

2026 全国青少年信息学奥林匹克竞赛

CCF NOI 2026

第一试

时间：2026 年 7 月 20 日 08:00 ~ 13:00

题目名称	线段	传送	布丁
题目类型	传统型	传统型	交互型
目录	segment	teleport	pudding
每个测试点时限	2.5 秒	3.5 秒	10.0 秒
内存限制	512 MiB	1024 MiB	1024 MiB
测试点数目	25	20	3
测试点是否等分	是	是	否
预测试点数目	25	20	3

提交程序源文件名

对于 C++ 语言	segment.cpp	teleport.cpp	pudding.cpp
-----------	--------------------	---------------------	--------------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	-------------------------------

注意事项（请仔细阅读）

- 赛后正式测试时将以选手留在题目目录下的程序源文件为准。对程序源文件以外文件的修改不会影响评测结果。
- 选手提交的程序源文件大小不得超过 100 KiB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。
- 禁止在提交的程序源文件中改变编译器参数（如使用 **#pragma** 命令），禁止使用系统结构相关指令（如内联汇编）或其他可能造成不公平的方法。
- 因违反上述规定而出现的问题，申诉时一律不予受理。
- 选手可使用快捷启动页面中的工具 selfEval 进行自测。选手需将待测程序的源文件置于相应题目目录下。每次自测时可选择全部或部分题目进行自测。注意：自测有次数限制，且自测结果仅供选手测试参考，不作为最终正式成绩。

传送 (teleport)

【题目描述】

C 国共有 n 座城市，编号为 $0 \sim n-1$ 。这 n 座城市由 $n-1$ 条道路连接，形成树形结构。第 i ($0 \leq i < n-1$) 条道路连接城市 u_i 和 v_i ，从其一端的城市到达另一端需要耗费 1 单位的时间。

为了提升通行效率，C 国研发了一种新型传送门。每座城市中均有一扇传送门。使用传送门同样耗费 1 单位时间，但由于系统尚不稳定，它会将使用者等概率地传送到所有 n 座城市之一。**注意：使用传送门也可能被传送到当前所在的城市。**

为了检测传送门的效果，C 国进行了 m 次测试。第 i ($0 \leq i < m$) 次测试要求测试员从城市 x_i 出发，去往城市 y_i 。在从起点前往终点的过程中，测试员可以选择沿道路移动，或是使用传送门。由于可能的通行方式很多，测试员需要计算出期望耗时最短的通行方式。具体地，定义一种通行方式如下：对于每座非终点的城市，选择一座与其相邻的城市或是使用传送门，每当测试员到达该城市时，均按事先确定的方式移动，即移动至该相邻的城市，或使用传送门。形式化地，一种通行方式可以用一个长度为 n 的序列 $[a_0, \dots, a_{n-1}]$ 表示，其中 $a_{y_i} = -1$ ，且对于所有 $j \neq y_i$ ，均有 a_j 与 j 相邻，或 $a_j = n$ 。每当测试员到达城市 j ($j \neq y_i$) 时，若 $a_j < n$ ，则测试员将移动到 a_j ，否则测试员将使用传送门。称一种通行方式是合理的，当且仅当其期望耗时为有限值。

对于每次测试，请计算在所有合理的通行方式中，期望耗时的最小值。

【实现细节】

选手不需要，也不应该实现 `main` 函数。

选手需要确保提交的程序源文件包含头文件 `teleport.h`，即在程序开头加入以下代码：

```
1 #include "teleport.h"
```

选手需要在提交的程序源文件 `teleport.cpp` 中实现以下函数：

```
1 std::vector<std::pair<long long, int>> teleport(int c, int
    n, int m, std::vector<int> u, std::vector<int> v,
    std::vector<int> x, std::vector<int> y);
```

- c, n, m 分别表示测试点编号、城市数量和测试的次数。 $c = 0$ 表示该测试点为样例。
- u, v 分别表示每条道路连接的两座城市。
- x, y 分别表示每次测试的起点与终点。

