

CSP 2022 模拟赛

(请选手务必仔细阅读本页内容)

一. 题目概况

	矩阵(啊)	GCD(波)	三维世界(吡)	函数(の)
英文题目名称	a	b	c	d
输入文件名	a.in	b.in	c.in	d.in
输出文件名	a.out	b.out	c.out	d.out
每个测试点时限	1.0s	0.5s	1.0s	1.0s
测试点数目	子任务	子任务	子任务	子任务
每个测试点分值	/	/	/	/
结果比较方式	文本比较	文本比较	文本比较	文本比较
题目类型	传统	传统	传统	传统
运行内存上限	1024 MB	1024 MB	1024 MB	1024 MB
提交源程序文件名	a.cpp	b.cpp	c.cpp	d.cpp

二. 注意事项

- 文件名（程序名和输入输出文件名）必须使用英文小写，不需要建子文件夹。
- 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）。
- 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。
- 编译附加命令为 `-lm -O2 -std=c++14 -Wl,-stack=2147483647`
- 选手提交的程序源文件必须不大于 100KB。
- 禁止在源代码中改变编译器参数（如使用 `#pragma` 命令），禁止使用系统结构相关指令（如内联汇编）和其他可能造成不公平的方法。
- 题面不一定包含全部样例，请查看下发附加文件以得到所有样例。
- 评测在 windows 下进行。但你仍然可以使用诸如 `%lld` 等的标识符用来表达不同的变量类型。
- 评测机配置为 Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz, 16GB。已确认时限均在 std 两倍运行时间的两倍以上。
- 评测使用 Lemon，禁止调用危险的系统函数，禁止访问除输入文件以外的文件，禁止修改除输出文件以外的文件。
- 温馨提示：你可以使用 `size` 命令（与 `g++` 在同一目录下）来测试程序所使用的静态内存。

啊(a)

【问题描述】

给定一个 $n \times m$ 的 01 矩阵，求包含 $[l, r]$ 个 1 的子矩形个数。

【输入格式】

从文件 a.in 中读入数据。

第一行读入两个正整数 n, m 。

接下来 n 行，每行一个长度为 m 的 01 串，表示给定的矩阵。

接下来一行，两个自然数 l, r 。

【输出格式】

输出到文件 a.out 中。

一行一个整数表示答案。

【样例输入输出】

Input	Output
2 3 100 011 2 3	5

样例2 请见下发文件。

【数据范围】

对于所有数据，有 $0 \leq l \leq r \leq n \times m, n \leq 30, m \leq 5 \cdot 10^4$,

子任务编号	$n \leq$	$m \leq$	特殊性质	分值
1	20	20	无	10
2	30	200	无	20
3	1	$5 \cdot 10^4$	无	5
4	5	$5 \cdot 10^4$	无	20
5	30	$5 \cdot 10^4$	矩阵全 0	2
6	30	$5 \cdot 10^4$	矩阵全 1	8
7	30	$5 \cdot 10^4$	$l = 0, r = nm$	10
8	30	$5 \cdot 10^4$	无	25

波(b)

【问题描述】

给定 n 个正整数序列 a_1, a_2, \dots, a_n 每个序列长度为 m 。
选择至少 1 个序列，然后在每个被选择的序列中选择恰好一个元素，求出所有被选择的元素的 gcd。
求所有方案的结果之和，答案对 $10^9 + 7$ 取模。两种方案不同，当且仅当存在至少一个元素，在一种方案中被选择，在另一种中没有。

【输入格式】

从文件 b.in 中读入数据。
第一行两个整数 n, m 。
接下来 n 行，每行 m 个正整数，第 i 行代表第 i 个序列 a_i 。

【输出格式】

输出到文件 b.out 中。
输出一个整数表示答案。

【样例输入输出】

Input	Output
2 3 5 15 8 3 12 10	78

样例2 请见下发文件。

【数据范围】

对于所有数据，有 $0 \leq n \leq 20, 0 \leq m \leq 10^5, 1 \leq a_{i,j} \leq 10^6$ 。

子任务编号	$n \leq$	$m \leq$	$a_{i,j} \leq$	特殊性质	分值
1	10	5	10^5	无	18
2	7	20	10^5	$a_{i,j}$ 在 $[1, 10^5]$ 中随机生成	26
3	1	10^5	10^5	无	1
4	20	10^3	10^5	无	15
5	20	10^5	2	无	10
6	20	10^5	10^5	$a_{i,j}$ 全部相同	10
7	20	10^5	10^5	无	10
8	20	10^5	10^6	无	10

