

2024 NOIP模拟赛

by GTYZ

题目名称	ti	tii	tiii	tiv
题目类型	传统型	传统型	传统型	传统型
目录	ti	tii	tiii	tiv
可执行文件名	ti	tii	tiii	tiv
输入文件名	ti.in	tii.in	tiii.in	tiv.in
输出文件名	ti.out	tii.out	tiii.out	tiv.out
每个测试点时限	1.0秒	2.0秒	2.0秒	3.0秒
内存限制	512M	512M	512M	512M
测试点数目	5	2	4	6
测试点是否等分	是	否	否	否

提交源程序文件名

对于C++语言	ti.cpp	tii.cpp	tiii.cpp	tiv.cpp
---------	--------	---------	----------	---------

编译选项

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

注意事项（请仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- 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。
- 提交的程序代码文件的放置位置请参考各省的具体要求。
- 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
- 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
- 选手提交的程序源文件必须不大于 100KB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。
- 全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。
- 只提供 Linux 格式附加样例文件。
- 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

ti (ti)

1s, 512MB

题目描述

给定一个长度为 n 的排列 a_1, a_2, \dots, a_n , 有以下几种操作:

1. 将 a_{n-1} 拿出来, 放到序列开头。即 $a_1, a_2, \dots, a_{n-1}, a_n$ 变成 $a_{n-1}, a_1, a_2, \dots, a_{n-3}, a_{n-2}, a_n$ 。
2. 连续进行任意次操作 1。
3. 将 a_1 拿出来, 放到序列末尾。即 $a_1, a_2, \dots, a_{n-1}, a_n$ 变成 $a_2, a_3, \dots, a_{n-1}, a_n, a_1$ 。

问只进行任意次操作 2 和操作 3, 最少需要多少次操作 2 才能将原排列按照升序排序。

输入格式

第一行一个整数 n 。

第二行 n 个整数, 第 i 个整数表示 a_i 。

输出格式

一行一个整数, 表示最少需要几次操作 2 才能将原排列排序。

输入输出样例

```
1 | 6
2 | 2 4 5 1 3 6
```

```
1 | 2
```

样例解释

- 进行 5 次操作 3, 排列变成 6, 2, 4, 5, 1, 3。
- 进行 1 次操作 2 (3 次操作 1), 排列变成 4, 5, 1, 6, 2, 3。
- 进行 4 次操作 3, 排列变成 2, 3, 4, 5, 1, 6。
- 进行 1 次操作 2 (1 次操作 1), 排列变成 1, 2, 3, 4, 5, 6。

容易发现不存在更优的解。

数据范围

- subtask1(20pts): $n \leq 6$ 。
- subtask2(20pts): $n \leq 10$
- subtask3(20pts): $n \leq 20$
- subtask4(20pts): $n \leq 500$
- subtask5(20pts): $n \leq 2000$ 。

对于所有数据, 保证 $1 \leq n \leq 2000$, a 是一个排列。

tii (tii)

2s,512MB

题目描述

有一个由 n 个 0 或 1 组成的序列 a_1, a_2, \dots, a_n 。请找到一个区间，使得其中 1 所占比例尽可能接近给定小数 p 。即该区间 1 的个数除以区间长度最接近 p 。输出该区间的左端点，如果有多个可能的答案，输出最小的那个。

多测。

输入格式

第一行一个整数 T ，表示数据组数。

对于每组数据，第一行一个整数和一个在 $[0, 1]$ 范围的小数，分别表示 n 和 p ；第二行一个由 0 或 1 组成的字符串，第 i 个字符表示 a_i 。

保证 p 小数点后的位数不超过 5 位。

输出格式

对于每组数据，输出一行一个整数，表示最小的左端点。

输入输出样例

```
1 4
2 7 0.5
3 1101110
4 11 0.8
5 11111011110
6 9 0.75
7 011001011
8 9 0.666
9 010111100
```

```
1 2
2 2
3 6
4 1
```

