

提高组水平测试

题目名称	sum	sequence	mod	merge
题目类型	传统型	传统型	传统型	传统型
输入文件名	sum.in	sequence.in	mod.in	merge.in
输出文件名	sum.out	sequence.out	mod.out	merge.out
每个测试点时限	1.0 秒	2.0 秒	2.0 秒	1.0 秒
内存限制	256MB	256MB	256MB	256MB
子任务数目	10	10	10	10
测试点是否等分	是	是	是	是

1. 每道题提交的源程序文件名为该题的题目名称.(cpp/c/pas)，文件名一律小写。
2. 题目难度与顺序无关。
3. 栈空间受题目的内存限制。

最大子段和（sum）

【题目描述】

Geobiyye 是一个喜欢思考问题的女孩子。

Geobiyye 给了你一个序列，她想求出这个序列的最大子段和。

Geobiyye 觉得这个问题太简单了，她将问题扩大了一倍。于是现在问题变成了：从这个序列中选出不相交的两个连续段，要求它们的和最大。

换句话说，对于给定的长度为 n 的序列 a_i ，你需要给出 A, B, C, D ，满足 $1 \leq A \leq B < C \leq D \leq n$ ，并且最大化下列式子：

$$\sum_{i=A}^B a_i + \sum_{i=C}^D a_i$$

现在 Geobiyye 不会这道题了，于是她将问题抛给了你。

【输入格式】

第一行一个正整数 n ，表示序列长度。

接下来一行 n 个整数 $a_1, a_2, a_3, \dots, a_n$ ，表示题目描述中的序列。

【输出格式】

一行一个整数表示最大值。

【输入样例】

```
7
2 -4 3 -1 2 -4 3
```

【输出样例】

```
7
```

【样例说明】

选择的两个区间分别为[3,5]和[7,7]。

【数据范围】

对于 30%的数据： $n \leq 100$ 。

对于 60%的数据： $n \leq 1000$ 。

对于 100%的数据： $n \leq 10^5, |a_i| \leq 10^9$ 。

序列（sequence）

【题目描述】

Geobiyye 是一个喜欢数据结构的女孩子。

Geobiyye 给了你一个长度为 n 的序列 a_i ，序列中每个元素的初始值为 0。

接下来她会对这个序列进行 m 次操作，每次操作有 4 个参数 l, r, s, e ，表示将区间 $[l, r]$ 加上一个首项为 s ，末项为 e 的等差数列。

若一次操作中 $l=1, r=5, s=2, e=10$ ，则对序列中第 1~5 个数分别加上 2,4,6,8,10。

现在 Geobiyye 要求你求出 m 次操作后序列中的每个数的值。

【输入格式】

第一行 2 个整数 n, m ，表示序列长度和操作数。

接下来 m 行，每行 4 个整数 l, r, s, e ，含义见题目描述。

数据保证等差数列中的每一项都是整数。

【输出格式】

由于输出数据过大，Geobiyye 只想要知道最终序列每一项的异或和，即 $a_1 \oplus a_2 \oplus a_3 \oplus \dots \oplus a_n$ 。（其中 \oplus 表示二进制下的异或操作，在 `c++` 中为 `^`）

【输入样例】

```
5 2
1 5 2 10
2 4 1 1
```

【输出样例】

3

【样例说明】

第一次操作加的数列：2 4 6 8 10

第二次操作加的数列：0 1 1 1 0

所有操作结束后序列每个元素值为：2 5 7 9 10。

输出异或和，就是 3。

【数据范围】

对于 30%的数据： $n, m \leq 1000$ 。

对于 50%的数据： $n, m \leq 100000$ 。

对于另外 20%的数据： $s = e$ 。

对于 100%的数据： $n, m \leq 500000, 1 \leq l < r \leq n$ 。

数据保证输入数据以及在任何时候序列中的数在 $[0, 9 \times 10^{18}]$ 范围内。

【HINT】

本题输入文件较大，Geobiiye 给了你一份快速读入的模板。

```
template <typename T> void read(T &x){
    int f=1;x=0;char c=getchar();
    for (;!isdigit(c);c=getchar()) if (c=='-') f=-f;
    for (; isdigit(c);c=getchar()) x=x*10+c-'0';
```

```
    x*=f;
}
```

