

2026 第五届昆明青少年人工智能综合实践大赛

C++ 编程决赛（入门组）

考试时间：2026 年 4 月 25 日 8: 30-12: 00

题目名称	挖通湖泊	封印	圣诞树	道路规划
题目类型	传统型	传统型	传统型	传统型
目录	dig	seal	tree	road
可执行文件名	dig	seal	tree	road
输入文件名	dig.in	seal.in	tree.in	road.in
输出文件名	dig.out	seal.out	tree.out	road.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	256MB	256MB	512MB	512MB
子任务数目	10	20	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	dig.cpp	seal.cpp	tree.cpp	road.cpp
对于 C 语言	dig.c	seal.c	tree.c	road.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。

挖通湖泊（dig）

【题目描述】

X 国地理特征独特，用一个 n 行 m 列的二维网格表示。X 国内有两个湖泊，其中一个湖泊在春夏水位暴涨，在秋冬水位下降；另外一个则是在秋冬水位暴涨，在春夏水位下降。一个湖泊由上下左右相邻的字符 ‘0’ 网格组成。除了湖泊的网格外，都是地面网格，地面网格都是用字符 ‘1’ 表示。

X 国为了保障国内的农耕活动，决定把两个湖泊挖通，以平衡水位。不过耕地面积也很重要，X 国希望能够保留尽可能多的地面网格数量。

于是这个重要的任务就落到了你身上，求在挖通两个湖泊后，X 国能保留的最大地面网格数。

【输入格式】

从文件 **dig.in** 中读入数据。

第一行 2 个整数 n 、 m ，表示网格的行数与列数。

接着 n 行，每行 m 个字符，其中字符 0 表示湖泊网格、字符 1 表示地面网格。

【输出格式】

输出到文件 **dig.out** 中。

输出仅 1 个整数，表示挖通两个湖泊后，X 国能保留的最大地面网格数。

【样例 1 输入】

```
1 4 4
2 0 1 1 1
3 1 1 1 1
4 1 1 1 1
5 1 1 1 0
```

【样例 1 输出】

```
1 9
```

【样例 1 解释】

二维网格情况如下图：

湖	地	地	地
地	地	地	地
地	地	地	地
地	地	地	湖

挖通左上角的湖和右下角的湖至少需要 5 格，方案有很多，下面给出其中一种：

湖	湖	湖	湖
地	地	地	湖
地	地	地	湖
地	地	地	湖

剩下地面的网格数是 9 格。

【样例 2 输入】

```
1 5 5
2 00000
3 01110
4 01010
5 01110
6 00000
```

【样例 2 输出】

1 7

【样例解释】

二维网格如下图。

湖1	湖1	湖1	湖1	湖1
湖1	地	地	地	湖1
湖1	地	湖2	地	湖1
湖1	地	地	地	湖1
湖1	湖1	湖1	湖1	湖1

只要挖掉湖 2 那个网格上、下、左、右相邻的其中任意一个网格就可以把两个湖连通，此时地面网格数为 7。

【数据范围】

对于所有测试数据保证： $1 \leq n, m \leq 2000$ 。保证只有两个湖泊，保证两个湖泊起始不连通。

测试点	$n, m \leq$
1~2	20
3~4	100
5~6	200
7~8	1000
9~10	2000

封印 (seal)

【题目描述】

在远古的神话时代，众神曾创造了一件拥有无限力量的神器——“万物之核”。为了防止凡人滥用这股力量，众神将它的力量分解为 n 个圣数，刻在一块名为“命运之碑”的石板上，它们分别是 a_1, a_2, \dots, a_n 。

