

2026 第五届昆明青少年人工智能综合实践大赛

C++ 编程决赛（提高组）

考试时间：2026 年 4 月 25 日 14: 00-18: 00

题目名称	墓穴探险	三角洲行动 II	覆盖圆环	古老典籍
题目类型	传统型	传统型	传统型	传统型
目录	tomb	force	ring	ancient
可执行文件名	tomb	force	ring	ancient
输入文件名	tomb.in	force.in	ring.in	ancient.in
输出文件名	tomb.out	force.out	ring.out	ancient.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	3.0 秒
内存限制	512MB	512MB	1024MB	1024MB
子任务数目	20	20	20	25
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	tomb.cpp	force.cpp	ring.cpp	ancient.cpp
对于 C 语言	tomb.c	force.c	ring.c	ancient.c

编译选项

对于 C++ 语言	-O2 -lm
对于 C 语言	-O2 -lm

注意事项（请仔细阅读）

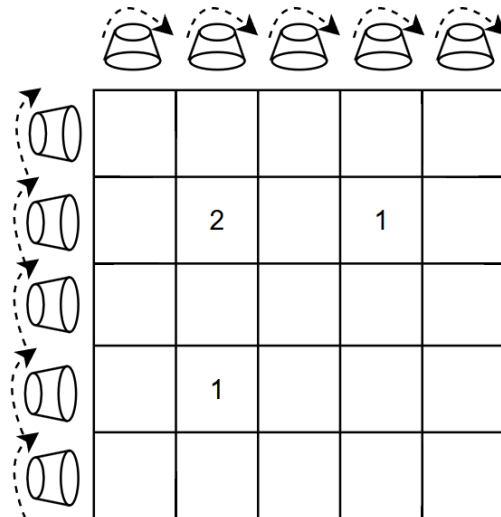
1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考文件提交格式的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 全省统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz 16G 内存。上述时限以此配置为准。
7. 只提供 Linux 格式附加样例文件。
8. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

墓穴探险 (tomb)

【题目描述】

小向很喜欢玩单机游戏《古墓丽影》，他扮演一个考古学家劳拉在全球各个墓穴探险。其中一个游戏关卡是在一个法老王的墓地探险的过程中，有一个谜题，他试了一整天才解决这个谜题。

谜题是有一个 n 行 m 列的矩阵，每个格子都有一个数字，数字的数值在谜题的一开始都是 0，在矩阵的每行每列都对应有一个可以旋转的按钮，左旋一次这个按钮就可以使得对应行或列矩阵内的所有数字减 1，右旋一次则可以使得对应行或列的矩阵内的所有数字加 1。



游戏关卡内另外有一些 k 条提示线索，每个线索要求 i 行 j 列的格子中最终的数字是 x ，对于没有提示的格子，那些都与谜题无关，可以不关心它的最终数字。

小向花了九牛二虎之力才解开这个谜题，于是他兴奋的向小李吹嘘自己的聪明才智。小李却不屑一顾，认为用最少的操作次数解开才是真正的高手。

于是小李把谜题抽象成若干组数据，需要小向用程序输出最少的左旋/右旋次数。不过小李也很懒，他设置的谜题数据是完全随机的，有时会出现无解的情况，那么对应输出字符串“No”即可。

【输入格式】

从文件 **tomb.in** 中读入数据。

每个测试点包含多组测试数据。

输入的第一行是一个正整数 t ，表示测试数据的组数。

每组测试数据的第一行包含三个正整数 n 、 m 、 k 分别表示矩阵的行数、矩阵的列数、谜题的提示数。

接着 k 行，每行三个数字 i 、 j 、 x ，表示一组线索，要求最终 i 行 j 列的格子内的数字是 x 。

【输出格式】

输出到文件 **tomb.out** 中。每组测试数据输出 1 行，如果有解，则输出最少的操作次数，如果无解，输出 “No” 即可。

【样例 1 输入】

```
1 2
2 2 2 4
3 1 1 0
4 1 2 0
5 2 1 2
6 2 2 2
7 2 2 4
8 1 1 0
9 1 2 0
10 2 1 2
11 2 2 1
```