吡(c)

【问题描述】

发起进攻。

在二维母世界诞生的你，第一次见到了三维世界的恢弘壮阔。可惜你无暇过多欣赏，你接收到的命令是：斩草除根！

三维世界可以看成是一个 $A \times B \times C$ 的立方体，由于你的文明还没有完全掌握三维世界的结构，现在你还只有三种不够成熟的进攻路线：

- 1 a b: 将所有 $x \leq a, y \leq b$ 的世界毁灭。
- 2 a b: 将所有 $y \leq a, z \leq b$ 的世界毁灭。
- 3 a b: 将所有 $z \leq a, x \leq b$ 的世界毁灭。

最终你计划了 n 次进攻，你想知道你摧毁了这个世界多少的体积。

【输入格式】

从文件c.in中读入数据。

第一行四个数 n, A, B, C 。接下来 n 行，每行三个数 $type, a, b$ ，表示一次进攻，其中 $type$ 表示进攻路线， a, b 为进攻的参数。

【输出格式】

输出到文件c.out中。

一个数表示答案。

【样例输入输出】

Input	Output
2 4 4 4 2 2 3 3 2 2	32

通过容斥原理可以计算得出， $24 + 16 - 8 = 32$

【数据范围】

对于所有数据，有：

$$n \leq 3 \times 10^5, 1 \leq A, B, C \leq 10^6, u \leq A, v \leq B, w \leq C$$

输入均为正整数。

子任务编号	特殊性质	分值
1	$n, A, B, C \leq 50$	20
2	$C = 1$	10
3	$type = 1$	10
4	$type \leq 2$	10
5	性质1	30
6	无	20

- 性质1：n 是 3 的倍数，且每三次切割可以分为一组，每组切割都遵循如下形式：

$$\begin{array}{l} 1 \ u \ v \\ 2 \ v \ w \\ 3 \ w \ u \end{array}$$

其中，相同字母代表相同的数。

の(d)

【问题描述】

考虑将Fenwick Tree 扩展到 k 进制意义下, C++代码如下:

```
1 void add(int x, int y) {
2     while (x <= n) {
3         s[x] ^= y;
4         x += lowbit(x);
5     }
6 }
7 int sum(int x) {
8     int res = 0;
9     while (x > 0) {
10        res ^= s[x];
11        x -= lowbit(x);
12    }
13    return res;
14 }
```

其中, $\text{lowbit}(x)$ 的定义为 x 在 k 进制下最低非 0 位的值。比如 $k = 5, \text{lowbit}(120) = 20$ 。

但是上述代码并不能正确地维护前缀异或和, 甚至会出现TLE。不过并不会出现死循环, 现请你快速维护上面的两个函数。

【输入格式】

从文件d.in中读入数据。

输入数据第一行包含三个正整数 n, q, k , 分别表示序列长度, 操作次数和进制。

接下来 q 行每行描述一个操作, 具体如下:

1 $x\ y$, 表示执行 $\text{add}(x, y)$ 。

2 $x\ y$, 表示输出 $\text{sum}(x)$ 。

【输出格式】

输出到文件d.out中。

对于每个 2 操作, 输出一行表示询问的结果。

【样例输入输出】

Input	Output
10 10 3	
1 4 10	
2 7	10
1 2 5	15
2 10	0
2 6	11
1 3 7	0
1 7 3	12
2 8	
2 9	
2 10	

样例2,3 请见下发文件。

【数据范围】

对于所有数据，有：

$$1 \leq x \leq n \leq 10^9, 1 \leq q, k \leq 2 \times 10^5, 0 \leq y \leq 10^9$$

子任务编号	$n \leq$	$q \leq$	k	分值
1	3000	3000	无	10
2	2×10^5	2×10^5	$= 2$	10
3	2×10^5	2×10^5	无	10
4	2×10^5	2×10^5	是奇数	10
5	10^9	2×10^5	是奇数	10
6	10^9	2×10^5	是奇数	10
7	10^9	2×10^5	$= 2$	10
8	10^9	2×10^5	无	10
9	10^9	2×10^5	无	10
10	10^9	2×10^5	无	10