

模拟赛

whzzt

Jan 1, 2019

题目名称	数字重排	硬币翻转	一道有趣的字符串题
输入文件名	sort.in	N/A	sam.in
输出文件名	sort.out	N/A	sam.out
时间限制	1s	10s	1s
是否捆绑测试	是	否	否
内存限制	1GB	1GB	1GB
是否有部分分	否	是	否
题目类型	传统	交互	传统
编译开关	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11

注意：1. 评测时的栈空间大小不做单独限制，但使用的总空间大小不能超过内存限制。

2. 考试之中不要交流，AK 之后不要大声喧哗，没有 AK 的同学请安静做题。

3. 不要倦生，不要高喊“这题太难我不会”之类的话。

4. 代码长度限制为 64 KB。

Problem A. 数字重排 (sort.c/cpp/pas)

Input file: `sort.in`
Output file: `sort.out`
Time limit: 1 seconds
Memory limit: 1 gigabytes

老虎最近在研究 Montgomery Modular Multiplication: 简单来说, 这是一种将取模运算转化为简单的乘法和位移的方法。

为了进行研究, 老虎随机了 n 个数 c_i , 并将取模运算写在了这些数字之间, 也就是 $c_1 \bmod c_2 \bmod \dots \bmod c_n$ 。随后老虎用 Montgomery Modular Multiplication 的方法高速地计算出了这个式子的结果。

显然, 蒜头和老虎的关注点不同: 蒜头想要知道, 任意重排序列 c 可能得到的最大结果是多少。

Input

输入的第一行包括一个正整数 n , 表示数列的长度。

接下来一行包括 n 个数, 表示数列中的元素。

Output

共一行一个数, 表示重排后最大的可能优秀度。

Examples

<code>sort.in</code>	<code>sort.out</code>
5 5 5 10 17 23	3

将序列重排为 23 10 5 17 5 即可。

Notes

对于 100% 的数据, $1 \leq n, c_i \leq 10^5$ 。下表是更详细的数据范围, 表中留空意为与以上数据范围相同。

子任务编号	子任务分值	$n \leq$	$c_i \leq$	特殊性质
1	13	10		
2	41		1000	
3	20			c_i 互不相同
4	26			

Problem B. 硬币翻转 (coin.cpp)

Input file: N/A
Output file: N/A
Time limit: 10 seconds
Memory limit: 1 gigabytes

这是一道交互题。

老虎最近沉迷一个翻硬币游戏。在游戏中，老虎有 100 枚硬币，每次老虎可以选择一枚硬币，并改变其状态——在正面朝上和背面朝上间发生变化。

显然，老虎和蒜头的关注点并不一样。蒜头在思考这样的游戏：蒜头会持有 100 枚不同的硬币，老虎每次可以给蒜头一个硬币的正反序列来猜测硬币的状态，随后蒜头会告诉老虎他猜对了多少枚硬币，并在 100 枚硬币中等概率选择一枚硬币。若老虎这一轮猜对了这枚硬币，它就会被翻转。老虎的目标是，在某一次完全猜对蒜头的硬币序列，也就是蒜头告诉老虎的答案会是 100。

你能帮助老虎完成这个问题吗？

Interaction Protocol

你的任务是实现一个函数：

```
void guess();
```

你可以调用下面的函数进行询问：

```
int ask(std::string coin)
```

你必须保证你传入的是一个长度为 100 的 01 字符串，其中 0 代表反面 1 代表正面，第 i 个字符表示第 i 种硬币的状态。该函数会返回蒜头的答案。当你给出的硬币状态和实际状态完全相同时，我们会结束程序，并按照你当前调用该函数的次数对你的得分进行计算。

你可以参考样例交互库对上述描述进行理解。

Notes

本题共包括 1 个测试点，我们会对你的函数进行不超过 100 次调用。如果某一次你失败了，你将得到 0 分的好成绩。否则，不妨设你最多一次调用了 T 次函数，设你的得分为 P ，评分标准如下：

$T \in$	$P =$
$[0, 200]$	100
$(200, 600]$	$120 - 0.15T$
$(600, \infty]$	$36 - 0.01T$

我们保证交互库占用的资源不会超过总资源的 10%。

在下发文件中，我们提供了 coin.h, grader.cpp, coin.cpp 三个文件，你最终提交的代码应当包含在 coin.cpp 中，并包含头文件 coin.h。

如果你想要命令行编译你的代码，应当执行：`g++ coin.cpp grader.cpp -o coin -O2 -std=c++11`。

Problem C. 一道有趣的字符串题 (sam.c/cpp/pas)