传说中，若要重新唤醒这件神器，必须将这 n 个圣数相乘，得到“本原之数”：

$$N = \prod_{1 \leq i \leq n} a_i = a_1 \times a_2 \times \dots \times a_n$$

而神器的封印能否解开，取决于一个更深的奥秘——本原之数的所有正因子之和。只有当这个和等于某个天启之数时，封印才会显现裂痕。

这个数字可能很大，人类很难计算，众神决定给人类一个破解的机会：结果必须对 $10^9 + 7$ 取模。

如今，你在古神殿的废墟中发现了这块命运之碑。作为一位精通数论的探险者，你必须迅速计算：

$$\sigma(N) = \left(\sum_{d|N} d \right) \bmod (10^9 + 7)$$

即 N 的所有正因子之和，再对 $10^9 + 7$ 取模。

【输入格式】

从文件 `seal.in` 中读入数据。

输入的第一行为数字 n 。

输入的第二行包含 n 个数字，分别为 a_1, a_2, \dots, a_n 。

【输出格式】

输出到文件 `seal.out` 中。

输出仅一个数字，即 N 的所有正因子之和，再对 $10^9 + 7$ 取模的结果。

【样例 1 输入】

```

1 3
2 2 5 10

```

【样例 1 输出】

```

1 217

```

【样例 1 解释】

$N = 2 \times 5 \times 10 = 100$ 。100 的因子有：1、2、4、5、10、20、25、50、100。
总和为 $1 + 2 + 4 + 5 + 10 + 20 + 25 + 50 + 100 = 217$ 。

【数据范围】

对于所有测试数据保证： $1 \leq n, a_i \leq 2 \times 10^5$

测试点	$n \leq$	特殊性质
1~3	10	$a_i \leq 5$
4~5	10^2	无
6	10^3	$a_i \leq 5$
7~8	10^3	无
9	10^4	a_i 为两两不同的质数
10	10^4	无
11~12	10^5	无
13~14	10^5	$a_i \leq 2$
15~16	2×10^5	$a_i = 2^k, k$ 为正整数
17~20	2×10^5	无

圣诞树 (tree)

【题目描述】

圣诞节到了,小基准备在他花园里的大树上布置若干灯泡庆祝节日。小基的大树的形状类似于树这种数据结构。

大树由若干分支节点和叶子节点组成,一共 n 个节点,编号为 $1\sim n$,其中编号为 1 的节点是根节点,小基希望在这些节点中选择若干节点,每个节点上放一个灯泡。

小基希望所有叶子节点到根节点的每条简单路径上,灯泡数量均为 3 的倍数,以表他对耶稣的崇拜。

求小基最多可以放多少灯泡。

【输入格式】

从文件 `tree.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 n ,表示树的节点的数量。

接着 $n-1$ 行,每行两个个数字 u, v ,表示节点 u 和节点 v 之间有一条边。

【输出格式】

输出到文件 `tree.out` 中。

输出仅一个数字,为最大的灯泡数量。

【样例 1 输入】

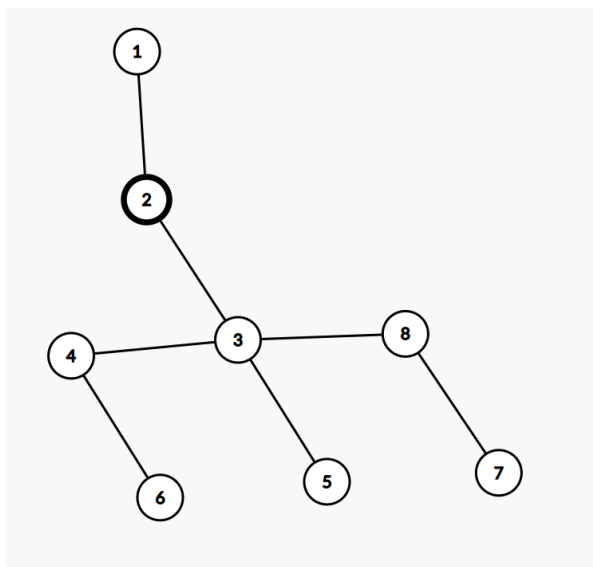
```
1 8
2 1 2
3 6 4
4 7 8
5 8 3
6 3 2
7 4 3
8 5 3
```

【样例 1 输出】

1 5

【样例 1 解释】

树的形状如下图。



最优方案为 1、2、3 中任选 2 个放置灯泡，4、6 中任选 1 个放置灯泡，7、8 中任选 1 个放置灯泡，5 必须放置灯泡，这样使得 1-6、1-5、1-7 的树上简单路径上均有 3 个灯泡。总计 5 个灯泡，明显不存在灯泡更多的方案。

