2019 年 CCF 非专业级软件能力认证提高级第二轮

2019 CCF CSP-S2

day2

时间： 2019 年 11 月 17 日 08:30 ~ 12:00

题目名称	Emiya 家今天的饭	划分	树的重心
题目类型	传统型	传统型	传统型
目录	meal	partition	centroid
可执行文件名	meal	partition	centroid
输入文件名	meal.in	partition.in	centroid.in
输出文件名	meal.out	partition.out	centroid.out
每个测试点时限	1.0 秒	2.0 秒	3.0 秒
内存限制	256 MiB	1024 MiB	256 MiB
子任务数目	25	25	20
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	meal.cpp	partition.cpp	centroid.cpp
对于 C 语言	meal.c	partition.c	centroid.c
对于 Pascal 语言	meal.pas	partition.pas	centroid.pas

编译选项

对于 C++ 语言	-lm
对于 C 语言	-lm
对于 Pascal 语言	

注意事项与提醒（请选手务必仔细阅读）

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. 提交的程序代码文件的放置位置请参照各省的具体要求。
4. 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。
5. 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）。

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. 全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，内存 32GB。上述时限以此配置为准。
8. 只提供 Linux 格式附加样例文件。
9. 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其为准。
10. 程序编译时编译命令中不包含任何优化开关。
11. $\sum$ 是求和运算符，$\sum\limits_{i=1}^{n} a_i$ 的值等于 $a_1 + a_2 + \cdots + a_n$。

Emiya 家今天的饭（meal）

【题目描述】

Emiya 是个擅长做菜的高中生，他共掌握 n 种**烹饪方法**，且会使用 m 种**主要食材**做菜。为了方便叙述，我们对烹饪方法从 $1 \sim n$ 编号，对主要食材从 $1 \sim m$ 编号。

Emiya 做的每道菜都将使用**恰好一种**烹饪方法与**恰好一种**主要食材。更具体地，Emiya 会做 $a_{i,j}$ 道不同的使用烹饪方法 i 和主要食材 j 的菜（$1 \le i \le n$，$1 \le j \le m$），这也意味着 Emiya 总共会做 $\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m} a_{i,j}$ 道不同的菜。

Emiya 今天要准备一桌饭招待 Yazid 和 Rin 这对好朋友，然而三个人对菜的搭配有不同的要求，更具体地，对于一种包含 k 道菜的搭配方案而言：

- Emiya 不会让大家饿肚子，所以将做**至少一道菜**，即 $k \ge 1$
- Rin 希望品尝不同烹饪方法做出的菜，因此她要求每道菜的**烹饪方法互不相同**
- Yazid 不希望品尝太多同一食材做出的菜，因此他要求每种**主要食材**至多在**一半**的菜（即 $\lfloor \frac{k}{2} \rfloor$ 道菜）中被使用
  - 这里的 $\lfloor x \rfloor$ 为下取整函数，表示不超过 x 的最大整数

这些要求难不倒 Emiya，但他想知道共有多少种不同的符合要求的搭配方案。两种方案不同，当且仅当存在至少一道菜在一种方案中出现，而不在另一种方案中出现。

Emiya 找到了你，请你帮他计算，你只需要告诉他符合所有要求的搭配方案数对质数 $998,244,353$ 取模的结果。

【输入格式】

从文件 *meal.in* 中读入数据。

第 1 行两个用单个空格隔开的整数 n, m。

第 2 行至第 $n+1$ 行，每行 m 个用单个空格隔开的整数，其中第 $i+1$ 行的 m 个数依次为 $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$。

【输出格式】

输出到文件 *meal.out* 中。

仅一行一个整数，表示所求方案数对 $998,244,353$ 取模的结果。

【样例 1 输入】

```
2 3
1 0 1
0 1 1
```

【样例 1 输出】

```
3
```

【样例 1 解释】

由于在这个样例中，对于每组 i, j，Emiya 都最多只会做一道菜，因此我们直接通过给出烹饪方法、主要食材的编号来描述一道菜。

符合要求的方案包括：

- 做一道用烹饪方法 1、主要食材 1 的菜和一道用烹饪方法 2、主要食材 2 的菜
- 做一道用烹饪方法 1、主要食材 1 的菜和一道用烹饪方法 2、主要食材 3 的菜
- 做一道用烹饪方法 1、主要食材 3 的菜和一道用烹饪方法 2、主要食材 2 的菜

