

# 2024年NOIP模拟赛

时间：2024 年 10 月 30 日 07:30 ~ 12:00

by 广铁一中，not “比”

题目名称	铭记 (remember)	世界(world)	桥梁(bridge)	生长(growing)
题目类型	传统型	传统型	传统型	传统型
目录	remember	world	bridge	growing
可执行文件名	remember	world	bridge	growing
输入文件名	remember.in	world.in	bridge.in	growing.in
输出文件名	remember.out	world.out	bridge.out	growing.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	1024 MiB	512 MiB	1024 MiB
子任务数目	5	7	5	6
是否采用子任务捆绑	是	是	是	是

提交源程序文件名

对于 C++ 语言	remember.cpp	world.cpp	bridge.cpp	growing.cpp
-----------	--------------	-----------	------------	-------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

### 注意事项（请仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- C/C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
- 提交的程序代码文件的放置位置请参考各省的具体要求。
- 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
- 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）。
- 选手提交的程序源文件必须不大于 100KB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。
- 全国统一评测时采用的机器配置为：Inter(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。
- 只提供 Linux 格式附加样例文件。
- 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

# 铭记(remember)

## 题目背景

本来这题题目背景很伤感，但出题人觉得模拟赛还是要欢乐一些的好。

于是在此介绍一个英雄：XQ。

XQ 是这个版本中非常超标的一个英雄，为什么呢？因为他的大招非常的强。

大招名称：听不懂。

大招描述：已读乱回。当 XQ 收到一句话后，会回复一句无关的话。

这么说可能没什么感觉，以下是一个例子：

JT：你过没过 A？

XQ：不是哥们瓦利瓦利？（意思为“我哪里\*了”）

JT：你是不是\*诗。

XQ：总有婴儿想害朕！

JT：不是你个随机已读乱回。

XQ：不处。

JT：不是你会不会说话啊？

XQ：Tringer（婴儿），Tringer，我亲爱的 Tringer。

综上，建议在下个版本削弱 XQ。

**注：此题目背景与现实中的任何人物均无关，如有雷同纯属巧合！**

## 题目描述

给你一个长度为  $n$  的序列  $a$ 。求

$$\sum_{i=1}^n \sum_{j=i+1}^n a_i \oplus a_j$$

当  $\oplus$  为按位与，按位或，按位异或时，分别求出答案。

答案不需要对 **998244853** 取模。

## 输入格式

第一行一个正整数  $n$ 。

第二行共  $n$  个正整数，表示序列  $a$ 。

## 输出格式

共三行，每行一个数。分别表示当  $\oplus$  为按位与，按位或，按位异或时的答案。

## 样例

## 样例1

输入

```
3
3 4 7
```

输出

```
7
21
14
```

解释

⊕ 为按位与时：

$$\begin{aligned}3 \oplus 4 &= 0 \\3 \oplus 7 &= 3 \\4 \oplus 7 &= 4 \\0 + 3 + 4 &= 7\end{aligned}$$

⊕ 为按位或时：

$$\begin{aligned}3 \oplus 4 &= 7 \\3 \oplus 7 &= 7 \\4 \oplus 7 &= 7 \\7 + 7 + 7 &= 21\end{aligned}$$

⊕ 为按位异或时：

$$\begin{aligned}3 \oplus 4 &= 7 \\3 \oplus 7 &= 4 \\4 \oplus 7 &= 3 \\7 + 4 + 3 &= 14\end{aligned}$$

## 样例2

输入

```
10
1 1 0 1 0 1 1 0 0 1
```

输出

```
15
39
24
```

### 样例3

输入

```
2
998244353 998244853
```

输出

```
998244353
998244853
500
```

解释

仔细检查你是不是读错题了???

