次成功的概率是  $p$ ，问在失败次数超过  $n-1$  次前成功  $n$  次的概率是多少。

直接暴力枚举实验输了的次数，用组合数计算二项分布的概率即可。答案即为：

$$p \times \sum_{i=0}^{n-1} \binom{n-1+i}{n-1} \times p^{n-1} \times (1-p)^i = p^n \times \sum_{i=0}^{n-1} \binom{n-1+i}{i} \times (1-p)^i$$

预处理阶乘及其逆元即可在算答案时  $O(1)$  求解组合数何概率的幂次。

时间复杂度  $O(n + \log m)$ 。

## D. 魔法少女素世世

### 简明题意

给定一个有向无环图，初始时从入度为 0 的点进入，初始魔力值为  $x$ 。

与每个途径节点上的敌人战斗，第  $i$  号节点的敌人魔力值为  $a_i$ 。每场战斗后  $x$  的值会发生如下变化：

$$x = \begin{cases} x + a_i, & \text{if } x \geq a_i, \\ x - (a_i - x), & \text{otherwise.} \end{cases}$$

若此时  $x \leq 0$ ，战斗失败。

问最小的初始魔力值  $x$ ，使得最终可以战胜  $n$  号节点的敌人。

观察到答案具有单调性，考虑通过二分答案求最小的  $x$ 。

## D. 魔法少女素世世

由于题目所给图为有向无环图，且初始从入度为 0 的点进入，可以通过拓扑序遍历 + 动态规划的方式判断某个  $x$  是否可行。

若我们设  $f_i$  表示到达  $i$  号节点，且尚未与该节点敌人进行战斗时的最大可能魔力值，则对于图中的每条边  $u \rightarrow v$ ，有：

$$f_v = \max \left( f_v, \begin{cases} f_u + a_u, & \text{if } f_u \geq a_u, \\ f_u - (a_u - f_u), & \text{otherwise.} \end{cases} \right)$$

若在按拓扑序遍历结束后， $f_n$  的值足以打败  $n$  号节点的敌人，则当前判断的  $x$  合法。  
时间复杂度  $O(m \log a)$ 。

## K. 说重话!!!

### 简明题意

有一个初始时全 0 的长度为  $n$  的数组，然后将首项为 1 公差为 1 等差数列或者完全平方数数列累加到数组的  $m$  个位置上，输出最后形成的数组。

先考虑如果我们想生成一个第一类的数组怎么做。我们可以通过在起始位置放置一个 1，然后进行两次前缀和得到一个目标数组。

接下来，我们考虑如果想要放置若干个第一类数组怎么做。我们发现，按照刚才的步骤分开进行求解，最后再加和到一起，与一开始就共用一个差分数组，然后进行两次前缀和是等效的。

这是因为上述操作具有可加性，求解的过程可以合并在一起，并不会相互干扰。

## K. 说重话!!!

接下来考虑第二个数列的累加怎么做。考察第二个数组第  $k$  项和第  $k+1$  项的关系：

$$(k+1)^2 = k^2 + 2 \cdot k + 1$$

我们发现，可以通过拆解平方项，将第二个数组相邻两项的线性关系表示出来。

因此，同样根据可加性，我们可以通过递推和前缀和将每一个第二类数组在某个位置的项的和算出。

考虑使用与第一个数组相同的办法处理得到  $2 \cdot k, 1$  也可以通过使用差分进行一次前缀和处理得到。

于是，我们就能通过这个转移式求出每个第二类数组在某位置的  $k^2$  的和了。

时间复杂度  $O(n)$ 。



## K. 说重话!!!

另一种可行解法：

直接加的复杂度是  $O(n \cdot m)$  的，考虑优化。对于一个  $a_1$  处为 1，其它为 0 的差分数组，有：

- 做一次前缀和，有  $\dot{a}_i = 1$ 。
- 做两次前缀和，有  $\ddot{a}_i = i$ 。
- 做三次前缀和，有  $\dddot{a}_i = \frac{i^2+i}{2}$ 。