因此输出结果为 $3 \bmod 998,244,353 = 3$。

需要注意的是，所有只包含一道菜的方案都是不符合要求的，因为唯一的主要食材在超过一半的菜中出现，这不满足 Yazid 的要求。

【样例 2 输入】

```
3 3
1 2 3
4 5 0
6 0 0
```

【样例 2 输出】

```
190
```

【样例 2 解释】

Emiya 必须至少做 2 道菜。

做 2 道菜的符合要求的方案数为 100。

做 3 道菜的符合要求的方案数为 90。

因此符合要求的方案数为 $100 + 90 = 190$。

【样例 3 输入】

```
5 5
1 0 0 1 1
0 1 0 1 0
1 1 1 1 0
```

```
1 0 1 0 1
0 1 1 0 1
```

【样例 3 输出】

```
742
```

【样例 4】

见选手目录下的 ***meal/meal4.in*** 与 ***meal/meal4.ans***。

【样例 5】

见选手目录下的 ***meal/meal5.in*** 与 ***meal/meal5.ans***。

【数据范围】

<table>
<tr><th>测试点编号</th><th>$n =$</th><th>$m =$</th><th>$a_{i,j} <$</th></tr>
<tr><td>1</td><td rowspan="2">2</td><td>2</td><td rowspan="6">2</td></tr>
<tr><td>2</td><td>3</td></tr>
<tr><td>3</td><td rowspan="2">5</td><td>2</td></tr>
<tr><td>4</td><td>3</td></tr>
<tr><td>5</td><td rowspan="2">10</td><td>2</td></tr>
<tr><td>6</td><td>3</td></tr>
<tr><td>7</td><td rowspan="2">10</td><td>2</td><td rowspan="5">1000</td></tr>
<tr><td>8</td><td>3</td></tr>
<tr><td>9~12</td><td rowspan="3">40</td><td>2</td></tr>
<tr><td>13~16</td><td>3</td></tr>
<tr><td>17~21</td><td>500</td></tr>
<tr><td>22~25</td><td>100</td><td>2000</td><td>$998,244,353$</td></tr>
</table>

对于所有测试点，保证 $1 \le n \le 100$，$1 \le m \le 2000$，$0 \le a_{i,j} < 998,244,353$。

划分（partition）

【题目描述】

2048 年，第三十届 CSP 认证的考场上，作为选手的小明打开了第一题。这个题的样例有 n 组数据，数据从 $1 \sim n$ 编号，i 号数据的规模为 a_i。

小明对该题设计出了一个暴力程序，对于一组规模为 u 的数据，该程序的**运行时间**为 u^2。然而这个程序运行完一组规模为 u 的数据之后，它将在任何一组规模**小于** u 的数据上运行错误。样例中的 a_i 不一定递增，但小明又想在不修改程序的情况下正确运行样例，于是小明决定使用一种非常原始的解决方案：将所有数据划分成若干个数据段，段内数据编号**连续**，接着将同一段内的数据合并成新数据，其规模等于段内原数据的**规模之和**，小明将让新数据的规模能够递增。

也就是说，小明需要找到一些分界点 $1 \le k_1 < k_2 < \cdots < k_p < n$，使得

$$\sum_{i=1}^{k_1} a_i \le \sum_{i=k_1+1}^{k_2} a_i \le \cdots \le \sum_{i=k_p+1}^{n} a_i$$

注意 p 可以为 0 且此时 $k_0 = 0$，也就是小明可以将所有数据合并在一起运行。

小明希望他的程序在正确运行样例情况下，运行时间也能尽量小，也就是**最小化**

$$(\sum_{i=1}^{k_1} a_i)^2 + (\sum_{i=k_1+1}^{k_2} a_i)^2 + \cdots + (\sum_{i=k_p+1}^{n} a_i)^2$$

