

慈溪中学 NOIP2018 提高组模拟二

慈中 7 月赛 Day2

(请选手务必仔细阅读本页内容)

一、题目概览

中文题目名称	旅行问题	离散对数	通信网络
英文题目名称	travel	logarithm	communication
可执行文件名	travel	logarithm	communication
输入文件名	travel.in	logarithm.in	communication.in
输出文件名	travel.out	logarithm.out	communication.out
每个测试点时限	1 秒	1 秒	2 秒
测试点数目	10	10	20
每个测试点分值	10	10	5
比较方式	全文比较 (过滤行末空格及文末回车)		
题目类型	传统	传统	传统
运行内存上限	256M	256M	256M

二、提交源程序文件名

对于 Pascal 语言	travel.pas	logarithm.pas	communication.pas
对于 C 语言	travel.c	logarithm.c	communication.c
对于 C++语言	travel.cpp	logarithm.cpp	communication.cpp

三、编译命令 (不包含任何优化开关)

对于 Pascal 语言	fpc travel.pas	fpc logarithm.pas	fpc communication.pas
对于 C 语言	gcc -o travel travel.c -lm	gcc -o logarithm logarithm.c -lm	gcc -o communication communication.c -lm
对于 C++语言	g++ -o travel travel.cpp -lm	g++ -o logarithm logarithm.cpp -lm	g++ -o communication communication.cpp -lm

注意事项:

1. 文件名 (程序名和输入输出文件名) 必须使用小写。
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. 提交文件时请以自己的姓名建立目录, 并为每道题单独建立子文件夹。

1. 旅行问题

(travel.pas/c/cpp)

【题目描述】

John 打算驾驶一辆汽车周游一个环形公路。公路上总共有 n 车站，每站都有若干升汽油（有的站可能油量为零），每升油可以让汽车行驶一千米。John 必须从某个车站出发，一直按顺时针（或逆时针）方向走遍所有的车站，并回到起点。在一开始的时候，汽车内油量为零，John 每到一个车站就把该站所有的油都带上（起点站亦是如此），行驶过程中不能出现没有油的情况。

请你判断以每个车站为起点能否按条件成功周游一周。

【输入数据】

输入文件名为 travel.in。

输入文件第一行是一个整数，表示环形公路上的车站数。

接下来 n 行，每行两个整数。第 $i+1$ 行含有 p_i 和 d_i ， p_i 表示第 i 号车站的存油量， d_i 表示第 i 号车站到下一站的距离（为了方便，不妨设顺时针走时， d_i 表示 i 到 $i+1$ 的距离，第 n 站的下一站为第 1 站；逆时针走时， d_i 表示 i 到 $i-1$ 的距离，第 1 站的下一站为第 n 站）。

【输出数据】

输出文件名为 travel.out。

输出文件共 n 行，如果从第 i 号车站出发，一直按顺时针（或逆时针）方向行驶，能够成功周游一圈，则在第 i 行输出“TAK”，否则输出“NIE”。

【输入输出样例 1】

travel.in	travel.out
5	TAK
3 1	NIE
1 2	TAK
5 2	NIE
0 1	TAK
5 4	

【输入输出样例 2】

输入输出样例 2 规模较大，详见选手目录下的 `travel/travel2.in` 和 `travel/travel2.out`。

【数据规模与约定】

对于不同的测试数据，其数据规模及约束条件如下：

测试点编号	n	pi, di
1	≤ 50	≤ 20000
2		
3	≤ 2000	
4		
5	≤ 100000	
6		
7		
8		
9	≤ 500000	
10		

另外对于所有测试数据，保证 n 正整数且大于 1，pi, di 均非负。

【题目来源】

POI 2004

2. 离散对数

(`logarithm.pas/c/cpp`)

【题目描述】

小 H 最近刚学了对数运算：对于正实数 a 和 b ，若有 $a^x = b$ ，则 $\log_a b = x$ 。但实数域的对数运算对他来说太简单了，于是他想把对数运算推广到整数域内。