其中第一类操作可以通过  $\ddot{a}_i$  得到；第二类操作可以通过  $2 \cdot \dddot{a}_i - \ddot{a}_i$  得到。

## K. 说重话!!!

具体一点，有  $a, b, c, d$  四个数组：

- 对数组  $a$  做前缀和得到数组  $b$ 。
- 对数组  $b$  做前缀和得到数组  $c$ 。
- 对数组  $c$  做前缀和得到数组  $d$ 。

数组  $d$  即为最终数组。

按照以上分析，对于第一类操作，将  $b_x$  的数  $+1$  即可；

而对于第二类操作，将  $a_x$  的数  $+2$ ，再将  $b_x$  的数  $-1$  即可。

最后再按照顺序依次进行操作和前缀和就可以得到答案。

时间复杂度  $O(n)$ 。

## G. 可乐喝雪碧

### 简明题意

给定  $n$  个整数，在整数之间填充加号或者乘号，问能否使得表达式的值为 0。

首先特判两个简单的情况：

- $a_i$  中有 0 时可以构造。
- $a_i$  中没有  $-1$  时无法构造。

接下来，我们先考虑所有数都是  $-1$ 。

- $-1$  的数量大于等于 3 且为奇数时，总是有解的。因为我们可以将偶数个  $-1$  合并成一个 1，最后再与剩下的  $-1$  相加得到 0。
- $-1$  的数量大于等于 6 且为偶数时，总是有解的。因为我们可以将偶数个  $-1$  拆分成个数均不小于 3 且都为奇数个  $-1$  两个的部分。故这种情况也能构造出 0。

因为所有的 1 都可以通过乘上其周围的数来直接消去，故它的数量对上述解没有影响。

## G. 可乐喝雪碧

接下来考虑 4 个  $-1$  的情况，可以发现，只有 4 个  $-1$  是无法构造的，但一旦在任意位置插入一个 1，就可以进行构造。因此 4 个  $-1$  的情况通过上述方式特判即可。

对于只有 1 个和 2 个  $-1$  的情况，可以发现必须有至少同样数量的 1 才能构造。

综上所述，记  $a$  为  $-1$  的数量， $b$  为 1 的数量， $c$  为 0 的数量，满足以下任意一个条件即可构造：

- $c > 0$
- $b \geq a$
- $a = 3$  或  $a > 4$
- $a = 4$  且  $b > 0$

如果以上条件均不满足，则无法构造出 0。

时间复杂度  $O(n)$ 。

## B. 超平坦棋盘

### 简明题意

在无尽大的棋盘上，不吃子的前提下问将一个指定位置的马移动到目标空格的最少步数是多少。

首先，我们可以把空格和马反转，把空格视作马，马视作空格。因此空格可以直接按照马的形式进行移动。要想让目标马进行第一步移动，我们至少要把空格移动到目标马的八个马脚处。

因此，我们先进行一次 BFS，求解将空格移动到目标马脚的最短路是多少。

接下来，考虑图论建模。我们将每个格子可以拆成八个点，分别代表了当前空格位于目标马的相对方向的状态。

## B. 超平坦棋盘

通过手动构造，可以发现目标马向不同方向移动的代价。

- 向当前空格的方向移动的代价是 1；
- 向当前空格的正对面移动的代价是 5；
- 向其他任何方向移动的代价都是 3；

于是我们可以进行连边，构造出一个图论模型。

## B. 超平坦棋盘

我们可以通过反向连边，然后求终点的单源点最短路来减少求解次数，最后枚举目标马起点周围的八个格子，合并答案取最小值即可。

实现上，可以不用存边来减少常数，拓展点的时候直接现场枚举所有边即可。

注意需要将棋盘扩大大约 5 格左右，因为有些边界也是可以去到的。但是不用补太多，启发性的思考马并不会偏离到达终点的直线路径太多。

时间复杂度  $O(nm \log nm)$ 。

## H. 雪碧喝可乐

### 简明题意

给定  $n$  个整数，在整数之间填充加号或者乘号，一共  $q$  个询问，问能否使得表达式的值为  $x$ 。

$x$  为  $1$  到  $10^9$  的整数。

设最大能达到的快乐值为  $m$ ，则可以发现， $m$  一定是由若干连续整数相乘得到的  $1$ 、 $0$ 、 $-1$  相加得来。因此，只需将其中任意的  $1$  由加改为乘入相邻的数，就可以得到  $m - 1$ 。