### 样例4

见附件中的 `remember4.in/ans`。

该组样例满足子任务 3 的要求。

### 样例5

见附件中的 `remember5.in/ans`。

该组样例满足子任务 5 的要求。

### 样例6

见附件中的 `remember6.in/ans`。

该组样例满足子任务 5 的要求。

### 数据范围

本题采用捆绑测试。

子任务编号	特殊性质	分值
1	$n \leq 10^3$	24
2	$a_i \leq 1$	14
3	$a_i \leq 3$	11
4	$n \leq 10^5$	33
5	无特殊限制	18

对于所有测试数据,  $1 \leq n \leq 10^6, 0 \leq a_i \leq 10^9$ 。

# 世界(world)

## 题目背景

首先这个世界不是《我的世界》，然后就没有然后了。

## 题目描述

你现在处于一个二维世界中，这个世界的大小为  $n \times \infty$ 。对于所有满足  $1 \leq i \leq n$  的正整数  $i$ ， $(i, -\infty)$  处至  $(i, a_i)$  处有石头。其余地方没有任何东西。

你一开始站在  $(1, a_1)$  石头的上面，无论何时你只会处在这个世界中某一个整点处的上面。

假设当前你在石头  $(x, y)$  的上面，每次你可以进行以下操作之一：

- 若向左走不会撞到石头（即  $(x - 1, y + 1)$  没有石头），则可以向左移动到  $(x - 1, y)$  处的上面。
- 若向右走不会撞到石头（即  $(x + 1, y + 1)$  没有石头），则可以向右移动到  $(x + 1, y)$  处的上面。
- 挖掉  $(x, y)$  处的石头。
- 若  $(x - 1, y + 1)$  处有石头，则可以挖掉  $(x - 1, y + 1)$  处的石头。
- 若  $(x + 1, y + 1)$  处有石头，则可以挖掉  $(x + 1, y + 1)$  处的石头。

如果  $(x, y)$  处的石头被挖掉，则对于所有满足  $i \geq y$  的整数  $i$ ，若  $(x, i)$  处有石头，则  $(x, i)$  处的石头将会立刻消失。

如果某个时刻你的脚底下没有石头，那么你将会不断下坠直到你的脚底下的位置有石头，假设原来的你在  $(x, y)$  上，停止下坠后你在  $(x, y - z)$  上（显然  $z$  是一个非负整数），则你会受到  $z^2$  点伤害。

移动和挖石头都需要耗费一个时刻的时间，下坠不需要耗费时间。

假设你至多只能承受  $k$  点伤害，对于每一个满足  $1 \leq i \leq n$  正整数  $i$ ，你需要回答至少需要经过多少个时刻，你可以抵达  $(i, 0)$  处的上面。

