

# 五校联考

## GZEE PION 3202 赛拟模 第二场

题目名称	点三连通分量计数	特征多项式加速矩阵快速幂	斯特恩布洛科树上倍增加速二分	单点修改矩形顺序对
题目类型	传统型	传统型	传统型	传统型
目录	three	prefix	frac	matrix
可执行文件名	three	prefix	frac	matrix
输入文件名	three.in	prefix.in	frac.in	matrix.in
输出文件名	three.out	prefix.out	frac.out	matrix.out
每个测试点时限	2000 ms	1000 ms	4000 ms	18000 ms
内存限制	512 MiB	512 MiB	1024 MiB	1024 MiB
子任务数目	6	4	10	4

提交源程序文件名

对于 C++ 语言	three.cpp	prefix.cpp	frac.cpp	matrix.cpp
-----------	-----------	------------	----------	------------

编译选项

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

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. 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。
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. 选手提交的程序源文件必须不大于 100KB。
5. 程序可使用的栈空间内存限制与题目的内存限制一致。
6. 提交方式、时空限制以及是否使用文件输入输出请以各校实际评测环境为准。
7. 请相信自己的做法，并且注意常数优化。
8. 对于确定无法通过的测试点，可以特判以减少评测机压力。

#1

传统

2000  
ms512  
MiB

## 点三连通分量计数 (three)

### 【题目描述】

给你一张  $n$  个点,  $m$  条边的无向图与一个整数  $d$ , 求有多少个大小为 3 的连通块使得其中点的编号的极差小于等于  $d$ 。

两个连通块不同当且仅当一个连通块内有某个点  $x$  而另一个连通块里没有。

### 【输入格式】

第一行三个整数  $n, m, d$ 。

接下来  $m$  行每行两个整数  $u, v$ , 表示图中有一条边  $(u, v)$ 。

### 【输出格式】

一行一个整数表示答案。

### 【样例数据】

#### 【样例 1 输入】

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

#### 【样例 1 输出】

```
4
```

#### 【样例 2 输入】

```
6 6 3
1 4
4 3
3 2
2 5
1 2
2 6
```

## 【样例 2 输出】

5

## 【样例 3】

见下发文件 `ex_three3.in` 和 `ex_three3.out`, 该测试用例满足子任务 1 的约束条件。

## 【样例 4】

见下发文件 `ex_three4.in` 和 `ex_three4.out`, 该测试用例满足子任务 3 的约束条件。

## 【样例 5】

见下发文件 `ex_three5.in` 和 `ex_three5.out`, 该测试用例满足子任务 5 的约束条件。

## 【样例 6】

见下发文件 `ex_three6.in` 和 `ex_three6.out`。

## 【数据规模与约定】

全部数据均保证  $3 \leq n, m \leq 5 \times 10^5$ ,  $2 \leq d < n$ , 保证图中无重边自环。

本题采用子任务捆绑测试, 各个子任务详细信息如下:

编号	子任务分数	$n, m \leq$	特殊性质
1	10	500	
2	10	$10^4$	
3	10	$5 \times 10^5$	图为菊花图
4	20	$5 \times 10^5$	$d = n - 1$
5	20	$5 \times 10^5$	图为二分图
6	30	$5 \times 10^5$	

#2

传统

1000  
ms512  
MiB

## 特征多项式加速矩阵快速幂 (prefix)

### 【题目描述】

给出素数  $p$ 。对于序列  $a$ ，定义其权值为最小的  $k > 0$  使得  $a$  在模  $p$  意义下的  $k$  阶前缀和为  $a$ 。特别地，若不存在这样的  $k$ ，则  $a$  的权值为 0。

求长度为  $n$ ，每个元素在  $[0, p)$  中的整数序列的权值的期望，对 998244353 取模。

### 【输入格式】

输入一行两个整数  $n, p$ ，以空格隔开。

### 【输出格式】

输出一行一个整数，表示所有整数序列的权值的期望对 998244353 取模后的结果。

### 【样例数据】

#### 【样例 1 输入】

```
2 2
```

#### 【样例 1 输出】

```
499122178
```

#### 【样例 1 解释】

序列  $\{0, 0\}$  的权值是 1，序列  $\{0, 1\}$  的权值是 1，序列  $\{1, 0\}$  的权值是 2，序列  $\{1, 1\}$  的权值是 2。

#### 【样例 2 输入】

```
4 5
```

#### 【样例 2 输出】

```
543044933
```

## 【样例 3 输入】

1000 19001

## 【样例 3 输出】

724295988

## 【样例 4 输入】

987831351 843547567

## 【样例 4 输出】

861261240

## 【数据规模与约定】

全部数据均保证  $1 \leq n, p < 998244353$ ，保证  $p$  为质数。

本题采用子任务捆绑测试，各个子任务详细信息如下：

编号	子任务分数	$n \leq$	$p \leq$
1	20	5	11
2	25	998244352	2
3	25	$10^5$	998244352
4	30	998244352	998244352

