

# NOIP2017 Simulation

## Day1

Sunshine\_cfbsl

2017 年 9 月 16 日

题目名称	a	b	c
题目类型	传统型	传统型	传统型
目录	a	b	c
可执行文件名	a	b	c
输入文件名	a.in	b.in	c.in
输出文件名	a.out	b.out	c.out
每个测试文件时限	1.0s	1.5s	2.0s
内存限制	256MB	256MB	512MB
测试点数目	5	10	10
每个测试点分值	20	10	10
编译选项	-lm -O2		

**a**

(*a.cpp/c/pas*)

## 【问题描述】

*Polycarp*最近突然对表演比较感兴趣。

于是他自然来请教了擅长表演的嚟薙薙薙。

嚟薙薙薙自然是不会直接道出表演的真谛的，因为他认为这种行为是不符合演员的自我修养的。

所以苦恼的*Polycarp*只能通过在生活中观察嚟薙薙薙来学习表演艺术。

通过一段时间的观察，他集齐了嚟薙薙薙的全套表演动作。其中一套表演动作是由一连串有序的动作构成的。

但是*Polycarp*觉得他要原创一套动作，而不是完全仿照嚟薙薙薙。尽管如此，*Polycarp*并没有足够的创造力，他不能够创造单个动作，只能利用嚟薙薙薙已有的单个动作去创作新的一套动作。

现在*Polycarp*将每一个动作用一个写字母表示，并把所有*n*套动作列出了一个表。*Polycarp*认为如果一套动作字典序越靠前，那么这套动作越高超。

嚟薙薙薙明天会从这*n*套动作中选出几套（不能为0）来进行表演，*Polycarp*可以从嚟薙薙薙的动作中选几个动作排列来创作一段表演。而*Polycarp*的第一次创作是追求长度的，并且由于他的求知心是非常强的，现在他想知道明天他一定能够创作出的最长的一套动作。

如果有多套这样的动作，*Polycarp*想知道最高超的一套动作。

## 【输入格式】

从文件*a.in*中读入数据。

第一行一个正整数*n*，表示嚟薙薙薙的全套动作数目。

接下来*n*行，每行一连串用小写字母表示的动作，表示一套动作。

## 【输出格式】

输出到文件*a.out*中

一行，即*Polycarp*一定能够创作出的最长的一套动作。

**【样例输入1】**

3  
ybzby  
zycby  
gbzby

**【样例输出1】**

bzy

**【样例输入2】**

见选手目录下的 *a2.in*。

**【样例输出2】**

见选手目录下的 *a2.out*。

**【数据规模和约定】**

对于100%的数据，保证每一套动作长度小于或等于100。

测试点	n	其他条件
1	$\leq 10$	无
2	$\leq 100$	
3	$\leq 100000$	所有字母均为‘ <i>a</i> ’
4		无
5		

**b****(b.cpp/c/pas)****【问题描述】**

不久前 *Axel* 因为为全国青少年数据结构竞赛准备了很久，整个人都变得工业了起来。

于是 *Axel* 决定做一做清新一点题目。

*Axel* 开始从莫队、树状数组做起，一直做到了点分治、时间线段树。接着他又做了几道 CDQ 分治。

然而就在这时，他发现了一个他很久没有写过的另类数据结构。没错，那就是线性基。

“异或好啊！” *Axel* 想到。于是他瞟了一眼他以前做的一道题，然后就秒掉了。

现在他想测试一下你对线性基是否熟悉，于是他给了你一道题。

给定一个长度为  $n$  的数列  $\{u_i\}$  和一个长度为  $m$  的数列  $\{v_i\}$ ，定义矩阵  $A$ ，满足  $A_{i,j} = u_i \text{ xor } v_j$ 。求  $A$  的最大子矩阵异或和。

注：非负整数  $x$  和  $y$  的异或值等于  $x$  和  $y$  二进制每位分别异或的值。且  $1 \text{ xor } 1 = 0, 1 \text{ xor } 0 = 1, 0 \text{ xor } 1 = 1, 0 \text{ xor } 0 = 0$ 。在 **c/c++** 程序中，整数的异或可以直接使用运算符  $\wedge$ 。多个数的异或和为其依次异或后的值。