在整数中，离散对数 (Discrete logarithm) 是一种基于同余运算和原根的一种对数运算。其定义为：在模 m 的整数群 G 中，对于给定的 a 和 b （其中 a 是模 m 的一个本原根，即 $\exists x_0 \in \mathbb{Z}^+, a^{x_0} \equiv 1 \pmod{m}$ ），存在一个数 x ，使得 $a^x \equiv b \pmod{m}$ ，则称 x 为 b 在模 m 下以 a 为底的离散对数，记为 $\text{Ind}_a b$ 。

现在给定 a, b 和 m ，请你求出 b 在模 m 下以 a 为底的离散对数。当然，由于整数同余性质，符合条件的 x 可能不止一个，此时请求出最小的非负整数解。

同时，为了简化题目，保证 $\gcd(a, m) = 1$ ，此时 a 一定为 m 的一个本原根。

【输入数据】

输入文件名为 `logarithm.in`。

输入文件的第一行为 T ，表示测试数据组数。

接下来的 T 行，每行包含三个正整数 a 和 b 和 m ，具体含义见题目描述。

【输出数据】

输出文件名为 `logarithm.out`。

输出文件包含 T 行，每行一个非负整数，表示离散对数；若不存在，则输出“NaN”。

【输入输出样例 1】

<code>logarithm.in</code>	<code>logarithm.out</code>
3	2
2 4 7	NaN
2 5 7	3
3 6 7	

对于第二组数据，由于 $2^x \bmod 7$ 只能取 1, 2, 4 这三个值，因此 $\text{Ind}_2 5$ 不存在。

【输入输出样例 2】

logarithm.in	logarithm.out
10	NaN
306924 332022 376921	2805
20743 9430 30197	NaN
230697 252685 535387	NaN
82324 915 189337	48343
94929 50874 97283	352628
203426 58924 437467	NaN
712730 499046 878821	NaN
285006 262944 478001	NaN
652655 567631 932917	115794
238581 177716 337793	

【数据规模与约定】

对于不同的测试数据，其数据规模及约束条件如下：

测试点编号	T	m	特殊性质
1	≤ 10	$\leq 10^6$	m 是质数
2			
3			
4			
5			无特殊性质
6			
7	≤ 50	$\leq 10^9$	m 是质数
8			
9			无特殊性质
10			

另外对于所有测试数据，保证 $a < m$ 且 $b < m$ ，同时 $\gcd(a, m) = 1$ 。

3. 通信网络

(communication.pas/c/cpp)

【题目描述】

H 国一共有 n 座城市，编号为 1 到 n 的正整数。现在 H 国要建设通信网络，将这 n 座城市通过 $n-1$ 条通信线路连接起来。为了保证所有城市之间都能相互通信，这 $n-1$ 条通信线路组成一个树状的通信网络。显然，对于任意两个城市 x 和 y ，它们之间的通信路径是唯一的。我们将直接相连的两座城市之间的通信线路称之为“信道”。由于信息传输的能力有限，每一条信道都有一个带宽上限值 P ，即所有使用这条信道进行通信的带宽之和不超过 P 。

经过多年的建设，H 国的通信网络终于进入测试阶段。现在共有 m 组通信的请求，每一组通信请求可以用一个三元组 (x_i, y_i, w_i) 来表示，其中 x_i 和 y_i 表示通信双方所在城市， w_i 表示完成这组通信所需要的带宽。若 $x_i = y_i$ ，则通信在同一座城市内部的完成，不会占用任何一条信道的带宽；若 $x_i \neq y_i$ ，那么这组通信需要占用 H 国的通信网络，即城市 x_i 与 y_i 之间的每一条信道都需要 w_i 的带宽。

当然，随着 H 国的科技水平发展，技术人员们开发出了一种“超级信道”。这种信道能够使相邻两座城市之间的带宽达到无穷大，但费用极其昂贵，因此只能将 K 个城市之间的信道升级为超级信道。

