
平方字符串 (square.cpp)

时间限制: 1s

空间限制: 512 MB

【问题描述】

如果一个字符串是连续写两次的字符串, 则该字符串称为平方字符串。例如, 字符串"aa","abcabc","abab"和"baabaa"都是平方字符串。但是字符串"aaa","abaaab"和"abcdabc"不是平方字符串。

对于给定的字符串 s , 确定它是否为平方字符串。

【输入格式】

第一行包含一个整数 $t(1 \leq t \leq 100)$ 表示测试数据的组数。

对于每组测试数据:

一行包括一个仅有小写字母构成的字符串 $S(1 \leq |S| \leq 100)$ 。

【输出格式】

对于每个测试数据, 输出一行一个字符串, 如果字符串 S 是平方字符串, 则输出"YES", 否则输出"NO"。

【输入样例】

```
10
a
aa
aaa
aaaa
abab
abcabc
abacaba
xxyy
xyyx
xyxy
```

【输出样例】

```
NO
YES
NO
YES
YES
YES
NO
NO
NO
YES
```

平方数和立方数 (cube.cpp)

时间限制: 1s

空间限制: 512 MB

【问题描述】

Polycarp喜欢是平方数和立方数的整数, 所以他所喜欢的整数可以构成这样的序列: 1,4,8,9,……

对于给定的数字 n , 请计算从1开始到 n , 这中间有多少个整数是被Polycarp喜欢的, 换句话说, 在区间 $[1,n]$ 中有多少个整数 x , 满足 x 是某个整数的平方或者某个整数的立方 (或同时是平方或立方)。

【输入格式】

第一行包含一个整数 $t(1 \leq t \leq 20)$ 表示测试数据的组数。

对于每组测试数据:

一行一个整数 $n(1 \leq n \leq 10^9)$

【输出格式】

对于每个测试数据, 输出一个整数表示Polycarp喜欢的从1到 n 的整数个数。

【输入样例】

```
6
10
1
25
1000000000
999999999
500000000
```

【输出样例】

```
4
1
6
32591
32590
23125
```

【数据范围及约定】

对于50%的数据, $1 \leq n \leq 1000$

禁止序列 (forbidden.cpp)

时间限制: 1s

空间限制: 512 MB

【问题描述】

给你两个仅有小写字母构成的字符串S和T。保证T是字符串"abc"的某种排列。现在你需要找到一个字符串S'，满足它是S的字典序最小的排列，并且T不是S'的子序列。

如果字符串a和b中每个不同字符的出现次数相同，则字符串a是字符串b的排列。

如果字符串a可以通过从b中删除若干个（可能是零个或全部）元素获得，则字符串a是字符串b的子序列。

字符串a在字典序上小于字符串b，当且仅当以下条件之一成立：

- a是b的前缀，但 $a \neq b$ ；
- 在a和b不同的第一个位置，字符串a的字母在字母表中出现的时间早于b中的对应字母。

【输入格式】

第一行包含一个整数t($1 \leq t \leq 1000$)表示测试数据的组数。

对于每组测试数据：

第一行包含一个字符串S ($1 \leq |S| \leq 100$)。

第二行包含一个字符串T，保证T是字符串"abc"的某种排列（显而易见， $|T|=3$ ）

【输出格式】

对于每个测试数据，输出一行一个字符串S'，满足它是S的字典序最小的排列，并且T不是S'的子序列。

【输入样例】

```
7
abacaba
abc
cccba
acb
dbsic
bac
abracadabra
abc
dddddddddddd
cba
bbc
abc
ac
abc
```

【输出样例】

```
aaaacbb
abccc
bcdis
aaaaacbbdr
dddddddddddd
bbc
ac
```

【样例解释】

对于第一个样例，"aaaabc"和"aaaabcb"在字典上都小于"aaaacbb"，但它们包含子序列"abc"。

对于第二个样例，“abccc”是“cccba”的最小字典序排列，且不包含子序列“acb”。

对于第三个样例，“bcdis”是“dbsic”的最小字典序排列，且不包含子序列“bac”。

排列问题 (permutation.cpp)

时间限制: 1s

空间限制: 512 MB

【问题描述】

Paprika喜欢排列。她有一个数组 a_1, a_2, \dots, a_n 。她想让数组变成整数1到n的排列。为了实现这一目标，她可以对数列执行若干次操作。在每次操作中，她可以选择两个整数 $i(1 \leq i \leq n)$ 和 $x(x > 0)$ ，然后执行 $a_i = a_i \% x$ （换句话说，用 a_i 除以 x 的余数替换 a_i ）。在不同的操作中，选择的 i 和 x 可以不同。

求出使数组成为整数1到n的排列所需的最小操作数。如果不可能，则输出-1。

排列是由n个从1到n的不同整数组成的任意顺序的数组。例如， $[2, 3, 1, 5, 4]$ 是一个排列，但 $[1, 2, 2]$ 不是置换（2在数组中出现两次）， $[1, 3, 4]$ 也不是一个排列（ $n=3$ ，但数组中有4）。

【输入格式】

第一行包含一个整数 $t(1 \leq t \leq 10^4)$ 表示测试数据的组数。

对于每组测试数据：

第一行包含一个整数 $n(1 \leq n \leq 10^5)$ 表示排列的长度。

第二行包含 n 个不同的整数 $a_1, a_2, \dots, a_n(1 \leq a_i \leq 10^9)$ 。

保证所有测试用例的 n 之和不超过 2×10^5 。

【输出格式】

对于每个测试数据，输出一行一个整数，表示最少的操作次数，如果不存在，则输出-1。

【输入样例】

```
4
2
1 7
3
1 5 4
4
12345678 87654321 20211218 23571113
9
1 2 3 4 18 19 5 6 7
```