**【输入格式】**

从文件 **b.in** 中读入数据。

第一行两个正整数  $n, m$ ，表示数列  $\{u_i\}$  和数列  $\{v_i\}$  的长度。

第二行  $n$  个正整数，表示数列  $\{u_i\}$ 。

第三行  $m$  个正整数，表示数列  $\{v_i\}$ 。

**【输出格式】**

输出到文件 **b.out** 中。

一行一个整数，表示  $A$  的最大子矩阵异或和。

## 【样例输入1】

```

3 4
5 3 1
2 1 2 4

```

## 【样例输出1】

```

7

```

## 【样例输入2】

见选手目录下的***b2.in***

## 【样例输出2】

见选手目录下的***b2.out***。

## 【数据规模和约定】

对于100%的数据， $0 \leq u_i, v_i < 2^{29}$ 。

测试点	$n, m$	其他条件
1	$\leq 10$	无
2	$\leq 50$	
3		
4	$\leq 100$	
5	$\leq 200$	
6		
7	$\leq 1000$	$u_i = 0$
8		无
9	$\leq 2000$	
10		

**C***(c.cpp/c/pas)***【问题描述】**

$\frac{1}{4}$ 已经厌倦了和 $Tom$ 在树上的博弈了。

于是他们又回到了一维的博弈，与之前不同的是，他们这一次的博弈是在序列上进行的。

他们请求某位贪心大师作为他们的裁判，并获得了他的同意。

现在贪心大师将会随意给出一个长度为 $n$ 的序列 $\{a_n\}$ 。且保证不存在 $i, j$ ，使得 $a_i = a_j$ 。

$\frac{1}{4}$ 和 $Tom$ 每一轮都可以选出一个长度为 $4x+2$ 或 $4x+3$ 的区间（ $x$ 为任意自然数），并将其翻转。

贪心大师规定每一次翻转以后整个序列的字典序必须变大。而最后无法操作的人将输掉游戏。

现在 $\frac{1}{4}$ 作为先手，他想知道游戏结果。

当然， $\frac{1}{4}$ 和 $Tom$ 都足够聪明。

**【输入格式】**

从文件`c.in`中读入数据。

第一行一个整数 $T$ ，表示数据组数。

对于每一组数据，第一行一个整数 $n$ ，表示序列长度，第二行 $n$ 个整数，表示序列 $\{a_n\}$ 。

**【输出格式】**

输出到文件`c.out`中。

对于每一组数据，如果 $\frac{1}{4}$ 必胜，则输出 $Q$ ；

如果 $Tom$ 必胜，则输出 $T$ ；

否则输出湖南最强选手（贪心大师）的名字（每一个字首拼大写）。

**【样例输入1】**

1

4

4 2 1 3

### 【样例输出1】

$T$

### 【样例输入2】

1

5

1 9 2 8 4

### 【样例输出2】

$T$

### 【数据规模和约定】

对于100%的数据，满足 $n \leq 100, T \leq 10, 0 \leq a_i \leq 2147483647$ 。

测试点	n	T
1	$\leq 5$	$= 1$
2		$\leq 10$
3		
4	$\leq 10$	$= 1$
5		$\leq 10$
6	$\leq 50$	$= 1$
7		$\leq 10$
8	$\leq 100$	$= 1$
9		$\leq 10$
10		

## 一些有趣的事实

(*some\_interesting\_facts*)

- 发现水题切勿伸张~
- 不要被稍微长一些的题面吓到了，其实是我把字体调大了。（滑稽）
- 如果你看不懂第一题的题面，不要悲伤，不要心急，你可以继续看。
- $c$ 题中的 $\frac{1}{4}$ 是一个人名（不了解这一点你可能看不懂题意）。
- 如果你不知道湖南最强选手的名字，你可以去问一问 $\frac{1}{4}$ 和Tom。
- Polycarp是某位擅长表演、字符串、组合数学的大佬（请不要对号入座），如果你没有做出来说明这位大佬很厉害。
- Axel是某位擅长工业数据结构和线性基的大佬（请不要对号入座），如果你没有做出来说明这位大佬很厉害。
- Tom和 $\frac{1}{4}$ 是两位热爱博弈的先生（呃~我真的不是故意的），如果你有做出来可以观看现场博弈~。



- 什么？你点开了全文然后看完了？抓紧时间贪心大师在看着你。