

必可赛前公益众筹赛

第三试

时间：2024 年 9 月 30 日 08:00 ~ 12:00

题目名称	人机识别	字符画	能量	最小生成树
题目类型	传统型	传统型	传统型	传统型
目录	iamhuman	picture	energy	mst
可执行文件名	iamhuman	picture	energy	mst
输入文件名	iamhuman.in	picture.in	energy.in	mst.in
输出文件名	iamhuman.out	picture.out	energy.out	mst.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MB	512 MB	512 MB	1024 MB
测试点数目	20	20	20	6
测试点是否等分	是	是	是	否

提交源程序文件名

对于 C++ 语言	iamhuman.cpp	picture.cpp	energy.cpp	mst.cpp
-----------	--------------	-------------	------------	---------

编译选项

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

注意事项与提醒（请选手务必仔细阅读）

1. 选手请直接提交源程序至 becoder.com.cn 上的对应比赛。
2. 输入输出文件名必须使用英文小写。
3. 选手提交的源程序必须存放在**已建立好的，且带有样例文件和下发文件**的文件夹中，文件夹名称与对应试题英文名一致。
4. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
5. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
6. 若无特殊说明，结果比较方式为**忽略行末空格、文末回车后的全文比较**。
7. 程序可使用的栈空间大小与该题内存空间限制一致。
8. 在终端中执行命令 ulimit -s unlimited 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
9. 每道题目所提交的**代码文件大小限制为 100KB**。
10. 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。

11. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。

12. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。

13. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。

14. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。

15. **对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。**

人机识别 (iamhuman)

【题目描述】

只需要一个脚本，就可以让一个网站收到大量请求，进而可能导致服务器崩溃。

为了避免这种情况，多数网站都会在检测到异常访问时（或者总是）要求进行真人验证，比如图片识别、输入验证码。

这次，我们要访问的网站的验证码机制特别高级。它需要我们做出这样一个问题：

- 找到一个最短的只包含小写字母的字符串，使得其中 iamhuman 作为子序列的出现次数不小于 k 。

iamhuman 作为字符串 s ($|s| = n$) 的子序列的意思是，存在 $1 \leq b_1 < b_2 < b_3 < \dots < b_8 \leq n$ 使得 s_{b_1} 为字符 i, s_{b_2} 为字符 a, s_{b_3} 为字符 m, 以此类推。**出现次数就是不同的 b 序列的数量。**

比如说，abcbcbcb 这个字符串中，abcbcb 就出现了 5 次。同时这也是让 abcbcb 出现至少 5 次的最短字符串。

然而，你作为 OI 界响当当的人物，还是准备写一个脚本攻击这个网站。这就需要你写一个程序来动态地解决这个问题了！

【输入格式】

从文件 *iamhuman.in* 中读入数据。

输入数据仅一行，包含一个正整数 k ，表示你需要达到的 iamhuman 出现次数的下界。

【输出格式】

输出到文件 *iamhuman.out* 中。

输出一个字符串，为你找到最短的字符串使得 iamhuman 出现不下 k 次。如果有多个最短的字符串，输出**任意一个**即可。

数据保证答案字符串长度不超过 10^7 ，因此，所有长度超过 10^7 的输出都被视为非法。

【样例 1 输入】

1 1

【样例 1 输出】

1 iamhuman

【样例 2 输入】

1 3

【样例 2 输出】

1 iiiamhuman

【数据范围】

对于 20% 的数据, $k \leq 10$ 。

对于 60% 的数据, $k \leq 10^9$ 。

对于 100% 的数据, $1 \leq k \leq 10^{16}$ 。

字符画 (picture)

【题目描述】

由于长期的 OI 经历, 大 M 看过了许多「尖锐」的代码——普通的行长度正常, 然后一两行突然很长, 接下来又回归正常。也看过许多「肥大」的代码——每一行都很长, 就像键盘上没有换行键似的。大 M 真的受够了!