【样例 1 输出】

```
1 2
2 No
```

【样例 1 解释】

输入包含 2 组测试数据。

第一组测试数据谜题的目标如下：

0	0
2	2

操作 2 次第二行的按钮，每次加 1，就可以满足题目条件。明显没有更优的方案。

第二组测试数据谜题的目标如下：

0	0
2	1

每次操作之后，一定是两个格子的数字发生了加 1 或者减 1，所以矩阵内所有数字之和一定是偶数，但是目标的所有格子的之和是奇数 ($0+0+2+1=3$)，所以必然无解。

【样例 2】

见选手目录下 `tomb / tomb2.in` 和 `tomb / tomb2.ans`。

【样例 3】

见选手目录下 `tomb / tomb3.in` 和 `tomb / tomb3.ans`。

【数据范围】

对于所有测试数据有： $1 \leq t \leq 5, 1 \leq n, m \leq 1000, 1 \leq k \leq n \times m, 1 \leq i \leq n, 1 \leq j \leq m, |x| \leq 10^5$ 。保证两两不同的线索之间的网格均不同。保证每个测试点的 k 之和不会超过 2×10^6 。

测试点	$n, m \leq$	特殊性质
1~3	10	无
4	10	$x = 1$
5~7	50	无
8	50	$n = 1$
9~12	100	无
13	100	$k = 1$
14	500	$n = 1$
15~16	500	$x = i$
17~20	1000	无

三角洲行动 II (force)

【题目背景】

三角洲是一款多平台跨端第一人称特战干员战术射击游戏，现在各个年龄段都非常流行，明金也是其中玩家之一。

【题目描述】

游戏的商店系统内有 n 种武器及配件，用 $1\sim n$ 编号，每个物品都最多购买一次。其中配件是需要装备在某个指定的武器或者某个指定的其他配件上才会发挥作用。

对于第 i 个物品，它在商店系统内售价 v_i 元，如果 $s_i = 0$ 则表示这个物品是武器，武器本身会提供 b_i 的战斗力；如果 $s_i \neq 0$ ，它是一种配件，它所装备上的武器或配件编号为 s_i ，增加 b_i 的战斗力。

明金有游戏币 m 元，他想要知道，在花费不超过 m 元的情况下，通过购买武器及相关配件，可以获得的最大战斗力。

【输入格式】

从文件 **force.in** 中读入数据。

输入的第一行是二个正整数 n 、 m ，武器及配件的总数量、明金拥有的金币数量。

接着 n 行，每行三个数字 v_i 、 b_i 、 s_i ，表示第 i 个物品的售价、战斗力、所装备上的武器或配件编号。

【输出格式】

输出到文件 **force.out** 中。

输出仅一个数字，为最大总战斗力。

【样例 1 输入】

```
1 8 12
2 6 10 0
3 8 5 1
4 3 2 0
```

```

5 1 8 1
6 1 3 3
7 3 1 2
8 2 7 3
9 3 9 0

```

【样例 1 输出】

```

1 27

```

【样例 1 解释】

共有 8 件物品，明金有 12 元，物品信息如下表。

物品编号	售价	战斗力	主件编号
1	6	10	无
2	8	5	1
3	3	2	无
4	1	8	1
5	1	3	3
6	3	1	2
7	2	7	3
8	3	9	无

最优方案为购买物品 1 及其附件 4，再单独购买物品 8。花费 $6+1+3=10$ 元，获得战斗力 $10+8+9=27$ 。

【样例 2】

见选手目录下 force / force2.in 和 force / force2.ans。

【样例 3】

见选手目录下 force / force3.in 和 force / force3.ans。

【数据范围】

对于所有测试数据有： $1 \leq n \leq 100$ ， $1 \leq v_i \leq m \leq 10^9$ ， $1 \leq b_i \leq 10$ ， $0 \leq s_i \leq i$ 。

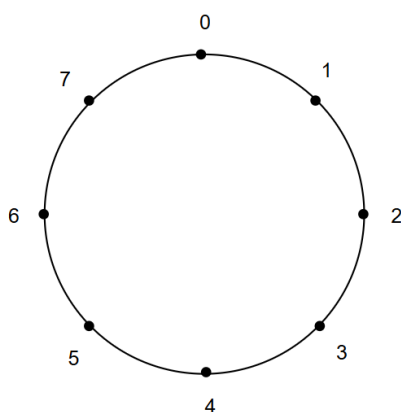
测试点	$n \leq$	$m \leq$	特殊性质
1~2	10	10^2	无

