

A 丑陋的鞋

时空限制：1s 256MB

文件名

cipele.in/cipele.out/cipele.cpp

题目描述

在各种各样的项目上消耗经费之后，Nadan 决定为软件开发者提供高质量的鞋子。

Nadan 在地下室找到了 N 只左脚鞋和 M 只右脚鞋。因为它们的来源未知，因此鞋子的大小也不尽相同。

Nadan 想让你配对尽可能多的鞋子（即在所有鞋子配对完毕后不能继续配对）。每一对应当包含一只左脚鞋和一只右脚鞋。当穿上一双鞋时，应当使得鞋子的丑陋度最小化。一双鞋子的丑陋度定义为：所有配对的鞋子中，左脚和右脚大小之差的绝对值的最大值。

输入格式

第一行输入正整数 N, M ，分别表示左脚鞋和右脚鞋的数量。

第二行输入 N 个整数 L_i ，表示左脚鞋的大小。

第二行输入 M 个整数 R_i ，表示右脚鞋的大小。

输出格式

输出所有配对方式中丑陋度的最小值。

样例 #1

样例输入 #1

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

样例输出 #1

```
0
```

样例 #2

样例输入 #2

```
4 3
2 39 41 45
39 42 46
```

样例输出 #2

```
1
```

样例 #3

样例输入 #3

```
5 5
7 6 1 2 10
9 11 6 3 12
```

样例输出 #3

```
4
```

提示

样例 2 解释

Nadan 的左脚鞋有 4 只，右脚鞋有 3 只，最多可以配成 3 对。一种配对方式为：39-46，41-42，45-39，第一双的两只鞋大小之差的绝对值最大，因而丑陋度为 7。

一种更好的配对方式为：39-39，41-42，45-46。该配对方式的丑陋度为 1，为所有配对方式中，丑陋度最小的。

数据规模与约定

- 对于 20% 的数据， $N = M$ 。
- 对于另外 50% 的数据， $N, M \leq 5000$ 。
- 对于 100% 的数据， $1 \leq N, M \leq 10^5$ ， $1 \leq L_i, R_i \leq 10^9$ 。

B 二叉树

时空限制：1s 256MB

文件名

setnja.in/setnja.out/setnja.cpp

题目描述

在二叉树中：

- 每个节点都有两个孩子——一个左孩子和一个右孩子。
- 如果节点标记为整数 x ，则其左子节点标记为 $2x$ ，右子节点标记为 $2x + 1$ 。
- 树的根标为 1。

在二叉树上从根开始遍历。遍历中的每一步要么是跳到左孩子上，要么是跳到右孩子上，或暂停休息（停留在同一节点上）。

用由字符 **L**，**R** 和 **P** 组成的字符串描述遍历过程。

- **L** 表示跳到左孩子；
- **R** 表示跳到右孩子；
- **P** 表示暂停一轮操作。

walk 的值是我们最终到达的节点的标签。例如，**LR** 的 *walk* 值为 5，而 **RPP** 的 *walk* 值为 3。

一次遍历由 **L**，**R**，**P** 和 ***** 描述。每个 ***** 可以是三个动作中的任何一个。例如，**L*R** 可能代表 **LLR**，**LRR** 和 **LPR**。集合 ****** 可能代表 **LL**，**LR**，**LP**，**RL**，**RR**，**RP**，**PL**，**PR** 和 **PP**。

最后，一次遍历后的 *walk* 的总值是该次遍历中所有可能的遍历顺序的每一步所形成的 *walk* 的值的总和。

计算给定遍历顺序后的 *walk* 的总值。

输入格式

一行一个字符串，表示遍历顺序。

输出格式

一行一个整数，表示 *walk* 的总值。

样例 #1

样例输入 #1

```
P*P
```

样例输出 #1

6

样例 #2

样例输入 #2

L*R

样例输出 #2

25

样例 #3

样例输入 #3

**

样例输出 #3

33

样例 #4

样例输入 #4

LLLLLRRRRRLLLLLRRRRRLLLLLRRRRRLLLLL

样例输出 #4

35400942560

提示

数据规模与约定

- 对于 30% 的数据，保证输入的字符串中无 * 字符。
- 对于 50% 的数据，保证输入的字符串中至多有三个 * 字符。
- 对于 100% 的数据，保证输入字符串长度小于 10000，字符串的每一位只可能是 L，R，P，*。

C 押韵

时空限制：1s 256MB

文件名

rima.in/rima.out/rima.cpp

题目描述

规定字符串 A, B 的最长公共后缀的长度为 $\text{LCS}(A, B)$ 。

当 $\text{LCS}(A, B) \geq \max(|A|, |B|) - 1$ 时，我们认为 A, B 两个字符串押韵。

给定 N 个字符串，要求从中组合出一个长度最长的字符串序列（序列长度为该序列所包含字符串的数量），使得序列中相邻两个字符串押韵。

输入格式

第一行，一个整数 N 。

接下来的 N 行，每行一个字符串。保证所有字符串互不相同，且总长度不超过 3×10^6 。

输出格式

输出字符串序列长度的最大值。

样例 #1

样例输入 #1

```
4
honi
toni
oni
ovi
```

样例输出 #1

```
3
```

样例 #2

样例输入 #2

```
5
ask
psk
krafna
sk
k
```

样例输出 #2

```
4
```

样例 #3

样例输入 #3

```
5
pas
kompas
stas
s
nemarime
```

样例输出 #3

```
1
```

提示

【样例 2 解释】

字符串序列 `ask-psk-sk-k` 长度最大，为 4。

【样例 3 解释】

没有任何两个字符串押韵，因此任何一个字符串都可以单独组成一个序列，答案为 1。

【数据规模与约定】

对于 30% 的数据， $N \leq 18$ 。

对于 100% 的数据， $1 \leq N \leq 5 \times 10^5$ 。

D 卡片

时空限制：5s 1024MB

文件名

sirmi.in/sirmi.out/sirmi.cpp

题目描述

有 N 张卡片，每张卡片上写有一个数值 P_i 。

可以连接任意两张卡片，但需要花费 $\min(P_a \bmod P_b, P_b \bmod P_a)$ 。

请你求出连接所有卡片所需要的最小花费。

输入格式

第一行，一个整数 N ，代表有 N 张卡片；

接下来 N 行，每行一个正整数 P_i ，代表第 i 张卡片上写的数值。

输出格式

一行，一个整数，代表最小花费。

样例 #1

样例输入 #1

```
4
2
6
3
11
```

样例输出 #1

```
1
```

样例 #2

样例输入 #2

```
4
1
2
3
4
```

样例输出 #2

```
0
```

样例 #3

样例输入 #3

```
3
4
9
15
```

样例输出 #3

```
4
```

提示

【样例解释 #1】

连接卡片 1 和卡片 2，花费 $\min(2 \bmod 6, 6 \bmod 2) = 0$ ；
连接卡片 2 和卡片 3，花费 $\min(3 \bmod 6, 6 \bmod 3) = 0$ ；
连接卡片 1 和卡片 4，花费 $\min(2 \bmod 11, 11 \bmod 2) = 1$ 。
总共花费 $0 + 0 + 1 = 1$ 。

【数据范围】

对于 30% 的数据， $1 \leq N \leq 10^3$ ；
对于另外 40% 的数据， $1 \leq P_i \leq 10^6$ ；
对于 100% 的数据， $1 \leq N \leq 10^5$ ， $1 \leq P_i \leq 10^7$ 。