

# 2020年CCF全国信息学奥林匹克联赛复赛提高组

## CCF CSP-S 2020

### 模拟赛 day1

时间: 2020 年 9 月 6 日 08:30 ~ 12:30

题目名称	异或帽子	传话游戏	全球覆盖	幂次序列
题目类型	传统型	传统型	传统型	传统型
目录	hat	string	globe	sequence
可执行文件名	hat	string	globe	sequence
输入文件名	hat.in	string.in	globe.in	sequence.in
输出文件名	hat.out	string.out	globe.out	sequence.out
每个测试点时限	1.0 秒	1.0 秒	2.0 秒	4.0 秒
内存限制	512 Mib	512 MiB	512 MiB	512 MiB
子任务数目	20	20	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	hat.cpp	string.cpp	globe.cpp	sequence.cpp
-----------	---------	------------	-----------	--------------

编译选项

对于 C++ 语言	<code>-lm -O2 -std=c++11</code>
-----------	---------------------------------

注意事项与提醒（请选手务必仔细阅读）

1. 提交的程序代码文件的放置位置请参照各省的具体要求。
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
4. C++ 中函数 `main()` 的返回值类型必须是 `int`，值为 0。
5. 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。
6. 程序可使用的栈内存空间限制与题目的内存限制一致。
7. 只提供 windows 格式附加样例文件。
8. 上述时限均以本次评测配置为准。

# 异或帽子 (hat)

## 【题目描述】

小黑的老师给他们班的同学做了一个小游戏。老师给班里 $n$ 个同学（由于小黑班上男女生人数一样，所以 $n$ 一定是偶数）每人戴上了一顶帽子，第 $i$ 个同学的帽子上写有一个数字 $A_i$ 。每个人只能看到其他人帽子上的数字，而看不到自己的。每个同学都把其他所有人的数字的二进制异或和算了出来，并记录了下来，第 $i$ 个同学记下来的数字是 $B_i$ 。根据这些消息，老师让同学们试着算出自己帽子上的数字，小黑很聪明，所以一下子就算出来了。

回家的路上，小黑向小白说了这个游戏，并只把所有的 $B_i$ 告诉了小白，让小白猜出来每个人的数字 $A_i$ 是多少。但是小白不会算，所以她来向你求助，请你告诉她所有 $A_i$ 的值。

## 【输入格式】

从文件 `hat.in` 中读入数据。

第一行一个整数 $n$ ，表示小黑班上有 $n$ 个同学，保证是偶数。

接下来一行 $n$ 个数字，第 $i$ 个数字表示 $B_i$ 。

## 【输出格式】

输出到文件 `hat.out` 中。

一行 $n$ 个数字，第 $i$ 个数字表示你求出来的 $A_i$ 。

## 【样例 1 输入】

```
4
20 11 9 24
```

## 【样例 1 输出】

```
26 5 7 22
```

## 【样例 1 解释】

我们用 $\oplus$ 表示异或运算

$$5 \oplus 7 \oplus 22 = 20$$

$$26 \oplus 7 \oplus 22 = 11$$

$$26 \oplus 5 \oplus 22 = 9$$

$$26 \oplus 5 \oplus 7 = 24$$

### 【样例 2】

见下发文件中的`hat2.in`和`hat2.ans`。

### 【数据范围】

对于前10% 的数据， $n \leq 5$ 。

对于前30% 的数据， $n \leq 5000$ 。

对于另10% 的数据，保证 $A_i, B_i < 32$ 。

对于100% 的数据， $n \leq 200000$ ， $0 \leq A_i, B_i < 2^{30}$ 。

# 传话游戏 (string)

## 【题目描述】

小白和小黑在做一个游戏。小白先在纸条上写下一个由 $n$ 个只包含小写字母的单词组成的句子，让小黑把它记下来，然后小白收回纸条，小黑凭着记忆向小白复述这个句子。

小白发现小黑的记忆力不是很好，经常把单词的顺序记错，但凭借脑海中的印象，小黑复述的句子一定也包含 $n$ 个只包含小写字母的单词，且其组成的可重集合与原来的句子完全相同。同时，对于某个单词 $S$ ，其在原句子和小黑复述的句子中第 $i$ 次出现的位置不会相差超过1。小白很好奇，小黑可能复述出多少种不同的句子，因此找你来帮忙计算。

两个句子被认为是不同的，当且仅当存在某个数 $i$ 满足两个句子的第 $i$ 个单词不同。注意，原句子也应被纳入统计，因为显然小黑可以复述出相同的句子。答案可能很大，请输出答案对1000000007取模的结果。