小明觉得这个问题非常有趣，并向你请教：给定 n 和 a_i，请你求出最优划分方案下，小明的程序的最小运行时间。

【输入格式】

从文件 *partition.in* 中读入数据。

由于本题的数据范围较大，部分测试点的 a_i 将在程序内生成。

第一行两个整数 $n, type$。n 的意义见题目描述，$type$ 表示输入方式。

1. 若 $type = 0$，则该测试点的 a_i **直接给出**。输入文件接下来：第二行 n 个以空格分隔的整数 a_i，表示每组数据的规模。
2. 若 $type = 1$，则该测试点的 a_i 将**特殊生成**，生成方式见后文。输入文件接下来：第二行六个以空格分隔的整数 x, y, z, b_1, b_2, m。接下来 m 行中，第 i（$1 \le i \le m$）行包含三个以空格分隔的正整数 p_i, l_i, r_i。

对于 $type = 1$ 的 $23 \sim 25$ 号测试点，a_i 的生成方式如下：

给定整数 x, y, z, b_1, b_2, m，以及 m 个三元组 (p_i, l_i, r_i)。

保证 $n \ge 2$。若 $n > 2$，则 $\forall 3 \le i \le n$，$b_i = (x \times b_{i-1} + y \times b_{i-2} + z) \mod 2^{30}$。

保证 $1 \le p_i \le n$，$p_m = n$。令 $p_0 = 0$，则 p_i 还满足 $\forall 0 \le i < m$ 有 $p_i < p_{i+1}$。

对于所有 $1 \le j \le m$，若下标值 i（$1 \le i \le n$）满足 $p_{j-1} < i \le p_j$，则有

$$a_i = \left(b_i \mod (r_j - l_j + 1)\right) + l_j$$

上述数据生成方式仅是为了减少输入量大小，标准算法不依赖于该生成方式。

【输出格式】

输出到文件 ***partition.out*** 中。

输出一行一个整数，表示答案。

【样例 1 输入】

```
5 0
5 1 7 9 9
```

【样例 1 输出】

```
247
```

【样例 1 解释】

最优的划分方案为 $\{5,1\},\{7\},\{9\},\{9\}$。由 $5+1 \le 7 \le 9 \le 9$ 知该方案合法。

答案为 $(5+1)^2+7^2+9^2+9^2=247$。

虽然划分方案 $\{5\},\{1\},\{7\},\{9\},\{9\}$ 对应的运行时间比 247 小，但它不是一组合法方案，因为 $5>1$。

虽然划分方案 $\{5\},\{1,7\},\{9\},\{9\}$ 合法，但该方案对应的运行时间为 251，比 247 大。

【样例 2 输入】

```
10 0
5 6 7 7 4 6 2 13 19 9
```

【样例 2 输出】

```
1256
```

【样例 2 解释】

最优的划分方案为 $\{5\},\{6\},\{7\},\{7\},\{4,6,2\},\{13\},\{19,9\}$。

【样例 3 输入】

```
10000000 1
123 456 789 12345 6789 3
2000000 123456789 987654321
7000000 234567891 876543219
10000000 456789123 567891234
```

【样例 3 输出】

```
4972194419293431240859891640
```

【样例 4】

见选手目录下的 ***partition/partition4.in*** 与 ***partition/partition4.ans***。

【样例 5】

见选手目录下的 ***partition/partition5.in*** 与 ***partition/partition5.ans***。

【数据范围】

测试点编号	$n \le$	$a_i \le$	$type =$
$1 \sim 3$	10	10	
$4 \sim 6$	50	10^3	
$7 \sim 9$	400	10^4	0
$10 \sim 16$	5000	10^5	
$17 \sim 22$	5×10^5	10^6	
$23 \sim 25$	4×10^7	10^9	1

所有测试点满足：$type \in \{0,1\}$，$2 \le n \le 4 \times 10^7$，$1 \le a_i \le 10^9$，$1 \le m \le 10^5$，$1 \le l_i \le r_i \le 10^9$，$0 \le x,y,z,b_0,b_1 < 2^{30}$。