3~5	15	10^9	无
6~9	50	50	无
10~12	50	10^4	保证每个武器最多有 2 个配件 且任何配件不会有它的配件
13~14	50	10^9	无
15	100	100	$s_i = i - 1$
16~17	100	100	无
18~20	100	10^9	无

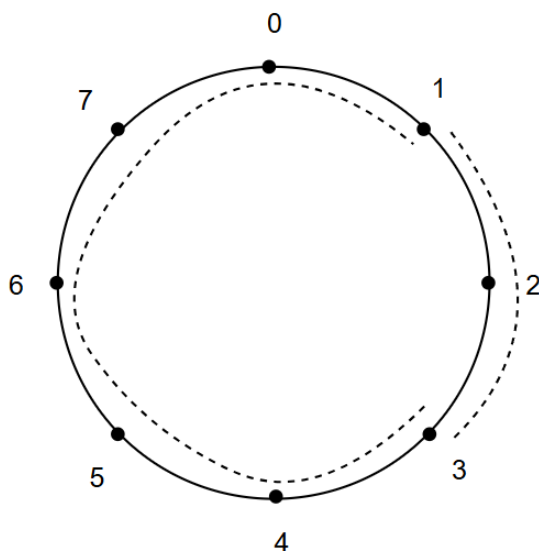
覆盖圆环 (ring)

【题目描述】

有一个周长为 m 的圆，我们从某个点位置为起始位置，从起始位置顺时针沿着圆上移动到达的位置的点的坐标等于其移动的距离。例如下图就是一个周长为8的圆以及部分点的坐标。



有 n 组路径，给出路径的两个端点，可以在两种路径中选择其中一个，比如下图中坐标点1和坐标点3作为路径端点，就有两种路径可以选择。



求这 n 组端点的所有选择中，覆盖的圆环长度的最小值。

【输入格式】

从文件 **ring.in** 中读入数据。输入的第一行包含两个整数 n 、 m ，分别表示点的数量和圆的周长。

接下来输入包含 n 行，每行两个整数 a_i 、 b_i ，表示路径两个端点的坐标。

【输出格式】

输出到文件 **ring.out** 中。输出仅一个数字，即最小覆盖的长度。

【样例 1 输入】

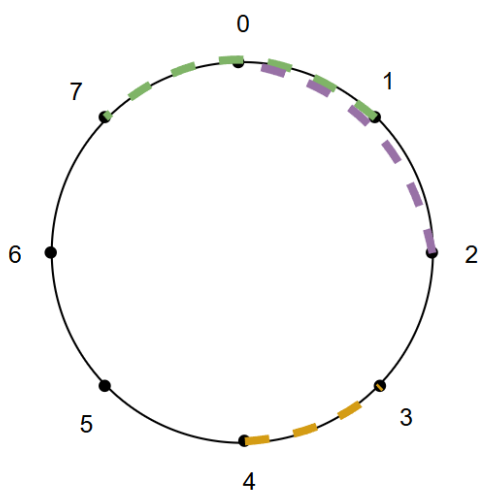
```
1 3 8
2 1 7
3 0 2
4 3 4
```

【样例 1 输出】

```
1 4
```

【样例 1 解释】

当点对选择的路径为 7 顺时针到 1，0 顺时针到 2，3 顺时针到 4 的路径所覆盖周长的长度最小，为 4。如下图。



【样例 2】

见选手目录下的 ring/ring2.in 和 ring/ring2.ans。

【样例 3】

见选手目录下的 ring/ring3.in 和 ring/ring3.ans。

【数据范围】

对于所有测试数据保证： $1 \leq n \leq 10^5, 1 \leq m \leq 10^9, 0 \leq a_i \leq b_i < m$

测试点	$n \leq$	$m \leq$	特殊性质
1	10	10	无
2	10	100	无
3~4	30	30	无
5~6	10^3	10^3	无
7	10^3	10^3	$a_i = b_i$
8~9	10^3	10^9	无
10	30	10^4	无
11~12	10^4	10^4	无
13	10^4	10^4	$a_i = b_i - 1$
14~15	10^4	10^9	无
16	10^5	10^5	$a_i = b_i - 1$
17	10^5	10^5	无
18~20	10^5	10^9	无