## 【输入格式】

从文件 `string.in` 中读入数据。

第一行一个整数 $n$ ，表示句子中包含 $n$ 个单词。

接下来一行 $n$ 个字符串，第 $i$ 个字符串表示第 $i$ 个单词。

## 【输出格式】

输出到文件 `string.out` 中。

一行，一个整数，表示答案对1000000007取模的结果。

## 【样例 1 输入】

```
3
it is me
```

## 【样例 1 输出】

```
3
```

## 【样例 1 解释】

小黑复述的3种句子可以是：

```
it is me
is it me
it me is
```

**【样例 2 输入】**

13

yi yi si wu yi si yi jiu yi jiu ba yao ling

**【样例 2 输出】**

233

**【样例 3】**

见下发文件中的string3.in和string3.ans。

**【数据范围】**

对于前20% 的数据， $n \leq 20$ 。

对于前50% 的数据， $n \leq 5000$ 。

对于另10% 的数据，所有单词长度均为1。

对于100% 的数据， $n \leq 100000$ ，所有单词总长不超过2000000。

# 全球覆盖 (globe)

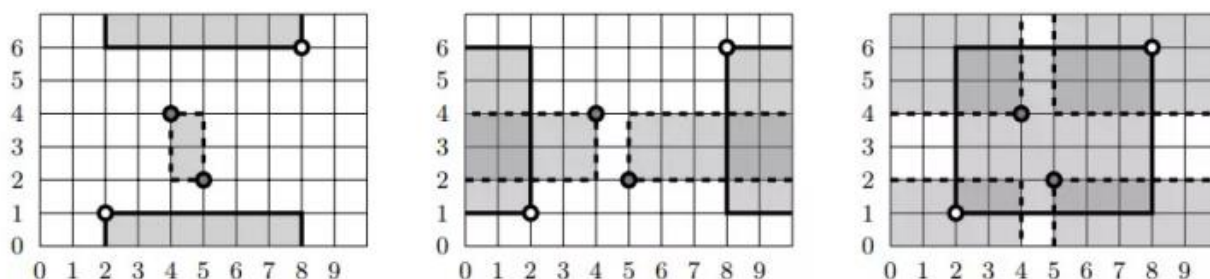
## 【题目描述】

小黑正在研发一款全球定位软件，想用它来定位小白的坐标。具体来说，地球可以看做一个 $X \times Y$ 的网格矩阵，横纵坐标范围分别是 $[0, X)$ 和 $[0, Y)$ ，由于地球是球形结构，网格的上边界和下边界是相通的，左边界和右边界也是相通的。

现在小黑获得了 $n$ 组坐标对，每组坐标对含有两个点的坐标 $(x_{i,0}, y_{i,0}), (x_{i,1}, y_{i,1})$ ，表示地球上一个两边平行于坐标轴的矩形的两个对角顶点，而小白就在这个矩形内部。

然而，由于地球是球形结构，确定了坐标对后仍然有多种可能的“矩形”（如下图所示）。小黑想知道最多可能有多少面积的网格出现在所有“矩形”的交集之中，以方便他确定小白的的位置。每个单元格的面积为1。

于是他把这个问题交给你了。



## 【输入格式】

从文件 `globe.in` 中读入数据。

第一行三个正整数 $n, X, Y$ ，含义如题目描述。

接下来 $n$ 行，每行四个正整数 $x_{i,0}, y_{i,0}, x_{i,1}, y_{i,1}$ ，描述一组坐标对。所有数据始终保证有 $x_{i,0} < x_{i,1}, y_{i,0} < y_{i,1}$ 。

## 【输出格式】

输出到文件 `globe.out` 中。

一行，一个整数，表示所求的答案。

## 【样例 1 输入】

```
2 10 7
2 1 8 6
4 2 5 4
```

## 【样例 1 输出】

```
15
```

## 【样例 1 解释】

样例中的情况和题目中图片一致，其中第三种情况的面积最大。

**【样例 2】**

见下发文件中的globe2.in和globe2.ans。

**【数据范围】**

对于前10% 的数据， $n \leq 10$ 。

对于前20% 的数据， $n \leq 20$ 。

对于另50% 的数据， $n \leq 3000$ 。

对于100% 的数据， $n \leq 500000, 2 \leq X, Y \leq 10^9, 0 \leq x_0, x_1 < X, 0 \leq y_0, y_1 < Y$ 。

# 幂次序列 (sequence)

## 【题目描述】

小黑和小白又在玩游戏。小黑有一个序列，每个元素都形如 $2^x$ ，其中 $x$ 是整数。小白每次可以选择序列里连续的一段，然后计算这段区间内所有元素的总和，记为 $s$ ，也就是将这段区间合并为一个数。为了让游戏更有难度，小黑要求小白合并时必须保证 $s$ 也是 $2^x$ 形式的数。

然而，小白不擅长计算，因此她很难找到一个合法的区间。于是她向你求助，想知道对于给定的初始序列，有多少区间可以保证合并后产生的 $s$ 也是 $2^x$ 形式的数。

注意，如果一个区间只有1个数，也被视为是合法的区间。

## 【输入格式】

从文件 `sequence.in` 中读入数据。

第一行一个正整数 $n$ ，表示序列的长度。

接下来一行 $n$ 个整数 $a_i$ ，表示序列的第 $i$ 个元素为 $2^{a_i}$ 。

## 【输出格式】

输出到文件 `sequence.out` 中。

一行，一个整数，表示所求的答案。

## 【样例 1 输入】

```
3
1 1 2
```

## 【样例 1 输出】

```
5
```

## 【样例 1 解释】

一共有5个合法区间， $[1,1]$ ,  $[2,2]$ ,  $[3,3]$ ,  $[1,2]$ ,  $[1,3]$ 。

## 【样例 2】

见下发文件中的`sequence2.in`和`sequence2.ans`。

## 【样例 3】

见下发文件中的`sequence3.in`和`sequence3.ans`。

## 【数据范围】

对于前10% 的数据， $n \leq 100$ 。

对于前20% 的数据， $n \leq 1000$ 。



对于前50% 的数据,  $n \leq 5000$ 。

对于前80% 的数据,  $n \leq 50000$ 。

前80% 的数据中, 有4个测试点满足 $a_i \leq 2$ , 还有另外4个测试点满足 $a_i \leq 30$ 。

对于100% 的数据,  $n \leq 200000, 1 \leq a_i \leq 10^9$ 。