

SDWC Day4

吴清月

February 10, 2022

中文题目	序列函数	构树	二分图
英文题目	Function	Tree	Tree
程序名	function.c/cpp	tree.c/cpp	graph.c/cpp
输入文件	function.in	tree.in	graph.in
输出文件	function.out	tree.out	graph.out
每个测试点时限	1.0 s	1.0 s	1.0 s
测试点数目	10	10	10
每个测试点分值	10	10	10
附加样例文件	有	有	有
题目类型	传统	传统	传统
运行内存上限	512M	512M	512M

注意事项:

1. 无特别声明时, 结果比较方式为逐行比较模式 (忽略多余空格和制表符).
2. C++ 语言源程序名称为题目名.cpp.
3. 编译命令为 `g++ -o 题目名 题目名.* -lm -Wl,--stack=512000000 -O2 -std=c++11`. 即评测时开启 **O2** 优化开关. 开启无限栈. 开启 **C++11**. 由于 11 特性导致的编译错误可以申请重测.
4. 文件名 (程序名和输入输出文件名) 必须使用英文小写.
5. 函数 `main()` 的返回值类型必须是 `int`, 程序正常结束时返回值必须是 0.
6. 评测在 Windows 11 环境下用 Lemon 进行, 对于 `long long` 类型使用 `%lld`. 实际评测时限可能更改.

Problem A. 序列函数 (function.c/cpp)

Input file: function.in
 Output file: function.out
 Time Limit : 1.0 second
 Memory Limit: 512 megabytes

有这么一个定义在序列上的函数 f ，它的计算方法如下：

$$f(a_1, a_2, \dots, a_n) = \begin{cases} a_1 & \text{if } n = 1 \\ f(a_1 \oplus a_2, a_2 \oplus a_3, \dots, a_{n-1} \oplus a_n) & \text{else} \end{cases}$$

其中 \oplus 表示按位异或。

对于给定的数列 a_1, a_2, \dots, a_n ，你需要回答 m 组询问，每次询问一个区间 $[l, r]$ ，你需要计算下面这个式子的结果：

$$\max_{l \leq i \leq j \leq r} f(a_i, a_{i+1}, \dots, a_j)$$

Input

第一行两个整数 n, m 。
 接下来一行 n 个整数 a_1, a_2, \dots, a_n 。
 接下来 m 行，每行两个整数 l, r 表示一次询问。

Output

对于每次询问，输出一行一个整数，表示答案。

Examples

function-sample0.in	function-sample0.ans
10 5	7
1 2 3 4 5 6 7 8 9 10	15
1 4	15
2 9	6
3 8	15
5 6	
7 10	

Notes

数据规模与约定：

对于 30% 的数据，保证 $n, m \leq 50$ ；

对于 50% 的数据，保证 $n, m \leq 300$ ；

对于 100% 的数据，保证 $1 \leq n \leq 1000, 0 \leq m \leq 10^6, 0 \leq a_i \leq 2^{64} - 1$ 。

Problem B. 构树 (tree.c/cpp)

Input file: tree.in
 Output file: tree.out
 Time Limit : 1.0 seconds
 Memory Limit: 512 megabytes

有一天，你得到了一个 n 个点的树。

你感到很开心，于是记了下来和每一个点相邻的所有点中，编号最小的那个点和编号最大的那个点是什么。

然而一段时间后，你把那棵树弄丢了，但是你记录的信息还保存着。

你想得到原来的那棵树，于是尝试构造一个满足条件的解。

Input

第一行一个整数 n ，表示树的点数。

接下来 n 行，每一行两个整数，表示和这个点相连的编号最小的点和编号最大的点。

Output

如果没有解，输出“-1”（不含双引号）。

否则，输出 $n - 1$ 行，每一行两个整数 (a, b) ，描述一条边。

如果有多种情况满足条件，你只需要输出任意一组合法解即可。

Examples

tree-sample0.in	tree-sample0.ans
6	1 5
5 5	2 5
5 5	3 4
4 4	4 5
3 5	5 6
1 6	
5 5	

tree-sample1.in	tree-sample1.ans
9	1 5
5 5	2 4
4 4	3 6
6 6	4 5
2 7	4 6
1 9	4 7
3 8	5 9
4 4	6 8
6 6	
5 5	

tree-sample2.in	tree-sample2.ans
3 2 3 1 3 1 2	-1

Notes

数据规模与约定：

对于 10% 的数据，保证 $1 \leq n \leq 3$ ；

对于 30% 的数据，保证 $1 \leq n \leq 10$ ；

对于另外 20% 的数据，保证存在一组解，使得树是一条链；

对于 100% 的数据，保证 $1 \leq n \leq 1000$ 。

Problem C. 二分图 (graph.c/cpp)

Input file: graph.in
 Output file: graph.out
 Time Limit : 1.0 seconds
 Memory Limit: 512 megabytes

从前有一张 n 个点, m 条边的二分图。
 你觉得这张图太过稀疏了, 于是想往这张图里添加一些边。
 为了方便自己写匹配算法, 你希望在添加完边之后, 图仍然是二分图。
 求最多能够添加多少条边。

Input

第一行两个整数 n, m , 表示图的点数和边数。
 接下来 m 行, 每行两个整数 u, v , 描述一条边。
 保证给出的是二分图。

Output

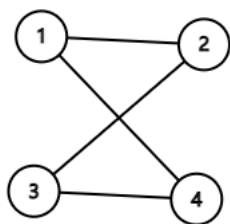
输出一行一个整数, 表示最多还能添加多少条边。

Examples

graph-sample0.in	graph-sample0.ans
4 2 1 2 3 4	2

样例解释:

一种可能的解如下:



Notes

数据规模与约定:

对于 20% 的数据, 保证 $n \leq 10$;

对于另外 20% 的数据, 保证给出的图连通且为一棵树;

对于另外 10% 的数据, 保证 $m = 0$;

对于另外 20% 的数据, 保证连通块数量 ≤ 3 ;

对于 100% 的数据, 保证 $1 \leq n \leq 5000, 0 \leq m \leq \frac{n(n-1)}{2}$ 。