

2019CSP-S 模拟赛第一试

(请选手务必仔细阅读本页内容)

一、题目概况

中文题目名称	追踪逃犯	放气球	防火长城
英文题目名称	pb	ball	gfw
可执行文件名	pb.exe	ball.exe	gfw.exe
输入文件名	pb.in	ball.in	gfw.in
输出文件名	pb.out	ball.out	gfw.out
时间限制	1s	1s	1s
空间限制	256MB	256MB	256MB
测试点数目	10	10	10
单测试点分值	10	10	10
比较方式	全文比较(忽略行末空格和结尾回车)	全文比较(忽略行末空格和结尾回车)	全文比较(忽略行末空格和结尾回车)
题目类型	传统	传统	传统

二、提交源程序文件名

对于 Pascal 语言	pb.pas	ball.pas	gfw.pas
对于 C 语言	pb.c	ball.c	gfw.c
对于 C++语言	pb.cpp	ball.cpp	gfw.cpp

三、编译命令

对于 Pascal 语言	fpc pb.pas	fpc ball.pas	fpc gfw.pas
对于 C 语言	gcc -o pb pb.c -lm	gcc -o ball ball.c -lm	gcc -o gfw gfw.c -lm
对于 C++语言	g++ -o pb pb.c -lm	g++ -o ball ball.c -lm	g++ -o gfw gfw.c -lm

四、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、选手提交以自己编号命名的文件夹,文件夹内包含 3 个源文件(.c,.cpp,.pas),并在文件夹下建立三个相应的子目录,并将 3 个对应的源程序分别放入对应的子文件夹中,另在子文件夹外放入 3 个源程序,所有名字必须使用小写。
- 3、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。
- 4、题目简单,请认真对待,争取三位数。
- 5、每道题源代码长度限制均为 50KB。
- 6、每道题的数据都有一定梯度。请尽量优化算法,争取拿高分。
- 7、评测在 linux 系统下全国评测机和 Windows 下的 lemon 分别评测。
- 8、编译时不打开任何优化选项。
- 9、题目不一定按照难度顺序,但保证第一题最简单。

追踪逃犯 pb

问题描述

CJ 市的大型监狱发生了严重的越狱事件，一个世界级的超级恐怖分子，编号 745271940 成功从狱中逃脱。鉴于该事件影响力过大，市长决定向 Arthur Charles Clarke 爵士与 Langly 博士借用 H AL9000 对恐怖分子进行全方位搜索。

因为监狱在每个囚犯不知情的前提下加入了一个封闭的卫星蜂窝网络终端，所以可以通过类似于手机在蜂窝网络中的定位一样搜寻到逃犯的位置。蜂窝网络已经得知终端处于 n 个区域中，分别为 C_1, C_2, \dots, C_n 。由于在精确搜寻前进行了预搜索，对每个区域都有一个估价 U_i ，终端在 C_i 的概率为 $P_i = \frac{U_i}{U_1 + U_2 + \dots + U_n}$ 。最简单的搜索方法是在这些区域内寻找

终端，但这样做资源浪费率过高，H AL 9000 此时发挥其高超的人工智能作用，想出了一个折中的办法：将这 n 个区域分成 w 组，对于每一组同时访问。比如，这个终端可能会在 5 个区域中，概率分别为 0.3, 0.05, 0.1, 0.3, 0.25 现在 $w=2$ ，则某一种访问方法为将其分为 $\{c_1, c_2, c_3\}$ 和 $\{c_4, c_5\}$ 两个部分。这样访问区域数的数学期望应该是： $3 \times (0.3 + 0.05 + 0.1) + (3 + 2) \times (0.3 + 0.25) = 4.1$ 。或者还可以分为 $\{c_1, c_4\}$ ， $\{c_2, c_3, c_5\}$ 两组，这样的数学期望应该是 $2 \times (0.3 + 0.3) + (3 + 2) \times (0.05 + 0.1 + 0.25) = 3.2$