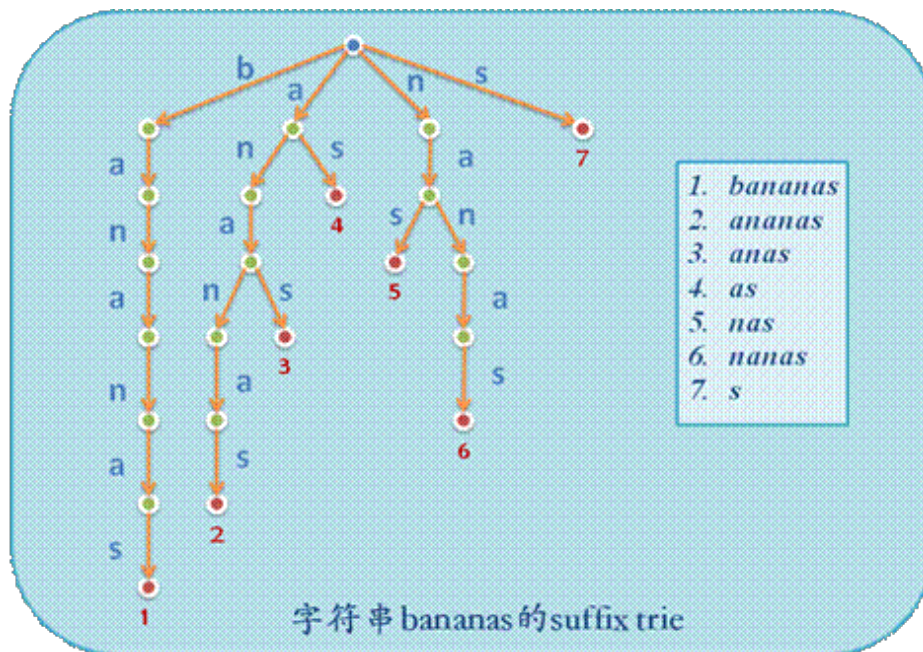
Input file: sam.in
 Output file: sam.out
 Time limit: 1 seconds
 Memory limit: 1 gigabytes

白云在学习后缀树。

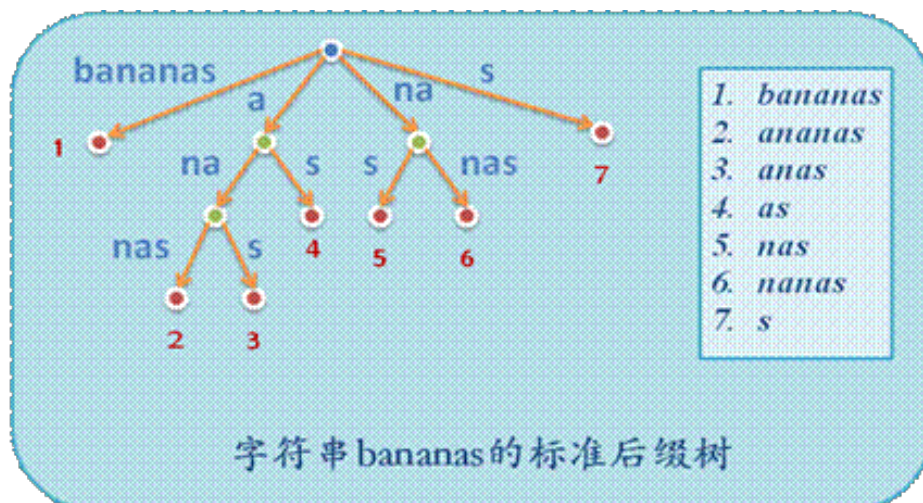
白云问白兔：后缀树是什么呢？

白兔回答：就是把一个字符串 S 的所有后缀拿出来建一棵 Trie 树，然后对于所有孩子数为 1 的结点，把他和他孩子合并。

比如串 “bananas” 的后缀组成的 Trie 树为



那么它的后缀树就为



白云问：天哪，把所有后缀拿出来都建 Trie，那节点数岂不是高达 $O(n^2)$ 个？

白兔回答：但是建出后缀树就只有 $O(n)$ 个了，不信你可以试试。

白云非常不服气，决定要找到反例，于是它给了长度为 n 字符集为 k 的所有字符串（共 k^n 个），想要知道这些字符串的后缀树结点分别是多少。然而数据量太大，白云无法计算出来，于是它决定求助于你。由于输出文件可能很大，你只需要输出所有字符串的后缀树节点数之和对 $10^9 + 7$ 取模后的值即可。

特别地，这里计算节点数时我们不忽略根节点。比如上述例子的答案为 11。

Input

输入的第一行包括一个正整数 T ，表示数据组数。

接下来 T 行，每行两个正整数 n 和 k 。

Output

输出 T 行表示答案。

Examples

sam.in	sam.out
3	12
2 2	447
4 3	14972
10 2	

Notes

对于 100% 的数据， $1 \leq k \leq n \leq 40, 1 \leq T \leq 820$ 。

本题共有 20 组测试数据，对于第 k 组数据有 $n \leq 2k$ 。