古老典籍 (ancient)

【题目描述】

在浩瀚的星空之下，有一座古老的智慧神殿，殿中藏有一棵名为“真理之树”的圣物。这棵树共有 n 个节点，编号为 $1\sim n$ ，其中节点1为根，每个节点都记载着一卷古老的典籍。

传说中，只有按照某种特定的从根节点1开始的深度优先搜索（DFS）顺序阅读学习这些典籍，才能领悟其中的终极智慧。然而，守护神殿的贤者们留下了一条禁忌规则：对于给定的 q 对典籍 (a, b) ，在阅读顺序中，节点 a 的典籍必须出现在节点 b 典籍之前。

现在，你作为新一代的贤者继承者，需要计算：在遵守所有禁忌规则的前提下，一共有多少种不同的阅读顺序？

由于答案可能非常巨大，你需要输出它对 $10^9 + 7$ 取模的结果。

【输入格式】

从文件 `ancient.in` 中读入数据。本题包含多组测试数据。

输入的第一行 t ，表示测试数据的组数。

对于每组测试数据内：

输入的第一行包含三个正整数 n, q ，分别表示树的节点数，以及规则的数量。

接下来 n 行，其中第 i 行的第一个数字 c_i 表示节点 i 的儿子数量，接下来 c_i 个数字，分别为节点 i 的儿子编号。

接下来 q 行，每行两个整数 a, b 。表示典籍 a 的阅读顺序必须出现在典籍 b 之前。

【输出格式】

输出到文件 `ancient.out` 中。

输出 t 行，每行一个数字，表示符合规则的阅读顺序方案数，对 $10^9 + 7$ 取模。

【样例 1 输入】

```
1 2
2 6 1
3 3 2 3 4
4 2 5 6
```

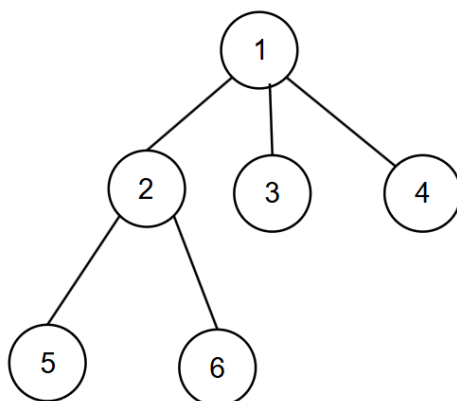
```
5 0
6 0
7 0
8 0
9 2 3
10 6 2
11 3 2 3 4
12 2 5 6
13 0
14 0
15 0
16 0
17 2 3
18 3 5
```

【样例 1 输出】

```
1 6
2 0
```

【样例 1 解释】

样例包含两组测试数据，两组测试数据的树都如下图：



其中第一组测试数据要求节点 2 的典籍必须出现在典籍 3 之前。有 6 种满足条件的 DFS 顺序：

1, 2, 5, 6, 3, 4

1, 2, 6, 5, 3, 4

1, 2, 5, 6, 4, 3

1, 2, 6, 5, 4, 3

1, 4, 2, 5, 6, 3

1, 4, 2, 6, 5, 3

其中第二组测试数据相比第一组测试数据增加一条限制，为节点 3 的典籍必须出现在典籍 5 之前。上面 6 种 DFS 顺序均不能使得 3 在 5 之前，满足条件的 DFS 顺序数为 0。

【样例 3】

见选手目录下 `ancient/ancient3.in` 和 `ancient/ancient3.ans`。

【数据范围】

对于所有测试数据有： $1 \leq t \leq 5, 1 \leq n \leq 5 \times 10^4, 0 \leq q \leq 5 \times 10^4, 0 \leq c_i \leq 15, a \neq b$ 。

测试点	$n \leq$	$q \leq$	$c_i \leq$	特殊性质
1~3	10	10	10	
4~5	10	10	10	$c_1 = n - 1$
6~7	30	30	10	
8~9	10^2	2	2	
10~11	10^2	10^2	10	
12~13	10^3	0	10	
14	10^3	1	10	
15~17	10^3	10^3	10	
18	10^4	10^4	10	所有规则的 a、b 其中一个为 1
19	10^4	3	10	
20	10^4	10^4	2	
21~25	5×10^4	5×10^4	15	