现在 H 国的技术人员想知道，合理地安排超级信道的位置能否全部满足 m 组通信请求。若所有请求均能被满足，则请你求出普通信道（除了超级信道外均为普通信道）中最大带宽的最小值；如果无法满足所有的通信请求，技术人员会优先满足编号小的通信请求，那么请你算出第一个无法被满足的通信请求编号的最大值。

【输入数据】

输入文件名为 communication.in。

输入文件的第一行包含四个整数 n 、 m 、 P 和 K ，具体含义见题目描述。

输入文件第二行的包含 $n-1$ 个正整数 f_i ，其中 i 为 2 到 n 的正整数，表示编号为 i 的城市与标号为 f_i 的城市之间相连，其中 f_i 为 1 到 $i-1$ 之间的正整数。

以后的 m 行，每行三个正整数 x_i, y_i 和 w_i ，含义见题目描述。

【输出数据】

输出文件名为 `communication.out`。

输出文件包含两行，每行各包含一个整数。

若 m 组通信均能满足，则输出的第一行为数字 0，第二行为普通信道最大带宽的最小值；若 m 组通信不能满足，则输出的第一行为数字 1，第二行为第一个无法被满足的通信请求的编号的最大值。

【输入输出样例 1】

<code>communication.in</code>	<code>communication.out</code>
5 3 2 0 1 2 3 4 2 4 1 3 5 1 1 4 1	1 3

本样例中城市 1 到 5 组成了一条单链，同时 $K=0$ ，即没有超级信道。第一组请求会占用 (2, 3) 和 (3, 4) 的带宽，第二组请求会占用 (3, 4) 和 (4, 5) 的带宽，第三组请求会占用 (1, 2)、(2, 3) 和 (3, 4)，带宽上限为 2，因此请求 3 不能满足。

【输入输出样例 2】

<code>communication.in</code>	<code>communication.out</code>
4 3 4 1 1 2 2 3 4 1 1 3 2 1 4 3	0 4

本样例中城市 2 与其他三座城市之间相连，同时 $K=1$ ，有一条超级信道。第一组请求会占用 (2, 3) 和 (2, 4) 的带宽值 1，第二组请求会占用 (1, 2) 和 (2, 3) 的带宽值 2，第三组请求会占用 (1, 2) 和 (2, 4) 的带宽值 3，将 (1, 2) 建为超级信道，则 (2, 3) 的带宽为 3，(2, 4) 的带宽为 4，能够满足要求。

【输入输出样例 3】

此样例的数据规模较大，具体见文件 `communication/communication3.in` 和 `communication/communication3.out`。

【数据规模与约定】

对于不同的测试数据，其数据规模及约束条件如下：

测试点编号	n	m	K	特殊性质
1	≤ 5	≤ 10	$\lt n-1$	f _i 随机生成
2	≤ 50	≤ 100		
3	≤ 150	≤ 300		
4	≤ 1000	≤ 2000		
5			$=0$	f _i =i-1
6			$\lt n-1$	
7	$=0$			
8	≤ 100000	≤ 200000	$\lt n-1$	
9			$=0$	
10			$\lt n-1$	
11			$=0$	
12				
13	≤ 30000	≤ 50000	$=0$	无特殊性质
14			$\lt n-1$	
15	≤ 50000	≤ 100000	$=0$	
16			$\lt n-1$	
17	≤ 100000	≤ 200000	$=0$	
18			$\lt n-1$	
19	≤ 200000	≤ 300000		
20				

另外对于所有测试数据，保证 N, M 为正整数, K 为非负整数, 且 $1 \leq w_i \leq 500$, $1 \leq P \leq 100000000$ 。

【额外说明】

此题评测时，C 和 C++ 语言栈空间的大小为 16MB，本地测试时请注意栈溢出情况，可在**编译命令**后加 “-Wl,--stack=16777216”。

Pascal 语言可以用 { \$M size } 命令调整栈空间大小 (size 请填入具体数字)，也可以自行模拟堆栈。

(OJ 是 linux 系统，默认栈空间较大，不用考虑此问题)