你可以使用函数 `read(x)` 读入一个 `int` 或 `long long` 类型的整数。

以下为示范程序：

```
#include<bits/stdc++.h>
using namespace std;
template <typename T> void read(T &x){
    int f=1;x=0;char c=getchar();
    for (;!isdigit(c);c=getchar()) if (c=='-') f=-f;
    for (; isdigit(c);c=getchar()) x=x*10+c-'0';
    x*=f;
}
int n;
long long m;
int main(){
    read(n);//读入 int 类型变量 n
    read(m);//读入 long long 类型变量 m
    return 0;
}
```

余数(mod)

【题目描述】

给定正整数 n ，定义一个长度为 n 的排列 a 的价值 w ：若不存在长度为 n 且字典序比 a 大的排列，则 $w=0$ ；否则令 b 为长度为 n 的字典序恰好比 a 大 1 的排列， w 为 a 和 b 不相同的位数的个数。

给定 n 和 p ，求所有长度为 n 的排列的 w 的对 p 取模的结果。

【输入描述】

第一行一个正整数 q ，表示询问组数。

接下来 q 行，每行两个正整数 n, p ，含义如题。

【输出描述】

q 行，每行一个正整数，表示答案。

【输入样例】

```
10
1 530093761
2 892208769
3 659909301
4 348347651
5 703875913
6 870865473
7 61687961
8 334373509
9 826163921
10 735259938
```

【输出样例】

```
0
2
12
```

58

310

1886

13244

106002

954090

9540982

【样例解释】

例如，当 $n=3$ 时

1 2 3 $w=2$

1 3 2 $w=3$

2 1 3 $w=2$

2 3 1 $w=3$

3 1 2 $w=2$

3 2 1 $w=0$

w 的和为 12

【数据范围】

30%的数据： $n \leq 10$

60%的数据： $n \leq 3000$

100%的数据： $1 \leq q \leq 10, 1 \leq n \leq 10^7, 1 \leq p \leq 10^9$

【友情提醒】

p 不一定是质数。

合并果子(merge)

【题目背景】

柠檬树上柠檬果，柠檬树下我和我

【题目描述】

Re-see 特别喜欢柠檬。

Re-see 一共采了 n 个柠檬。一开始每个柠檬自成一堆。之后她又做了 Q 次操作

1 x y:Re-see 觉得不够酸爽，决定把第 x 个柠檬和第 y 个柠檬所在的柠檬堆合并

特别的，如果 x,y 本来就在一堆里，那么什么也不做

2 a b:Re-see 酸了，对第 a 个柠檬所在的柠檬堆中每个柠檬挤了 b 毫升柠檬汁喝

Re-see 操作完后决定吃柠檬，请你回答此时每个柠檬被挤了多少毫升柠檬汁

【输入格式】

第一行 2 个正整数 n,Q

接下来 Q 行表示操作

【输出格式】

输出 1 行表示每个柠檬被挤了多少毫升柠檬汁，空格隔开

【样例输入】

2 3

2 1 1

1 2 1

2 1 3

【样例输出】

4 3

【样例解释】

第一次操作后柠檬堆为[1][2],答案为[1][0]

第二次操作后柠檬堆为[1,2],答案为[1][0]

第三次操作后柠檬堆为[1,2],答案为[4][3]

【数据范围】

保证 $1 \leq x, y, a \leq n; 1 \leq b \leq 100$

请使用较快的输入输出方式

编号	N<=	Q<=	分数
1	10	50	5
2	100	100	10
3	1000	1000	15
4	6000	6000	10
5	6000	500000	35
6	50000 0	500000	25