#3

传统

4000  
ms1024  
MiB

## 斯特恩布洛科树上倍增加速二分 (frac)

## 【题目描述】

给你两个序列  $a, b$ ，每次询问给出  $l, r, x$ ，求

$$\sum_{i=l}^r \frac{a_i}{b_i + x}$$

## 【输入格式】

第一行两个整数  $n, q$  分别表示序列长度、询问次数。第二行  $n$  个整数表示  $a$ 。  
第三行  $n$  个整数表示  $b$ 。

接下来  $q$  行，每行三个整数  $l, r, x$  表示询问。

## 【输出格式】

输出  $q$  行，每行一个实数表示答案。你的答案只需与标准答案相差不超过  $10^{-6}$  即可。

## 【样例数据】

## 【样例 1 输入】

```
5 5
3 4 1 2 8
10 9 2 2 9
3 5 1
3 4 6
2 2 8
4 4 1
4 4 3
```

## 【样例 1 输出】

```
1.800000000000
0.375000000000
0.235294117647
0.666666666667
0.400000000000
```

**【样例 2】**

见下发文件 `ex_frac2.in` 和 `ex_frac2.out`, 该测试用例满足测试点 1 ~ 2 的约束条件。

**【样例 3】**

见下发文件 `ex_frac3.in` 和 `ex_frac3.out`, 该测试用例满足测试点 3 ~ 4 的约束条件。

**【样例 4】**

见下发文件 `ex_frac4.in` 和 `ex_frac4.out`。

**【数据规模与约定】**

全部数据均保证  $1 \leq n \leq 3 \times 10^5$ ,  $1 \leq q \leq 10^6$ ,  $1 \leq x \leq 10^6$ ,  $1 \leq l \leq r \leq n$ ,  $1 \leq a_i \leq b_i \leq 10^6$ 。

本题共有 10 个测试点, 每个测试点 10 分, 各个测试点详细信息如下:

测试点编号	$n \leq$	$q \leq$
1 ~ 2	1000	1000
3 ~ 4	$10^5$	$10^5$
5 ~ 10	$3 \times 10^5$	$10^6$

**【提示】**

可以使用下发文件中的 `checker.cpp` 以检验你的输出的正确性, 使用时先将其编译为可执行文件 `checker` (在 Windows 系统下为编译为 `checker.exe`)。

- Linux 系统使用 `./checker <input-file> <output-file> <answer-file>` 测试;
- Windows 系统使用 `checker <input-file> <output-file> <answer-file>` 测试。

其中 `<input-file>` `<output-file>` `<answer-file>` 分别是输入文件、选手输出和标准答案。

#4

传统

18000  
ms1024  
MiB

## 单点修改矩形顺序对 (matrix)

「看着眼下逐渐铺开的纯白世界」  
「闪耀着光的地方是积雪皑皑的山巅」  
「那座山叫乞力马扎罗山」  
「男人心想」  
「自己该去的地方就是那里了」

## 【题目描述】

请注意本题特殊的时间限制。

给你平面上  $n$  个点，第  $i$  点初始时有点权  $v_i$ 。现在一共有  $m$  次操作，每次操作给出一个矩形，把被包含矩形在中的点的点权减去  $c$ 。

每次操作后，请你求出点权大于等于 0 的点的个数。本题强制在线，即你只有在算出前一次操作后的答案后，才能得到下一次操作的信息。

## 【输入格式】

第一行两个整数  $n, m$ 。

下面  $n$  行，第  $i+1$  行是两个整数  $p_i, v_i$  表示点  $(i, p_i)$  的点权是  $v_i$ 。  $p_i$  构成  $1 \dots n$  的排列。

下面  $m$  行，每行五个整数  $x'_1, y'_1, x'_2, y'_2, c'$ 。为了体现你的程序的在线性，操作进行了加密：记  $lastans$  为上次询问的答案，初始为 0，将这五个数分别异或  $lastans$  后得到  $x_1, y_1, x_2, y_2, c$  表示矩形的左下角  $(x_1, y_1)$ 、右上角  $(x_2, y_2)$  和减数  $c$ 。

## 【输出格式】

输出  $m$  行，每行一个整数表示第  $i$  次操作后的答案。



**【样例数据】****【样例 1 输入】**

```
5 5
5 3
2 3
4 13
3 4
1 10
4 1 5 1 3
4 6 0 1 3
6 7 6 7 0
6 7 0 0 14
1 0 7 0 4
```

**【样例 1 输出】**

```
5
4
4
3
3
```

**【样例 1 解释】**

样例 1 的输入解密后为：

```
5 5
5 3
2 3
4 13
3 4
1 10
4 1 5 1 3
1 3 5 4 6
2 3 2 3 4
2 3 4 4 10
2 3 4 3 7
```

**【样例 2】**

见下发文件 ex\_matrix2.in 和 ex\_matrix2.out, 该测试用例满足子任务 1 的约束条件。

**【样例 3】**

见下发文件 ex\_matrix3.in 和 ex\_matrix3.out, 该测试用例满足子任务 2 的约束条件。

**【样例 4】**

见下发文件 `ex_matrix4.in` 和 `ex_matrix4.out`。

**【数据规模与约定】**

全部数据均保证  $1 \leq n \leq 10^4$ ,  $1 \leq m \leq 4 \times 10^6$ ,  $0 \leq v_i, c \leq 4 \times 10^7$ ,  
 $1 \leq x_1 \leq x_2 \leq n$ ,  $1 \leq y_1 \leq y_2 \leq n$ 。

本题采用子任务捆绑测试，各个子任务详细信息如下：

编号	子任务分数	$n \leq$	$m \leq$
1	20	1000	1000
2	30	$10^4$	$1.5 \times 10^6$
3	30	$10^4$	$3 \times 10^6$
4	20	$10^4$	$4 \times 10^6$

**【提示】**

为了防止因为读入规模过大而造成卡常，我们下发了基于 `fread` 的快读模板，  
请见下发文件 `FastRead.cpp`。

使用此模板时请保证您已经引用头文件 `bits/stdc++.h`，且应保证不使用其他读入方式。如使用文件输入，请在首次调用函数 `read()` 前将 `stdin` 重打开为输入文件。  
调用 `read()` 函数即可获取输入中下一个未被读入的 64 位有符号整数。