【样例 2】

见选手目录下 `tree / tree2.in` 和 `tree / tree2.ans`。

【样例 3】

见选手目录下 `tree / tree3.in` 和 `tree / tree3.ans`。

【数据范围】

对于所有测试数据有： $1 \leq n \leq 2 \times 10^5$ 。

测试点	$n \leq$	特殊性质
1~2	10	无
3~5	15	无
6~7	10^2	树是满 k 叉树
8~9	10^2	$u + 1 = v$
10~12	10^3	无
13	10^4	树是满 k 叉树
14~15	2×10^4	无

16	10^5	$u + 1 = v$
17~20	2×10^5	无

道路规划 (road)

【题目背景】

小 A 所在的岛上一共有 n 座城市，城市之间一共有 m 条单向道路连接，保证这些道路不会形成环路，也就是不存在从某座城市出发，经过一系列的道路最终又回到原点的情况。

出于某种原因，小 A 需要从起点城市 s 前往城市 t ，可能会有许多种不同的路径。小 A 想知道途径的可能的城市（除 s 、 t 外）中，哪一个城市对他是最重要的，换句话说，从 s 到 t 的所有路径中，需要经过哪个城市的路径条数最多。

【题目描述】

给定一个 n 个点 m 条边的有向无环图，所有点的编号从 1 到 n ，给出一个起点 s 和一个终点 t ，求出一个点 ans ，使得经过 ans 的从 s 到 t 的路径最多。

输出这个点 ans 和具体的路径数，如果有多个点经过的数量相同，取编号最小的那一个；如果 s 不能到达 t ，或从 s 到 t 的路径中间不能途径其他点，输出 -1 。

【输入格式】

从文件 `road.in` 中读入数据。

输入的第一行包含 4 个整数，分别为 n, m, s, t ，表示点数、边数、起点和终点。

接下来 m 行，每行 2 个整数 u, v ，表示存在一条从 u 到 v 的有向路径。

【输出格式】

输出到文件 `road.out` 中。

输出仅 1 行。如果 s 不可达 t ，或中间不能途径其他点，输出 -1 ；否则输出两个数字，表示最重要的点的编号和经过它的路径条数。

【样例 1 输入】

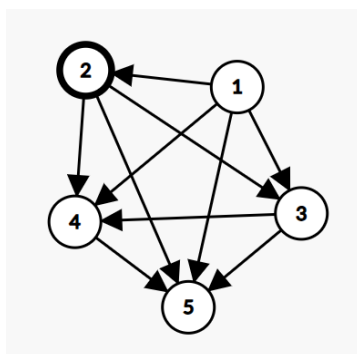
```
1 5 10 1 5
2 1 2
3 1 3
4 1 4
5 1 5
6 2 3
7 2 4
8 2 5
9 3 4
10 3 5
11 4 5
```

【样例 1 输出】

```
1 2 4
```

【样例 1 解释】

样例图如下，起点是 1，终点是 5：



经过 2 的路径最多，一共有 4 条：

{1,2,3,5}, {1,2,4,5}, {1,2,5}, {1,2,3,4,5}。所有从 1 出发到达 5 的路径中，经过其他点的路径均小于 4。比如，经过 3 的路径只有 3 条 {1,2,3,5}, {1,3,5}, {1,3,4,5}。

【样例 2】

见选手目录下 road / road2.in 和 road / road2.ans。

【样例 3】

见选手目录下 `road / road3.in` 和 `road / road3.ans`。

【数据范围】

对于所有测试数据保证： $1 \leq n \leq 10^4$, $1 \leq m \leq 5 \times 10^5$, 保证不会出现重边和自环。保证 s 到 t 的路径条数不超过 $2^{63} - 1$ 。

测试点	$n \leq$	$m \leq$	特殊性质
1~2	10	30	无
3~4	15	45	无
5	10^2	3×10^2	A
6~7	10^2	3×10^2	无
8~9	60	3×10^3	C
10	10^3	3×10^3	B
11~13	10^3	5×10^3	无
14~15	10^4	5×10^4	无
16	10^4	5×10^5	B
17~20	10^4	5×10^5	无

特殊性质 A: 保证有且仅有一个点入度为 0, 出度为 1; 有且仅有一个点入度为 1, 出度为 0, 其余所有的点入度出度均为 1。

特殊性质 B: 除起点 s 和终点 t 外, 所有点的出度和入度均为 1。

特殊性质 C: $m = \frac{n(n-1)}{2}$ 。