（提示：你始终处于一个位置的上面，而不是这个位置里。也就是说你所处的位置没有石头，而你的脚下应当始终有石头）

## 输入格式

第一行两个数正整数  $n, k$ ，意义同题意。

第二行  $n$  个数，表示序列  $a$ 。

## 输出格式

共一行  $n$  个数，第  $i$  个数表示抵达  $(i, 0)$  处的上面所需的最少时刻。

## 样例

### 样例 1

输入

```
5 30
11 6 8 5 2
```

输出

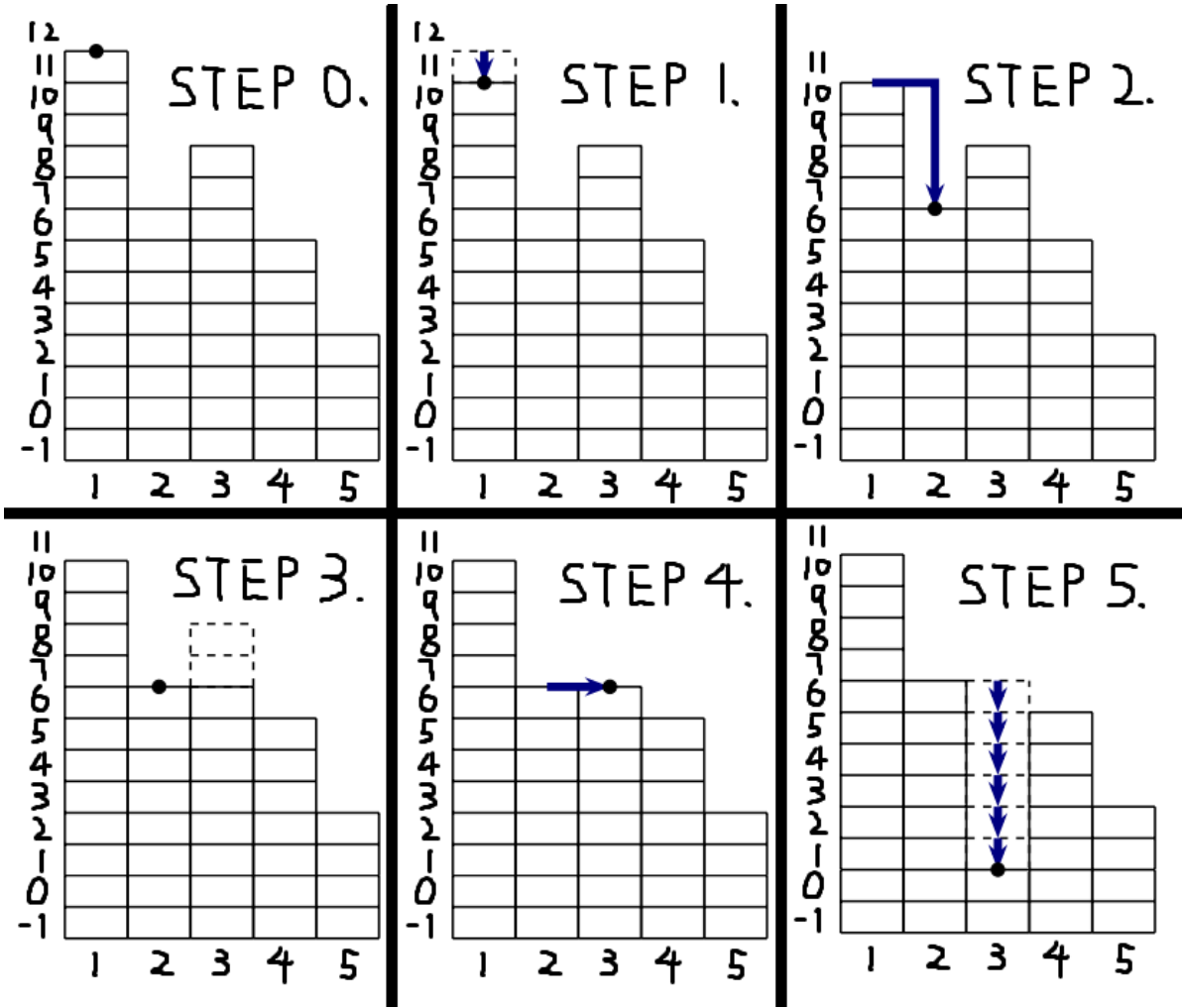
10 8 10 10 8

解释

对于  $i = 3$ , 你的最优行动之一如下:

- 1. 挖掉 (1, 11) 处的石头, 下坠至 (1, 10) 处上并受到 1 点伤害。
- 2. 移动至 (2, 10) 处上, 下坠至 (2, 6) 处上并受到 16 点伤害。
- 3. 挖掉 (3, 7) 处的石头, (3, 8) 处的石头一并消失。
- 4. 移动至 (3, 6) 处上, 此时不会受到伤害。
- 5. 挖掉 (3, 6) 至 (3, 1) 处的石头, 共受到 6 点伤害。