在 11 月的最后一天, 大 M 光荣地成为了同机房的大佬的分母。为了庆祝他脱离苦海, 你要送他一张字符画。这个字符画是直接画在 `picture.txt` 里面的, 由空格和 `#` 号组成, 并且需要满足以下条件:

- 字符画是连通的。也就是说, 从任何一个 `#` 号出发, 走若干步, 即使每一步只能走向相邻的 4 个位置中是 `#` 号的位置, 也能够走到任何一个 `#` 号。(注: 没有 `#` 号是画在边缘的, 所以每个 `#` 号都有 4 个相邻的位置。)
- 字符画并不「尖锐」, 即, 没有一个 `#` 号的相邻 4 个位置中只有 1 个 `#` 号。(否则画出来就是尖的啦。)
- 字符画并不「肥大」, 即, 没有一个 `#` 号的相邻 4 个位置中有 3 个 `#` 号。

由于你看不起愚蠢的大 M , 你不会花费太多心思。如果一个 `#` 号的相邻 4 个位置全都是 `#` 号, 它就是「耗心思」的。你希望「耗心思」的 `#` 号只有 **恰好** n 个。当然, 对于其他的 `#` 号的数量就没有限制了。

那么, 你送给大 M 的字符画到底可以是什么呢?

【输入格式】

从文件 `picture.in` 中读入数据。

输入数据仅一行, 包含一个正整数 n , 表示字符画中「耗心思」的 `#` 号的个数。

【输出格式】

输出到文件 `picture.out` 中。

为了方便, 你无需输出整个字符画, 而是输出每个 `#` 号的坐标 (x, y) , 表示第 x 行、第 y 列上有一个 `#` 号。行、列均从 0 开始编号, 你应当保证 x, y 为正整数, 因为 `#` 号不会被画在边缘。

第一行输出一个正整数 k , 为 `#` 号的数量。由于 Special Judge 有一定的实现难度, **不得不要求** $k \leq 900$, 希望选手见谅。

接下来的 k 行, 每行包含两个正整数 x, y , 含义见上。如果有多个可能的字符画, 只需要输出 **任意一个**。也烦请保证 $x, y \in [1, 10^7]$ 。

【样例 1】

见选手目录中的 `picture/ex_picture1.in` 与 `picture/ex_picture1.ans`。

【样例 1 解释】

见选手目录下的 `picture/picture1.txt` , 即 `picture/picture1.ans` 对应的字符画。

【样例 2】

见选手目录中的 `picture/ex_picture2.in` 与 `picture/ex_picture2.ans`。

【样例 2 解释】

见选手目录下的 `picture/picture2.txt`。

【数据范围】

对于全部数据, $1 \leq n \leq 300$, 请注意输出的 k 有范围限制。

对于第 1 ~ 4 个测试点, $n \leq 10$ 。

对于第 5 ~ 7 个测试点, $n \leq 127$ 。

对于第 8 ~ 10 个测试点, $n = 200$ 。

对于第 11 ~ 16 个测试点, $n \leq 298$ 。

对于第 17 ~ 20 个测试点, $n = 300$ 。

Windows 下, 选手可以编译选手目录下的 `picture/checker.cpp` 得到 `checker.exe` 。当选手运行程序, 从 `picture.in` 中读入数据、输出到 `picture.out` 后, 在命令行中输入 `checker picture.in picture.out` , 校验器会判断选手的方案是否合法。

Linux 下同理。

能量 (energy)

【题目描述】

小苗自家底下有一个核融合炉，里面住了一只地狱鸦灵乌路空。不过空并不是本题的主角，本题的主角是八坂神奈子。

神奈子最近提出了 n 项能源计划，第 i 项计划会先消耗 a_i 的能量，最终产生 b_i 的能量。**需要注意的是，我们并不保证 $a_i \leq b_i$ 总是成立。**

神奈子想从里面选出 k 项计划，并按一定的顺序实施。在整个过程中能量不能为负，所以需要有一个初始能量使得计划能够正常实施下去。

神奈子想要最小化初始能量。由于计划还处于初步商议阶段，所以神奈子想要对于每个 $k \in [1, n]$ 都计算出最小的初始能量。但由于神奈子忙着打麻将，所以她把任务交给了小苗计算；但是小苗也不会，所以她按响了你家的门铃。