并以此类推可以得出：如果表达式的最大值能达到  $m$ ，则可以使表达式的值为  $1$  到  $m$  的任何数。

接下来考虑如何求表达式最大值  $m$ 。



## H. 雪碧喝可乐

考虑 dp, 定义  $f_0 = 0$ ,  $f_i$  表示以  $i$  结尾的表达式的最大值, 通过枚举最后一段连续乘的长度进行转移, 即:

$$f_i = \max_{0 \leq j < i} (f_j + \prod_{k=j+1}^i a_k)$$

但是这样的复杂度是  $O(n^2)$  的, 考虑优化。

- 若  $a_i = 1$ , 则加上一定比乘上更优。因此这种情况下  $f_i$  从  $f_{i-1}$  转移即可。
- 若  $a_i = -1$ , 则要尽量找前一个  $-1$  与之相乘为  $1$ , 或者找前一个  $0$  与之相乘为  $0$ 。对于第一种情况可以发现, 只有在与前一个  $-1$  之间有  $0$  到  $2$  个  $1$  时, 乘起来更优。对于第二种情况, 只有  $a_{i-1}$  为  $0$  时与之相乘更优。因此这种情况下  $f_i$  从  $f_{i-j} (1 \leq j \leq 4)$  转移即可。
- 若  $a_i = 0$ , 则要么直接加上, 要么与前一个  $-1$  与之相乘为  $0$ , 而只有前一个  $-1$  是  $a_{i-1}$  更优。因此这种情况下  $f_i$  从  $f_{i-1}$  和  $f_{i-2}$  转移即可。

## H. 雪碧喝可乐

总结下来，只需要按照  $a_i$  的值对  $f_i$  进行转移即可。但由于最多转移不超过 4 个数，转移方程可以直接写成：

$$f_i = \max_{\max(0, i-4) \leq j < i} (f_j + \prod_{k=j+1}^i a_k)$$

这样表达式最大值  $m = f_n$ ，而对于  $q$  个询问，只需判断  $x$  是否大于  $m$  即可。

时间复杂度  $O(n + q)$ 。

## M. GLLF 砍木棍

### 简明题意

给定长度为  $n$  的木棍，问有多少种切割方式使得切出的所有木棍可以拼成凸多边形，两种方案不同当且仅当切割的点位不同。

先考虑什么情况下能构成凸多边形。与三角形类似，我们发现只要除最长边以外的所有边的长度和大于最长边，即可构成一个凸多边形。简要证明：考虑除了最长边以外所有的边的长度之和等于最长边的长度，此时任意边增加一点长度，即可产生一个凸起，形成凸多边形。

因此问题转化为，有多少种切割方案可以使得除了最长边外的所有边的长度和大于最长边。我们发现直接去求解这个问题并不好做。考虑转换思维，如果我们能求出有多少种不合法的方案，也是可以算出答案的。只需用总方案数减去不合法的方案数即可。

## M. GLLF 砍木棍

总方案由每个切割点是否切割得到，即  $2^{n-1}$ 。先去除显然不合法的  $n$  个方案：

- 不进行切割，共 1 个方案；
- 只切割一次，共  $n-1$  个方案；

先考虑朴素的求解，我们可以暴力枚举最长边的长度  $k$ ，有  $(n-k+3) \times 2^{n-k-2} - 2$  个方案：

- 最长边不在两端时的位置有  $n-k-1$  个，可以随意选择是否切割的点位有  $n-k-2$  个，总方案数为  $(n-k-1) \times 2^{n-k-2}$ 。
- 最长边在两端时至少切两次的方案数： $2 \times 2^{n-k-1} - 2$

于是答案即为：

$$2^{n-1} - n - \sum_{i=\lfloor \frac{n+1}{2} \rfloor}^{n-2} [(n-i+3) \times 2^{n-i-2} - 2]$$