【输出样例】

```
1
-1
4
2
```

【样例解释】

对于样例一，使操作次数最少的唯一可能的操作顺序是：

选择 $i=2, x=5$ ，执行 $a_2 = a_2 \% 5 = 2$ 。

对于样例二，不可能获得整数1到n的排列。

【数据范围及约定】

对于30%的数据， $1 \leq n \leq 1000$

数组操作 (array.cpp)

时间限制: 1s

空间限制: 512 MB

【问题描述】

给你一个长度为 n 的数组 a 以及一个整数 k , 满足 $2k \leq n$ 。

您必须对这个数组执行恰好 k 次操作。在一个操作中, 您必须选择数组中的两个元素 a_i 和 a_j (它们可以相等或不同, 但它们在数组中的位置不能相同, 即 $i \neq j$), 将它们从数组中删除, 并获得 $\lfloor a_i/a_j \rfloor$ 的分数, 其中 $\lfloor x/y \rfloor$ 是不超过 x/y 的最大整数。最初, 您的分数为0。当执行了恰好 k 次操作之后, 你需要将数组中所有剩余的元素添加到分数中。请计算你能得到的最低分数。

【输入格式】

第一行包含一个整数 t ($1 \leq t \leq 500$) 表示测试数据的组数。

对于每组测试数据:

第一行包含两个整数 n, k ($1 \leq n \leq 100, 0 \leq k \leq \lfloor n/2 \rfloor$)

第二行包括 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 2 \times 10^5$)

【输出格式】

对于每个测试数据, 输出一行一个整数, 即你可以获得的最小分数。

【输入样例】

```
5
7 3
1 1 1 2 1 3 1
5 1
5 5 5 5 5
4 2
1 3 3 7
2 0
4 2
9 2
1 10 10 1 10 2 7 10 3
```

【输出样例】

```
2
16
0
6
16
```

【样例解释】

对于第一个样例, 一种可以获得分数2的方法如下:

1. 选择 $a_7=1$ 和 $a_4=2$, 获得分数 $\lfloor 1/2 \rfloor=0$, 数组变成 $[1,1,1,1,3]$
2. 选择 $a_1=1$ 和 $a_5=3$, 获得分数 $\lfloor 1/3 \rfloor=0$, 数组变成 $[1,1,1]$
3. 选择 $a_1=1$ 和 $a_2=1$, 获得分数 $\lfloor 1/3 \rfloor=1$, 数组变成 $[1]$
4. 把剩余的元素1添加到总分中, 所以结果为2。

对于第二个样例, 无论你怎么进行操作, 结果为16。

对于第三个样例, 一种可以获得分数0的方法如下:

1. 选择 $a_1=1$ 和 $a_2=3$, 获得分数 $\lfloor 1/3 \rfloor=0$, 数组变成 $[3,7]$
2. 选择 $a_1=3$ 和 $a_2=7$, 获得分数 $\lfloor 3/7 \rfloor=0$, 数组变为空

因为数组为空, 所以不需要再进行任何操作了

对于第四个样例, 不需要进行操作, 所以最终的分数为 $4+2=6$

【数据范围及约定】

对于30%的数据, $1 \leq n \leq 10$

交易游戏 (trade.cpp)

时间限制: 1s

空间限制: 512 MB

【问题描述】

Monocarp 又一次在玩电脑游戏! 该游戏具有独特的交易机制。

如果想要与一个角色进行交易, Monocarp 必须选择他拥有的一件物品并将其换成另一个角色拥有的一件物品。每个物品都有一个整数价格。如果 Monocarp 选择的物品的价格为 x , 那么他可以用它换取价格不大于 $x+k$ 的任何物品 (仅能恰好是一件物品)。

Monocarp 最初有 n 件物品, 他拥有的第 i 件物品的价格是 a_i 。Monocarp 交易的角色有 m 件物品, 他拥有的第 i 件物品的价格是 b_i 。Monocarp 可以与该角色交易任意多次 (甚至可能为零次), 每次根据上述限制将他的一件物品与另一个角色的一件物品交换。请注意, 如果 Monocarp 在交换期间获得了某件物品, 他可以用它换取另一件物品 (因为现在该物品属于他), 反之亦然; 如果 Monocarp 想要之前他拥有的并被换走的某样物品, 他依然可以用他现在拥有的物品换回来。

您必须回答 q 次查询。每个查询由一个整数 k 组成。并要求您计算 Monocarp 在一系列交易后可能拥有的物品的最大可能总价格。假设在每一次交易中, 他都可以用一件价格为 x 的物品换取一件加个不超过 $x+k$ 的物品。请注意, 每次查询之间是独立的, 交易实际上并没有发生, Monocarp 只想计算他可以获得的最大总价格。

【输入格式】

第一行包括三个整数 n, m, q ($1 \leq n, m, q \leq 2 \times 10^5$)

第二行包括 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) 表示 Monocarp 拥有的物品的价格

第三行包括 m 个整数 b_1, b_2, \dots, b_m ($1 \leq b_i \leq 10^9$) 表示交易角色拥有的物品的价格

第四行包括 q 个整数, 第 i 个整数描述第 i 次询问的 k 值 ($0 \leq k \leq 10^9$)

【输出格式】

共 q 行, 对于每次询问, 一行一个整数, 表示 Monocarp 在若干次交易之后所拥有的物品的价格总和可能的最大值。

【输入样例】

```
3 4 5
10 30 15
12 31 14 18
0 1 2 3 4
```

【输出样例】

```
55
56
60
64
64
```

【数据范围及约定】

对于30%的数据, $1 \leq n, m \leq 100, 1 \leq q \leq 10$

对于另外30%的数据, $k \geq \max\{b_i\}$