

2023 CCF 非专业级软件能力认证模拟赛

提高组第一场

2023/7/4

(请选手务必仔细阅读本页内容)

| | | | | |
|---------|-----------|--------------|----------|------------|
| 题目名称 | 面包箱 | 异或三角形 | 种树 | 邪恶入侵 |
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 英文题目名称 | bread | triangle | tree | invade |
| 输入文件名 | bread.in | triangle.in | tree.in | invade.in |
| 输出文件名 | bread.out | triangle.out | tree.out | invade.out |
| 输出文件名 | 1s | 1s | 1.5s | 3s |
| 内存上限 | 512M | 512M | 512M | 512M |
| 测试点数目 | 10 | 20 | 20 | 每档部分分捆绑测试 |
| 每个测试点分值 | 10 | 5 | 5 | / |
| 附加样例文件 | 有 | 有 | 有 | 有 |
| 结果比较方式 | 全文比较 | 全文比较 | 全文比较 | 全文比较 |

注意事项：

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 `main()` 的返回类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
3. 提交的程序代码文件放置位置请参考各省的具体要求。
4. 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。
5. 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）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 全国统一评测时采用的机器配置为：Inter(R) Core(TM) i7-8700K CPU@3.70GHz，内存 32G。上述时限以此配置为准。

9. 只提供 Linux 格式的附加样例文件。
10. 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。

1 面包箱

(bread.cpp/c)

【题目描述】

这是一道签到题。

藤藤最近购买了一个面包箱，他会向面包箱中**依次**放入 n 个面包，每个面包有参数 a_i, b_i 。

每次放入一个面包后，藤藤就会思考面包箱中面包们的美味差异，他希望从面包箱中取出两个面包 i, j ，能得到的 $\frac{a_i - a_j}{b_i - b_j}$ 的最大值。

由于藤藤并不是那么聪明，于是这个问题就交给了睿智的你。

当面包箱中只有 1 个面包时，无需输出

【输入输出格式】

【输入格式】

从文件 *bread.in* 中读入数据。

第一行一个整数 n ，表示依次放入面包箱的面包个数。

之后 n 行每行两个整数 a_i, b_i ，表示这 n 个面包的两个参数。

【输出格式】

输出到文件 *bread.out* 中。

共一行 $n - 1$ 个用空格隔开的实数，表示每次加入面包后 $\frac{a_i - a_j}{b_i - b_j}$ 的最大值。

结果保留两位小数即可。

【样例】

【样例 1 输入】

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

【样例 1 输出】

-1.00 0.00 4.00 4.00 5.00 5.00 5.00 7.00 7.00

【样例 2】

见下发文件, 数据规模与性质与前 60% 的测试点相同。

【样例 3】

见下发文件, 数据规模与性质与另外 20% 的测试点相同。

【数据范围与提示】

对于 60% 的数据, $1 \leq n \leq 10^3$

对于另外 20% 的数据, $1 \leq b_i \leq n$, 即 b_i 为 n 阶全排列。

对于 100% 的数据, 保证 b_i 互不相同。

对于 100% 的数据, $1 \leq n \leq 10^5, 1 \leq a_i, b_i \leq 10^8$

2 异或三角形

(triangle.cpp/c/pas)

【题目描述】

题意超简单：给你一个数 n ，问：有多少对数 $0 \leq a, b, c \leq n$ 满足 $a \oplus b, b \oplus c, a \oplus c$ 三个数字构成一个非退化三角形，也就是两条短边大于第三边的长度。 \oplus 表示二进制下的异或操作。

【输入输出格式】

【输入格式】

从文件 triangle.in 中读入数据。

输入就一个数字 n ，表示给定的 n 在二进制下的表示。

【输出格式】

输出到文件 triangle.out

输出答案 $\text{mod } 998244353$ 。

【样例】

【样例 1 输入】

101

【样例 1 输出】

12

【样例 2 输入】

1110

【样例 2 输出】

780

【样例 3 输入】

11011111101010010

【样例 3 输出】

141427753

【数据范围与提示】

对于 10% 的数据满足： $0 \leq n \leq 2^{10}$ 。

对于 20% 的数据满足： $0 \leq n \leq 2^{20}$ 。

对于 100% 的数据满足： $0 \leq n \leq 2^{200000}$ 。

3 种树

(tree.cpp/c/pas)