## M. GLLF 砍木棍

然而直接求解是  $O(n)$  的，需要优化。可以使用错位相减法求出求和的通项公式，就能快速进行计算了。

记  $k = \lfloor \frac{n}{2} \rfloor$  最终答案的公式为：

$$2^{n-1} - n - 2^{k-1} \times (k+2) + 2 \times (k-1) + 3$$

时间复杂度  $O(T \log n)$ 。

## L. 向日葵的坡道

### 简明题意

给定一个  $n$  个点  $m$  条边的有向二分图，其中所有的边都从同一个独立点集出发指向另一个独立点集，问最少需要加多少条边，才能使得原图强连通，并给出构造方案。

本题的一般形式都可以化简为本题给出的图论模型，因此本题的解法也可以适用于一般模型。

首先我们定义一些概念：

- 入点：入度为 0 的点。
- 出点：出度为 0 的点。
- 孤立点：入度和出度均为 0 的点。
- 菊花图：一颗大小至少为 2 的树，除树根外的所有点都是树根的叶子。树上所有的边都由叶子指向树根，或者所有的边都由树根指向叶子。
- 菊花图森林：由菊花图构成的森林。

## L. 向日葵的坡道

考察答案的下界，因为每个入点都至少要被加一条入边，每个出点都至少要加一条出边，答案的下界为  $\max(\text{入点点数}, \text{出点点数})$ 。

先考虑一个比较简单的模型，如果给定一个菊花图森林，能否构造出一种方案取到答案的下界。

## L. 向日葵的坡道

答案是肯定的。一种朴素的构造方法是：

- 将每个菊花图取出一条边作为构造大环的边，将这些边上的点依次排开，然后前后相连，构造出一个大环。此时，所有的菊花图都在环上。在这一步中，每个入点匹配到了一个出点。
- 接下来可以将任意的入点和出点进行配对，因为无论如何连边，最后都可以进入大环，也就得到了强连通的性质。在这一步中，每个入点匹配到了一个出点。
- 容易发现，到了最后一步的时候，只剩下了入点或者出点，将他们随意的连上大环即可。

观察过程可以发现，前两步增加的边数是  $\min(\text{入点个数}, \text{出点个数})$ ，而最后一步增加的边数是  $\max(\text{入点个数}, \text{出点个数}) - \min(\text{入点个数}, \text{出点个数})$ ，因此一共增加了的边数是  $\max(\text{入点个数}, \text{出点个数})$ 。

所以通过这种构造方式，我们总是能取到答案的下界，也就是最优解。



## L. 向日葵的坡道

既然菊花图森林有这么好的性质，那么我们能不能将原图转化为菊花图森林，然后就能进行进一步处理了呢？

幸运的是，我们确实有一种方法，总是能把原图转为菊花图森林。

对于孤立点，我们考察它的性质。因为孤立点同时满足入度为 0 且出度为 0，因此它既是入点，也是出点，且与一条大小为 2 的链是等价的。因此我们可以把每个孤立点当作大小为 2 的链进行处理。

## L. 向日葵的坡道

接下来，我们可以把二分图中的节点分为三种：叶子、根、未访问点。

- 1 如果存在度数为 1 的节点，则将其设为叶子，并将其唯一相连的节点设为根。
- 2 否则随意选择一个节点，将其设为叶子，然后任意选择一个与其相连的节点，并设为根，再断开其它与其相连的边。

重复以上操作，直到没有未访问点为止。

关于正确性的证明：

可以发现，在以上操作中，每个节点都至少有一条边连向其它节点，而叶子只能连向根且只有一条边，根只能连向叶子。

这样，节点之间就组成了一个菊花图森林，并且没有新增入点、出点和孤立点。

## L. 向日葵的坡道

综上，本题的解法就是先将原图转化成性质较好的菊花图森林，然后按照上述步骤连边即可。

由于读入数据量较大，如果使用 vector 存图，建议先统计度数，然后提前分配好内存后建边，这样做可以减少大量的常数。也可以使用链式前向星存图。

时间复杂度  $O(n + m)$ 。

## 结语

Thank you!