

NOI p 普及模拟赛

2019 年 9 月 27 日

题目名称	简单题	温暖题	送分题
题目类型	传统型	传统型	交互型
目录	easy	warm	score
可执行文件名	easy	warm	score
输入文件名	easy.in	warm.in	
输出文件名	easy.out	warm.out	
每个测试点时限	0.1s	0.4s	0.1s
内存限制	512MB	512MB	512MB
测试点/包数目	20	20	4
测试点是否等分	是	是	否

提交源程序文件名 注意事项：

对于 C++ 语言	easy.cpp	warm.cpp	score.cpp
对于 C 语言	easy.c	warm.c	

1. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
2. 结果比较方式为忽略行末空格、文末回车后的全文比较。
3. C/C++ 中函数 main() 的返回值类型必须是 int，值为 0。
4. 编译选项为-O2 -std=c++11
5. 如果对题目有疑问（如样例出锅），可以来 X503-B06 找出题人
6. **考试时间 8:00 至 13:00**

简单题 (easy)

题目描述

给定一颗有 n 个点的边有长度的树，求随机从一个点出发，每次随机一条经过次数不大于 1的边，并移动到另外那个端点上，求期望移动距离

输入格式

从 *easy.in* 中读入数据

- 第一行一个整数 n
- 接下来 $n - 1$ 行，每行三个数 $u\ v\ w$ 表示有一条连接 $u\ v$ 的长为 w 的边

输出格式

输出到 *easy.out* 中

- 一行一个实数，当且仅当你的绝对或相对误差不大于 10^{-3} 时视为正确

样例 1 输入

3
1 2 1
2 3 2

样例 1 输出

5.00000

样例 1 解释

- 1 号点出发期望长度为 4
- 2 号点出发期望长度为 6
- 3 号点出发期望长度为 5

样例 2

见选手文件夹下 *easy/easy2.in* 与 *easy/easy2.out*

样例 3

见选手文件夹下 *easy/easy3.in* 与 *easy/easy3.out*

样例 4

见选手文件夹下 *easy/easy4.in* 与 *easy/easy4.out*

样例 5

见选手文件夹下 *easy/easy5.in* 与 *easy/easy5.out*

子任务

对于所有数据 $2 \leq n \leq 5 \times 10^5$ 且 $0 \leq w \leq 10^2$

测试点	n	特殊约束
1	≤ 2	
2	≤ 5	
3		
4	≤ 20	
5		
6	≤ 2000	A
7		B
8		C
9		
10		AC
11		A
12		BC
13		B
14		C
15		
16		
17		
18		
19		
20		

其中特殊约束有：

- A: 所有 u, v 都满足 $v = u + 1$
- B: 满足树是一颗以 i 为根的二叉树
- C: 所有 $w = 1$

温暖题 (warm)

题目描述

给定 n, f_1, k 和 p, q, r 数组, 且 f 满足:

$$f_i = (r_i + \min_{1 \leq j < i} (p_i f_j + q_j)) \bmod k$$

求 f 数组

输入格式

从 *warm.in* 中读入数据

- 第一行三个整数 $n \ x \ k$, 然后 $f_1 = x \bmod k$
- 第二至四行, 每行 n 个整数, 分别表示数组 $p \ q \ r$

输出格式

输出到 *warm.out* 中

- 一行 n 个整数, 表示 f

样例 1 输入

5 5 3
5 9 4 8 7
5 4 1 9 8
10 5 4 6 7

样例 1 输出

2 1 0 1 2

样例 1 解释

$$f_1 = 5 \bmod 3 = 2$$

$$f_2 = ((2 \times 9 + 5) + 5) \bmod 3 = 1$$

$$f_3 = ((4 \times 1 + 4) + 4) \bmod 3 = 0$$

$$f_4 = ((8 \times 0 + 1) + 6) \bmod 3 = 1$$

$$f_5 = ((7 \times 0 + 1) + 7) \bmod 3 = 2$$

样例 2

见选手文件夹下 *warm/warm2.in* 与 *warm/warm2.out*

样例 3

见选手文件夹下 *warm/warm3.in* 与 *warm/warm3.out*

样例 4

见选手文件夹下 *warm/warm4.in* 与 *warm/warm4.out*

样例 5

见选手文件夹下 *warm/warm5.in* 与 *warm/warm5.out*

子任务

对于所有数据: $2 \leq n \leq 10^5$ $0 \leq f_1 \leq 10^9$ $1 \leq k \leq 10^9$ $0 \leq p_i, q_i, r_i \leq 10^9$