就在即将开始测试的时候，Langly 博士发现了一个几乎是史无前例的问题。A AL 9000 的智能分组能力发生了史上第一次故障。所以 Langly 博士遗憾的表示：将 n 个区域分成 w 组只能自己做了。又幸而发现，H AL 9000 的其他智能功能还十分完美，他甚至还可以做到对得到的访问区域数的数学期望的最小值进行倒推分组。所以，我们只用找到这个最小值就可以了。

Input

第一行两个整数 n, w 。

第二行 n 个整数，第 i 个整数表示第 i 个区域的估价 u_i 。

Output

一行一个整数，表示访问区域数的期望最小值，保留四位小数。

Sample Input

```
5 2
30 5 10 30 25
```

Sample Output

```
3.2000
```

数据范围

对于 30% 的数据， $1 \leq n \leq w \leq 10$ 。

对于 100% 的数据， $1 \leq n \leq w \leq 100, 1 \leq U_i \leq 10000$

放气球 ball

题目描述

某人（此处略去名字）开发出了一种恶心的方式放气球，当然，这里认为，气球都是一个圆。

气球在地平面上会有一个固定点 $(x_i, 0)$ ，我们认为气球是一直与地面相切，且固定点为切点。

每个气球有一个值 r_i ，表示气球能够膨胀的最大的半径。

气球以固定点的 **x 坐标从小到大的顺序** 开始膨胀，对于一个气球。它停止膨胀时当且仅当它与某一个气球相切或者它的半径已经达到最大。

现在告诉你 n 个固定点，请你求出每个气球能够膨胀的半径的大小。

Input

第一行一个整数 n 。

接下来 n 行，每行两个整数 x_i 和 r_i 。

Output

输出文件一共有 n 行。

每行一个数，表示对于第 i 个气球，它能够膨胀的最大半径的大小。误差不超过 0.001。

Sample Input

```
3
0 9
8 1
13 7
```

Sample Output

```
9.000
1.000
4.694
```

数据范围

对于 40% 的数据， $n \leq 2000$ 。

对于 100% 的数据， $1 \leq n \leq 200000, 0 \leq x_i \leq 10^9, 1 \leq r_i \leq 10^9$

防火长城 gfw

问题描述

在最近一次战争中，国家被摧毁了。你，这个国家的主人，决定提升首都防火长城的级别。防火长城能力很大一部分是由一排的魔法塔决定的，这一排魔法塔从城市一直延续到北边荒野丛林中。你的顾问告诉你，防火长城的能力其实是只由魔法塔的一个特性决定的：就是最长的连续的并且魔法塔高度是递增的魔法塔序列的长度。（顾问给了一个冗长的解释，但你只要知道这与控制敌军翻墙能力有关就可以了）。

为了提升防火长城的防御等级，你开始了刻苦的谈判，法师最终同意解除一部分魔法塔的封印以便你拆除并提升防御等级。然后其余的魔法塔将会重新合并，并保持相对位置不变。现在，你可以拆除任意数量的魔法塔。但法师又露出了邪恶的笑容，缓缓说道：这些被拆除的塔必须是连续的。

请问，删除一些魔法塔后，新的魔法塔序列中最长的连续的并且魔法塔高度是递增的魔法塔序列的长度最大是多少？

Input

第一行一个整数 n ，表示原来有 n 个魔法塔。

第二行 n 个整数，为依次的每个魔法塔的高度 H_i

Output

一行一个整数，为删除一些塔甚至是不删后最长连续且高度递增的魔法塔序列的长度。

Sample Input

9

5 3 4 9 2 8 6 7 1

Sample Output

4

样例解释

删除 9, 2, 8 后，最长连续且高度递增的序列为 3, 4, 6, 7 长度为 4。

数据范围

对于 30% 的数据， $n \leq 10$ 。

对于 50% 的数据， $n \leq 100$ 。

对于 100% 的数据， $n \leq 200000$ 。