树的重心（centroid）

【题目描述】

小简单正在学习离散数学，今天的内容是图论基础，在课上他做了如下两条笔记：

1. 一个大小为 n 的树由 n 个结点与 $n-1$ 条无向边构成，且满足任意两个结点间**有且仅有**一条简单路径。在树中删去一个结点及与它关联的边，树将分裂为若干个子树；而在树中删去一条边（保留关联结点，下同），树将分裂为**恰好**两个子树。
2. 对于一个大小为 n 的树与任意一个树中结点 c，称 c 是该树的**重心**当且仅当在树中删去 c 及与它关联的边后，分裂出的所有子树的大小均**不超过** $\lfloor \frac{n}{2} \rfloor$（其中 $\lfloor x \rfloor$ 是下取整函数）。对于包含至少一个结点的树，它的重心只可能有 1 或 2 个。

课后老师给出了一个大小为 n 的树 S，树中结点从 $1 \sim n$ 编号。小简单的课后作业是求出 S 单独删去每条边后，分裂出的两个子树的重心编号和之和。即：

$$\sum_{(u,v)\in E}\left(\sum_{\substack{1\le x\le n \\ \text{且 } x \text{ 号点是 } S'_u \text{ 的重心}}} x+\sum_{\substack{1\le y\le n \\ \text{且 } y \text{ 号点是 } S'_v \text{ 的重心}}} y\right)$$

上式中，E 表示树 S 的边集，(u,v) 表示一条连接 u 号点和 v 号点的边。S'_u 与 S'_v 分别表示树 S 删去边 (u,v) 后，u 号点与 v 号点所在的被分裂出的子树。

小简单觉得作业并不简单，只好向你求助，请你教教他。

【输入格式】

从文件 *centroid.in* 中读入数据。

本题输入包含多组测试数据。

第一行一个整数 T 表示数据组数。

接下来依次给出每组输入数据，对于每组数据：

第一行一个整数 n 表示树 S 的大小。

接下来 $n-1$ 行，每行两个以空格分隔的整数 u_i, v_i，表示树中的一条边 (u_i, v_i)。

【输出格式】

输出到文件 *centroid.out* 中。

共 T 行，每行一个整数，第 i 行的整数表示：第 i 组数据给出的树单独删去每条边后，分裂出的两个子树的重心编号和之和。

【样例 1 输入】

```
2
5
```

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

【样例 1 输出】

```
32
56
```

【样例 1 解释】

对于第一组数据：
删去边 $(1,2)$，1 号点所在子树重心编号为 $\{1\}$，2 号点所在子树重心编号为 $\{2,3\}$。
删去边 $(2,3)$，2 号点所在子树重心编号为 $\{2\}$，3 号点所在子树重心编号为 $\{3,5\}$。
删去边 $(2,4)$，2 号点所在子树重心编号为 $\{2,3\}$，4 号点所在子树重心编号为 $\{4\}$。
删去边 $(3,5)$，3 号点所在子树重心编号为 $\{2\}$，5 号点所在子树重心编号为 $\{5\}$。
因此答案为 $1+2+3+2+3+5+2+3+4+2+5=32$。

【样例 2】

见选手目录下的 *centroid/centroid2.in* 与 *centroid/centroid2.ans*。

【样例 3】

见选手目录下的 *centroid/centroid3.in* 与 *centroid/centroid3.ans*。
该数据满足特殊性质 A，具体信息见数据范围中的描述。

【样例 4】

见选手目录下的 *centroid/centroid4.in* 与 *centroid/centroid4.ans*。
该数据满足特殊性质 B，具体信息见数据范围中的描述。

【数据范围】

测试点编号	$n =$	特殊性质
$1 \sim 2$	7	无
$3 \sim 5$	199	
$6 \sim 8$	1999	
$9 \sim 11$	49991	A
$12 \sim 15$	262143	B
16	99995	无
$17 \sim 18$	199995	
$19 \sim 20$	299995	

表中特殊性质一栏，两个变量的含义为存在一个 $1 \sim n$ 的排列 p_i（$1 \le i \le n$），使得：

A：树的形态是一条链。即 $\forall 1 \le i < n$，存在一条边 (p_i, p_{i+1})。

B：树的形态是一个完美二叉树。即 $\forall 1 \le i \le \frac{n-1}{2}$，存在两条边 (p_i, p_{2i}) 与 (p_i, p_{2i+1})。

对于所有测试点：$1 \le T \le 5$, $1 \le u_i, v_i \le n$。保证给出的图是一个树。