测试点	n	特殊约束
1	≤ 1000	
2		
3		
4		
5		
6		
7		$p_i \leq 10$
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		

送分题 (score)

题目描述

有一个取值范围为 $\{0,1\}$ 的数组 a

你需要通过有限次调用函数 `int query(int *b,int l)`, 得到 a
`int query(int *b,int l)` 中

- b 是你的询问, 它的取值范围是 $\{0,1\}$, 注意, 它下标 1 到 l
- l 是 b 的长度
- 返回值表示: b 是否是 a 的一个连续子序列
- 只要在有最后一次询问中 $l = n$ 且返回值为 1, 才时为正确
- 复杂度为 $O(n)$, 请注意: 时间限制为 **0.1s** (当询问数正确时交互库使用时间不大于 0.03s)

交互方式

你需要实现 `void run(int n)` 函数

你可以在你的程序中使用 `int query(int *b,int l)` 函数

如果需要调用, 请在文件头部加上 `#include "score.h"`

你不能在你的文件中定义 `main()`

提示

为了让大家在 A 完前两题后有题目打发时间,同时让大家熟悉的 HNOI,NOI 的非传统题型,良心的出题人在这放了一道简单的交互题。

如果你还是不知道交互题怎么写,可以这样理解:

```
/*
your solution
*/
int query(int *b,int l)
{
    ...
}
int main()
{
    ...
}
```

query,main 函数是交互库中已经完成了的,不需要自己额外定义

你只需要在 *your solution* 部分补充上你的程序(其中需要包括 *run* 函数)

并且除了不能与交互库所实现的 *query,main* 函数重名外,几乎不需要考虑交互库所带来的影响

你可以定义其它函数,变量,引用需要的头文件等,除上一句所说明的外和传统题的实现方式完全相同

(如果需要,可以用 *using namespace std;* 来避免每次用 *std::* 来使用 *std*)

下发文件中有 *score.h* 和 *grader.cpp*,还有一个示例文件 *sample.cpp*

你可以使用 *sample.cpp* 并加以修改(包括重命名),也可以自己根据理解完成

如果你还是不理解,可以向 B05 提出你的问题

编译方式为 `g++ score.cpp grader.cpp -o score -O2 -std=c++11`

grader.cpp 测试时和选手文件夹下的几乎相同,但**不要试图攻击 *grader.cpp***

你自己测试时输入格式为:

- 第一行两个整数 *n l*

- 第二行为 n 个整数，表示数组 a

`grader.cpp` 会输出以下信息：

- 一行两个整数，如果正确输出 $l\ cnt$ ，否则输出 $-1\ -1$

或者输出以下任意一个信息：

- $-2\ -2$ 表示你的询问范围不是 $\{0, 1\}$

计分方式

p 表示这个测试点你的分数与总分的比（分数即总分 $\times p$ 向下取整）， l 的意义见下文

如果：

- $cnt \leq l, p = 1$
- $cnt > l, p = \frac{1 - \sqrt{1 - (\frac{l}{cnt})^2}}{1.5}$

样例 1 输入

```
4 100
0 0 1 1
```

样例 1 输出

```
100 1
```

样例 1 解释

只调用了一次 `query({0,0,1,1},4)`

子任务

保证 a 数组随机生成

- 5pts : $n = 10, l = 1000$
- 10pts : $n = 100, l = 300$
- 15pts : $n = 100, l = 150$
- 70pts : $n = 1000, l = 1100$