以上所有操作共经过 10 个时刻。



样例 2

输入

5 11  
11 6 8 5 2

输出

11 11 13 13 13

### 样例 3

输入

```
6 100
36 30 24 18 12 6
```

输出

```
30 28 26 24 22 21
```

### 样例 4

输入

```
2 998244853993244853
1000000000 0
```

输出

```
877962 877960
```

### 样例 5

输入

```
5 38
21 50 9 3 50
```

输出

```
21 22 20 19 21
```

### 样例6

见附件中的 `world6.in/ans`。

该样例满足子任务 1 的要求。

### 样例7

见附件中的 `world7.in/ans`。

该样例满足子任务 4 的要求。

### 样例8

见附件中的 `world8.in/ans`。

该样例满足子任务 5 的要求。

## 样例9

见附件中的 `world9.in/ans`。

该样例满足子任务 6 的要求。

## 样例10

见附件中的 `world10.in/ans`。

该样例满足子任务 7 的要求。

## 样例11

见附件中的 `world11.in/ans`。

该样例满足子任务 7 的要求。

## 数据范围

本题采用捆绑测试。

子任务编号	特殊性质	分值
1	$n, a_i \leq 1000$	16
2	$n \leq 1000$	13
3	$a_i \leq 1000$	13
4	$k = a_1$	10
5	A 且只有一个测试点	14
6	$n, a_i \leq 10^5$	20
7	无特殊性质	14

特殊性质 A：保证  $n = 10^6$  且  $k = 10^{12}$ ，对于所有  $a_i$  保证其在  $[0, 10^9]$  内等概率选取一个整数随机生成。

对于所有测试数据， $1 \leq n \leq 10^6, 0 \leq a_i \leq 10^9, a_1 \leq k \leq 10^{18}$ 。



# 桥梁(bridge)

## 题目背景

有人 fridge 和 bridge 分不清，我不说是谁。

原来是我自己🤡。

## 题目描述

请注意本题特殊的数据范围。

潇柵国是一个水上之国，它的每座城市都是一个小岛。

近日，潇柵国国王束鉅打算修筑若干座桥梁连接这  $n$  座城市。潇柵国的桥梁设计师给出了  $m$  个桥梁建造方案，第  $i$  个建造方案可以连接第  $u_i$  座城市与第  $v_i$  座城市，需要花费  $c_i$  的代价。

由于潇柵国的桥梁设计师受到贿赂，对于建造方案的选择将会受到  $q$  条限制，第  $i$  条限制为在第  $a_i$  个与第  $b_i$  个两个建造方案中必须**至少选择一个**建造。

由于潇柵国国王束鉅只会做计数题，所以你需要帮他求出连接所有城市的**最小代价**，如果无论如何都无法连接所有桥梁，输出 mumuyibarenkougumumu。

## 输入格式

第一行两个正整数  $n, m$ ，意义同题目描述。

接下来  $m$  行每行三个正整数  $u_i, v_i, c_i$ ，意义同题目描述。

接下来一行一个整数  $q$ ，意义同题目描述。

接下来  $q$  行每行三个正整数  $a_i, b_i$ ，意义同题目描述。

## 输出格式

共一行一个整数表示最小代价。

如果无论如何都无法连接所有桥梁，输出 mumuyibarenkougumumu。

## 样例

### 样例 1

输入

```
4 6
2 4 4
1 1 4
1 1 2
3 2 4
1 3 4
2 4 3
1
3 6
```

输出

### 解释

选择第 4, 5, 6 个方案进行建造, 满足第 3, 6 个建造方案中必须至少建造一个的条件, 总代价为  $4 + 4 + 3 = 11$ 。

## 样例 2

### 输入

```
5 4
1 2 5
3 4 5
1 3 1
2 3 1
0
```

### 输出

```
mumuyibarenkoumumumu
```

### 解释

可以证明, 无论如何选择建造方案都无法连接所有城市。

## 样例3

见附件中的 `bridge3.in/ans`。

该样例满足子任务 1 的要求。

## 样例4

见附件中的 `bridge4.in/ans`。

该样例满足子任务 3 的要求。

## 样例5

见附件中的 `bridge5.in/ans`。

该样例满足子任务 4 的要求。

## 样例6

见附件中的 `bridge6.in/ans`。

该样例满足子任务 5 的要求。

## 样例7

见附件中的 `bridge7.in/ans`。

该样例满足子任务 5 的要求。

## 数据范围

本题采用捆绑测试。

子任务编号	特殊性质	分值
1	$n, m \leq 100$	10
2	$q \leq 4$	10
3	A	15
4	B	45
5	无特殊性质	20

特殊性质 A：保证对于任意满足  $1 \leq i \leq q$  的正整数  $i$ ，满足  $a_i = b_i$ 。

特殊性质 B：保证对于任意满足  $1 \leq i < j \leq q$  的正整数  $i, j$ ，满足  $a_i \neq a_j$  且  $b_i \neq b_j$  且  $a_i \neq b_j$ 。

对于所有测试数据， $1 \leq n \leq 10^5, 1 \leq m \leq 5 \times 10^5, 1 \leq c_i \leq 10^3, 0 \leq q \leq 16$ 。

# 生长(growing)

## 题目背景

"我们都在一天天长大....."