【题目描述】

在室友 yql 的指引下, xjc 爱上了种树。

这天 xjc 得到了一种特殊的种子, 这些种子可以根据指令长出树, xjc 特别喜欢树, 所以他种下了 n 个种子排成一列, 编号为 $1 \sim n$ 。一开始所有的树只有根

yql 告诉 xjc 这些树非常特殊, 每一棵树由节点和边构成, 有一个特殊的点叫做生长节点。一开始所有的树只有一个节点, 我们称之为根, 此时生长节点就是根。我们定义一次“生长”为新长出一个节点把它与当前生长节点连边, 并且把生长节点替换为新长出的节点。

现在 xjc 要开始发号施令啦! 根据 yql 的指导, xjc 可以发布如下指示, 同时树苗会根据指令生长:

- 1 l r 表示编号在 $l \sim r$ 的树全部“生长”一次
- 2 l r 表示编号在 $l \sim r$ 的树的生长节点同时向根的方向移动一格, 保证他们的生长节点在移动前不是根

xjc 觉得光发号施令很没有意思, 于是他有的时候想要查询某一棵树当前生长节点的子树大小。子树大小的定义是断掉向根方向的边, 分裂出的不包含根的连通块的大小, 特别地, 根的子树大小就是树的节点个数。

作为 xjc 的小弟, 你对此很头疼, 但是看在 xjc 这么可爱的份上, 你还是打算帮他一把。

【输入输出格式】

【输入格式】

从文件 tree.in 中读入数据。

第一行两个整数 n, m , 表示树的个数以及 xjc 的操作次数

接下来是 m 行, 每一行包含一个操作, 符合下面的 3 个形式之一

- 1. 1 l r xjc 现在命令 $l \sim r$ 的所有树生长一次
- 2. 2 l r xjc 命令 $l \sim r$ 的所有树的生长节点向根移动一条边
- 3. 3 x xjc 现在想要知道第 x 棵树生长节点的子树的大小

【输出格式】

输出到文件 tree.out

对于每次查询输出一个结果

【样例】**【样例 1 输入】**

```
3 3
1 2 3
2 2 3
3 2
```

【样例 1 输出】

```
2
```

【数据范围与提示】

40% 的数据, $1 \leq n, m \leq 2000$, 无任何特殊性质

另有 20% 的数据, $1 \leq n, m \leq 5 \times 10^5$, 保证所有的树一旦生长节点上移了就不再生长

100% 的数据, 保证 $1 \leq n, m \leq 5 \times 10^5$,

4 邪恶入侵

(invade.cpp/c/pas)

【题目描述】

在无边无际的银河系中，广袤的宇宙就像一张 n 个点， m 条边相连的图，邪恶的加坦杰厄在 1 号节点建立了根据地，时间每过一年，邪恶的加坦杰厄都会进行根据地扩张。

也就是说在第 i 个年，所有和 1 号节点距离等于 i 的节点都会被邪恶的加坦杰厄扩张占据。

现在正义的迪迦奥特曼想在同一时间选择一个节点建立一个奥特根据地，正义的迪迦奥特曼占据节点的方式和邪恶的加坦杰厄遵循同一个形式。

邪恶的加坦杰厄和正义的迪迦奥特曼均不能去占据一个已经被对方占据的节点。

如果双方同时扩张到一个节点，那么这个节点就继续保持中立，均不属于双方。

现在正义的迪迦奥特曼想知道，通过合理选择初始节点，邪恶的加坦杰厄最少占领多少个节点？

正义的迪迦奥特曼的初始节点不能选择一号节点。

【输入输出格式】

【输入格式】

从文件 *invade.in* 中读入数据

第一行两个正整数 n, m 。

接下来 m 行，每行两个整数 x, y 表示一条边。

数据保证 n 个点两两联通，且没有重边和自环。

初始是第 0 年。

【输出格式】

输出到文件 *invade.out* 中

一行一个整数表示答案。

【样例】

【样例 1 输入】

```
3 2
1 2
1 3
```

【样例 1 输出】

```
2
```

【样例 2 输入】

见下发文件

【样例 2 输出】

见下发文件

【数据范围与提示】

对于所有数据：

- $2 \leq n \leq 10^5$
- $n - 1 \leq m \leq n + 100$
- $1 \leq x, y \leq n$

| 测试点编号 | n | m |
|---------|-------------|----------------|
| 1 ~ 4 | $\leq 10^3$ | $\leq n + 100$ |
| 5 ~ 7 | $\leq 10^5$ | $= n - 1$ |
| 8 ~ 10 | $\leq 10^5$ | $= n$ |
| 11 ~ 14 | $\leq 10^5$ | $\leq n + 10$ |
| 15 ~ 20 | $\leq 10^5$ | $\leq n + 100$ |