【输入格式】

从文件 *energy.in* 中读入数据。

第一行包含一个正整数 n 。

接下来一行共 n 个正整数 a_1, a_2, \dots, a_n 。

接下来一行共 n 个正整数 b_1, b_2, \dots, b_n 。

为了避免较大的输入量造成的时间差异，你可以使用下发的 *fast_input.cpp* 中的读入函数来加快你的读入效率。

【输出格式】

输出到文件 *energy.out* 中。

输出一行 n 个整数，第 i 个整数表示当 $k = i$ 时的答案。

【样例 1 输入】

```
1 3
2 2 3 1
3 1 5 4
```

【样例 1 输出】

```
1 1 1 1
```

【样例 2 输入】

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

3 1 1 2 3

【样例 2 输出】

1 1 2 3 4

【样例 3】

见选手目录中的 `energy/ex_energy3.in` 与 `energy/ex_energy3.ans`。
该样例与测试点 6 ~ 8 限制相同。

【样例 4】

见选手目录中的 `energy/ex_energy4.in` 与 `energy/ex_energy4.ans`。
该样例与测试点 9 ~ 11 限制相同。

【样例 5】

见选手目录中的 `energy/ex_energy5.in` 与 `energy/ex_energy5.ans`。
该样例与测试点 12 ~ 16 限制相同。

【数据范围】

对于全部数据, $n \leq 3 \times 10^5, a_i, b_i \leq 10^9$ 。记 `cnt` 表示满足 $a_i > b_i$ 的 i 的数量。

对于第 1 ~ 2 个测试点, $n \leq 5, cnt \leq n$ 。

对于第 3 ~ 5 个测试点, $n \leq 20, cnt \leq n$ 。

对于第 6 ~ 8 个测试点, $n \leq 3000, cnt \leq n$ 。

对于第 9 ~ 11 个测试点, $n \leq 300000, cnt = 0$ 。

对于第 12 ~ 16 个测试点, $n \leq 300000, cnt \leq 1000$ 。

对于第 17 ~ 20 个测试点, $n \leq 300000, cnt \leq n$ 。

最小生成树 (mst)

【题目描述】

小 Y 给了你一张图，共有 $n + 1$ 个点，编号为 $0 \sim n$ ，一共有两类边：

n 条边，小 Y 会给你 n 个整数， a_1, a_2, \dots, a_n ，其中 a_i 表示 0 号节点与 i 号节点有一条边权为 a_i 的边。

m 条边，连接 $u, v (u, v \in [1, n])$ ，长度为 w 。

显而易见，这个图是一个联通图，你要求最小生成树。

但是不幸的是，小 Y 会做出 q 次修改，每次给出两个数 x, y ，表示将 a_x 改为 y ，然后你需要输出最小生成树大小。

注意，所有修改之间不独立。

【输入格式】

从文件 `mst.in` 中读入数据。

第一行一个正整数 n 。

第二行 n 个正整数 $a_1, a_2 \dots a_n$ ，表示第一类边。

接下来 m 行每行三个正整数 u, v, w ，表示第二类边。

接下来一行一个正整数 q ，表示询问个数。

最后有 q 行，每行两个正整数 x, y ，表示修改。

【输出格式】

输出到文件 `mst.out` 中。

q 行每行一个整数，分别表示第 i 个询问的答案。

【样例 1 输入】

```
1 9 6
2 121 105 110 106 105 110 114 117 110
3 4 5 100
4 5 2 190
5 1 9 132
6 9 7 2
7 9 8 210
8 6 3 128
9 6
10 1 321
11 2 213
12 3 932
13 4 21
```

14 4 239
15 5 114

【样例 1 输出】

1 891
2 976
3 994
4 910
5 994
6 1003

【样例 2】

见选手目录中的 `mst/ex_mst2.in` 与 `mst/ex_mst2.ans`。
该样例满足子任务 1 的性质。

【样例 3】

见选手目录中的 `mst/ex_mst3.in` 与 `mst/ex_mst3.ans`。
该样例满足子任务 2 的性质。

【样例 4】

见选手目录中的 `mst/ex_mst4.in` 与 `mst/ex_mst4.ans`。
该样例满足子任务 3 的性质。

【样例 5】

见选手目录中的 `mst/ex_mst5.in` 与 `mst/ex_mst5.ans`。
该样例满足子任务 4 的性质。

【数据范围】

对于全部数据, $n, m, q \leq 3 \times 10^5, 1 \leq a_i \leq 10^9, 0 \leq w \leq 10^9, 1 \leq x \leq n, 1 \leq y \leq 10^9$ 。

子任务 1 (20 分), $n, m, q \leq 2000$ 。

子任务 2 (5 分), $n, m, q \leq 3 \times 10^5, x = 1$ 。

子任务 3 (10 分), $n, m, q \leq 3 \times 10^5, x \leq 5$ 。

子任务 4 (20 分), $n, m, q \leq 3 \times 10^5, m = n - 1$, 保证第二类边构成了一条链。

子任务 5 (25 分), $n, m, q \leq 1 \times 10^5$ 。

子任务 6 (20 分), $n, m, q \leq 3 \times 10^5$ 。