其实这是一首被用作过起床铃的歌的歌词，但是出题人忘了这首歌是什么了。

## 题目描述

你在潇柵国一座城市的街道上，这条街道是一个很长的直线。

在这条街道上，从左到右依次排布着  $n$  棵树。开始时，第  $i$  棵树的高度为  $h_i$ 。

因为人在挫折中成长，所以树在风暴中成长。有  $m$  场风暴依次降临，每场风暴会**从左向右**或**从右向左依次**席卷这条街道上的每一颗树。当第  $i$  棵树被风暴席卷时，如果前一棵被风暴席卷的树比第  $i$  棵树高，那么第  $i$  棵树会长高 1 个单位的高度。最开始被席卷的树的高度不会有任何变化。

每场风暴会席卷每棵树**恰好**一次，每棵树是否长高取决于其前一棵树被**本场**风暴席卷后的高度。

由于潇柵国国王柬𣎵只会做计数题，所以你需要帮他求出  $m$  场风暴结束后每一棵树的高度。

## 输入格式

第一行两个正整数  $n, m$ ，分别表示树的数量和风暴数量。

第二行共  $n$  个正整数，第  $i$  个数为  $h_i$ ，表示第  $i$  棵树一开始的高度。

第三行一个长度为  $m$  的字符串  $S$ ，第  $i$  个字符为 **A** 或 **B**。**A** 表示这个风暴从左到右席卷街道，**B** 表示这个风暴从右到左席卷街道。

## 输出格式

共一行  $n$  个整数，第  $i$  个数表示第  $i$  棵树最后的高度。

## 样例

### 样例 1

输入

```
6 5
4 3 5 3 6 6
BABAA
```

输出

```
5 5 6 6 6 6
```

解释

对于前三场风暴，树木的高度变化如下：

```
8 10
5 4 12 3 11 6 5 12
BABAABABBB
```

## 输出

```
10 10 12 12 12 12 12 12
```

## 样例6

见附件中的 `growing6.in/ans`。

该样例满足子任务 1 的要求。

## 样例7

见附件中的 `growing7.in/ans`。

该样例满足子任务 2 的要求。

## 样例8

见附件中的 `growing8.in/ans`。

该样例满足子任务 3 的要求。

## 样例9

见附件中的 `growing9.in/ans`。

该样例满足子任务 4 的要求。

## 样例10

见附件中的 `growing10.in/ans`。

该样例满足子任务 5 的要求。

## 样例11

见附件中的 `growing11.in/ans`。

该样例满足子任务 6 的要求。

## 样例12

见附件中的 `growing12.in/ans`。

该样例满足子任务 6 的要求。

## 样例13

见附件中的 `growing13.in/ans`。

该样例满足子任务 6 的要求。

## 数据范围

本题采用捆绑测试。

子任务编号	特殊性质	分值
1	$n, m \leq 10^4$	12
2	A 且 $n, m \leq 10^5$	12
3	B 且 $n, m \leq 10^5$	12
4	C 且只有一个测试点	25
5	$n, m \leq 10^5$	22
6	无特殊性质	17

特殊性质 A：保证满足  $h_i \neq h_{i-1}$  且  $2 \leq i \leq n$  的正整数  $i$  至多有 100 个。

特殊性质 B：保证  $S$  中只有字符 **A**。

特殊性质 C：保证  $n = 10^5$  且  $m = 10^5$ ，对于所有  $h_i$  保证其在  $[1, 10^9]$  内等概率选取一个数随机生成， $S$  中的每个字符在 **A** 和 **B** 中等概率随机选取。

对于所有测试数据， $2 \leq n, m \leq 3 \times 10^5, 1 \leq h_i \leq 10^9$ 。