- 该函数需要返回一个长度恰好为 m 的二元组序列 $(a_0, b_0), (a_1, b_1), \dots, (a_{m-1}, b_{m-1})$, 其中 a_i, b_i ($0 \leq i < m$) 表示第 i 次测试中, 期望耗时的最小值的最简分数形式为 a_i/b_i 。特别地, 若期望耗时的最小值为正整数, 则视为 $b_i = 1$ 。
- 对于每个测试点, 该函数会被评测程序调用恰好一次。

本试题目录下的 `template_teleport.cpp` 是提供的示例代码, 选手可参考并实现自己的代码。

【测试程序方式】

选手可以在本题目录下使用如下命令编译得到可执行文件:

```
1 g++ grader.cpp teleport.cpp -o teleport -O2 -std=c++14
   -static
```

对于编译得到的可执行文件 `teleport`:

- 可执行文件将从标准输入读入以下格式的数据:
 - 第一行包含三个非负整数 c, n, m 。
 - 第 $i + 2$ ($0 \leq i < n - 1$) 行包含两个非负整数 u_i, v_i 。
 - 第 $i + n + 1$ ($0 \leq i < m$) 行包含两个非负整数 x_i, y_i 。
- 可执行文件将输出以下格式的数据至标准输出:
 - 第 $i + 1$ ($0 \leq i < m$) 行包含两个正整数 a_i, b_i 。

【样例 1 输入】

```
1 0 4 4
2 0 1
3 1 2
4 2 3
5 0 3
6 0 1
7 0 2
8 1 2
```

【样例 1 输出】

```
1 7 3
2 1 1
3 2 1
```

4 | 1 1

【样例 1 解释】

对于第 0 次测试，

- 若通行方式为 $[1, 2, 3, -1]$ ，则耗时为固定值 3；
- 若通行方式为 $[4, 2, 3, -1]$ ，则测试员将不断使用传送门直至离开城市 0，因此期望耗时为 $7/3$ ；
- 若通行方式为 $[4, 4, 3, -1]$ ，则测试员将不断使用传送门直至到达城市 2 或城市 3，因此期望耗时为 3；
- 若通行方式为 $[1, 0, 4, -1]$ ，则测试员将永远在城市 0 与城市 1 间移动，因此该通行方式不是合理的。

可以证明，期望耗时的最小值为 $7/3$ 。

【样例 2】

见选手目录下的 *teleport/teleport2.in* 与 *teleport/teleport2.ans*。

该样例满足测试点 2, 3 的约束条件。

【样例 3】

见选手目录下的 *teleport/teleport3.in* 与 *teleport/teleport3.ans*。

该样例满足测试点 4 ~ 6 的约束条件。

【样例 4】

见选手目录下的 *teleport/teleport4.in* 与 *teleport/teleport4.ans*。

该样例满足测试点 7 ~ 8 的约束条件。

【样例 5】

见选手目录下的 *teleport/teleport5.in* 与 *teleport/teleport5.ans*。

该样例满足测试点 9 的约束条件。

【样例 6】

见选手目录下的 *teleport/teleport6.in* 与 *teleport/teleport6.ans*。

该样例满足测试点 16 的约束条件。

【样例 7】

见选手目录下的 `teleport/teleport7.in` 与 `teleport/teleport7.ans`。
该样例满足测试点 17 ~ 20 的约束条件。

【数据范围】

- 对于所有测试数据，均有：
- $2 \leq n \leq 5 \times 10^5, 1 \leq m \leq 10^6$;
 - 对于所有 $0 \leq i < n - 1$ ，均有 $0 \leq u_i, v_i < n$ ，且所有 (u_i, v_i) 构成一棵树；
 - 对于所有 $0 \leq i < m$ ，均有 $0 \leq x_i, y_i < n$ 且 $x_i \neq y_i$ 。

测试点编号	$n \leq$	$m \leq$	特殊性质
1	4	20	无
2, 3	5	30	
4 ~ 6	10^2	1	
7, 8	10^3	2, 000	A
9		10^6	无
10, 11	A		
12 ~ 15	无		
16	5×10^5	5×10^5	B
17 ~ 20		10^6	无

特殊性质 A：对于所有 $0 \leq i < n - 1$ ，均有 $u_i = i$ 且 $v_i = i + 1$ 。
特殊性质 B：对于所有 $0 \leq i < m$ ，均有 $y_i = 0$ 。