样例解释

区间分别为 $[2, 3], [2, 6], [6, 9], [1, 6]$ 。

数据范围

- subtask1(40pts): $\sum n \leq 5000$
- subtask2(60pts): $\sum n \leq 5 \times 10^5$ 。

对于所有数据，保证 $1 \leq n, \sum n \leq 5 \times 10^5$ ，字符串每个位置是 0 或 1， p 在 $[0, 1]$ 范围内。

tiii (tiii)

2s,512MB

题目描述

在数轴上有 n 只鸡，第 i 只鸡初始在位置 a_i 。你需要捉鸡。

你初始在原点 0 。任意时刻，第 i 只鸡都会以每秒 b_i 个单位的速度朝着远离你的方向逃跑。你可以以每秒 C 个单位 ($b_i < C$) 的速度去捉鸡。如果任意时刻你和某只鸡在同一位置，那么你就可以不花费时间把这只鸡捉住。

你可以在任意时刻任意改变方向。求捉住所有鸡的最小时间。容易发现一定能捉到所有鸡。

多测。

输入格式

第一行一个整数 T ，表示数据组数。

对于每组数据，第一行两个整数，表示速度 C 和鸡数 n 。

第二行 n 个整数，表示第 i 只鸡的初始位置 a_i 。

第三行 n 个整数，表示第 i 只鸡的速度 b_i 。保证 $b_i < C$ 。

输出格式

对于每组数据，输出一行一个浮点数，表示最少需要多少秒。你只需要保证你的答案和标准答案的相对误差或绝对误差小于 10^{-6} 即可。

输入输出样例

```
1 | 2
2 | 4 3
3 | 3 6 9
4 | 3 2 1
5 | 2 2
6 | 1 -1
7 | 1 1
```

```
1 | 3.0000000
2 | 5.0000000
```

样例解释

对于第一组数据，直接往正方向一直走即可。

对于第二组数据，一种方法是，先往负半轴走 1 秒，再往正半轴走 4 秒。

数据范围

- subtask1(11pts): $1 \leq n \leq 9$ 。
- subtask2(17pts): $1 \leq n \leq 19$ 。
- subtask3(33pts): $1 \leq n \leq 80$ 。
- subtask4(39pts): 无特殊限制。

对于所有数据, 保证 $1 \leq T \leq 10, 1 \leq n \leq 500, 1 \leq b_i < C \leq 1000, -10^7 \leq a_i \leq 10^7$, 所有 $a_i \neq 0$ 。

tiv (tiv)

3s,512MB

题目描述

给定一棵 n 个点的无根树。令 u, v 间的距离为点 u 到点 v 的最短路径的边数。

有 Q 个询问，每个询问形如查询树上距离 x, y 最短路径上的任意一点的距离小于等于 d 的点的个数。注意，只需要满足距离其中某个点的距离小于等于 d 即可。

输入格式

第一行一个整数，表示树的点数 n 。

接下来 $n - 1$ 行每行两个整数 u, v ，表示树上存在一条连接 (u, v) 的边。

接下来一行一个整数 Q ，表示询问个数。

接下来 Q 行每行三个整数 x, y, d ，表示一次查询，意义见题面。

输出格式

共 Q 行，每行一个整数，第 i 行表示第 i 个询问的答案。

输入输出样例

1	6
2	1 2
3	2 3
4	2 4
5	1 5
6	4 6
7	3
8	2 4 1
9	1 1 0
10	3 2 1

1	5
2	1
3	4

数据范围

- subtask1(11pts): $n, q \leq 5000$ 。
 - subtask2(12pts): $x_i = y_i$ 。
 - subtask3(14pts): $x_i \rightarrow y_i$ 链上最多只会包含 20 个点。
 - subtask4(18pts): $d_i \leq 20$ 。
 - subtask5(22pts): $n, q \leq 5 \times 10^4$ 。
 - subtask6(23pts): 无特殊限制。
- 对于所有数据，保证 $1 \leq n, q \leq 2 \times 10^5$, $1 \leq u, v, x, y \leq n$, $0 \leq d \leq n - 1$ 。