

冲刺 CCF NOIP2024 模拟试题

BSZX

时间：2024 年 11 月 12 日 08:00 ~ 12:30

题目名称	树	宠物狗	波特方阵	工厂
题目类型	传统型	传统型	传统型	传统型
目录	tree	dog	matrix	factory
可执行文件名	tree	dog	matrix	factory
输入文件名	tree.in	dog.in	matrix.in	factory.in
输出文件名	tree.out	dog.out	matrix.out	factory.out
每个测试点时限	1.0 秒	3.0 秒	1.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	5	6	7	12
测试点是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	tree.cpp	dog.cpp	matrix.cpp	factory.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++ 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`。
- 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。
- 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。
- 直接复制 PDF 题面中的多行样例，数据将带有行号，并且某些字符可能无法正常显示，建议选手直接使用对应目录下的样例文件进行测试。
- 评测时采用的机器配置为：Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz，内存 32GB。上述时限以此配置为准。

树 (tree)

【题目描述】

给定一棵 n 个点的树，对于每个 k ($1 \leq k \leq n$)，定义一个无向图 G_k ，其中两个节点 u, v 之间有边当且仅当它们在树上的距离至少为 k 。请你计算 G_k 的连通块个数。

【输入格式】

从文件 `tree.in` 中读入数据。
第一行包含整数 n 。
接下来 $n - 1$ 行，每行包含两个整数 u 和 v ，表示树上的一条边。

【输出格式】

输出到文件 `tree.out` 中。
输出一行 n 个整数，第 k 个数表示 G_k 的连通块个数。

【样例 1 输入】

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

【样例 1 输出】

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

【样例 1 解释】

$k = 4$ 时，联通块为 $\{1\}, \{2\}, \{3\}, \{4, 5, 6\}$ 。

【样例 2 输入】

```
1 5
2 1 2
```

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

【样例 2 输出】

```
1 1 1 3 5 5
```

【样例 2 解释】

$k = 3$ 时，联通块为 $\{1, 4, 5\}, \{2\}, \{3\}$ 。

【样例 3】

见选手目录下的 `tree/tree3.in` 与 `tree/tree3.ans`。
该样例满足子任务 2 的限制。

【样例 4】

见选手目录下的 `tree/tree4.in` 与 `tree/tree4.ans`。
该样例满足子任务 5 的限制。

【数据范围】

对于所有数据，保证 $2 \leq n \leq 5 \times 10^5, 1 \leq u, v \leq n$ 。

子任务编号	$n \leq$	特殊性质	分值
1	500	无	20
2	5000		30
3	5×10^5	树为一条链	10
4		树为一个菊花	10
5		无	30

宠物狗 (dog)

【题目描述】

bot 是一位宠物狗商家。

最近，它进货了 n 条狗并排成一排。这些狗有 26 种不同的品种，从 a 到 z 依次标注，我们用 s_i 表示第 i 条狗的品种。

bot 将选择 $d \in \{1, 2, \dots, n\}$ ，即狗宠物栏的宽度。然后，他把狗群分为若干连续段，得到长度为 d 的 $\lfloor \frac{n}{d} \rfloor$ 排狗和一排长度为 $n \bmod d$ 的狗（可能为 0）。注意这 $\lfloor \frac{n}{d} \rfloor$ 排狗都是连续段。

所有长度为 d 的 $\lfloor \frac{n}{d} \rfloor$ 排将被排列成宠物栏，每排都是之前的相对顺序。

bot 想知道它能得到多少个不同的宠物栏。当且仅当两个宠物栏的宽度不同，或者至少有一个对应位置的狗的品种不同时，才会认为它们是不同的。

由于结果可能非常大，因此输出答案模 998244353。

【输入格式】

从文件 *dog.in* 中读入数据。

本题有多组测试数据。

第一行一个整数 T ，表示数据组数。

对于每组数据，一行包含一个字符串 s 。

【输出格式】

输出到文件 *dog.out* 中。

共 T 行，每行一个数表示每组数据的答案。

【样例 1 输入】

```
1 2
2 aab
3 ababccd
```

【样例 1 输出】

```
1 6
2 661
```

【样例 1 解释】

对于第一组询问, $d = 1$ 时分割方式为 $\{[1, 1], [2, 2], [3, 3]\}$, 有 3 种不同的方案; $d = 2$ 时分割方式为 $\{[1, 2], [3, 3]\}, \{[1, 1], [2, 3]\}$, 其中 $[1, 1], [3, 3]$ 长度不为 d , 不用于排列, 有 2 种不同的方案; $d = 3$ 时分割方式为 $\{[1, 3]\}$, 有 1 种不同的方案。共 6 种方案。

对于第二组询问, 共有 661 种不同的方案。

【样例 2】

见选手目录下的 *dog/dog2.in* 与 *dog/dog2.ans*。

该样例满足子任务 1 的限制。

【样例 3】

见选手目录下的 *dog/dog3.in* 与 *dog/dog3.ans*。

该样例满足子任务 2 的限制。

【样例 4】

见选手目录下的 *dog/dog4.in* 与 *dog/dog4.ans*。

该样例满足子任务 3 的限制。

【样例 5】

见选手目录下的 *dog/dog5.in* 与 *dog/dog5.ans*。

该样例满足子任务 4 的限制。

【样例 6】

见选手目录下的 *dog/dog6.in* 与 *dog/dog6.ans*。

该样例满足子任务 5 的限制。

【样例 7】

见选手目录下的 *dog/dog7.in* 与 *dog/dog7.ans*。

该样例满足子任务 6 的限制。

【数据范围】

对于所有数据, 保证 $1 \leq T \leq 10^5, n \leq 2 \times 10^5, \sum n \leq 10^6, s_i \in \{a, b, c, , z\}$ 。

子任务编号	$n \leq$	$\sum n \leq$	特殊性质	分值
1	8	100	无	15
2	500	5000		15
3	2×10^5	10^6	$s_i \in \{a\}$	10
4	10^5	2×10^5	s 随机生成	10
5			无	25
6	2×10^5	10^6		25

波特方阵 (matrix)

【题目描述】

有 $n \times m$ 个 bot 排成一个 n 行 m 列的方阵，第 i 行 j 列的 bot 型号为 $a_{i,j}$ 。你可以进行如下两种操作若干次（也可以不操作）：

- 选择一个整数 j ，然后对于第 j 列的所有 bot，位于第 i 行的移动到第 p_i 行。即对于 $1 \leq i \leq n$ ，同时令 $a_{p_i,j} \leftarrow a_{i,j}$ 。其中 p 为给定的一个排列。
- 选择一个整数 i ，然后对于第 i 行的所有 bot，位于第 j 列的移动到第 q_j 列。即对于 $1 \leq j \leq m$ ，同时令 $a_{i,q_j} \leftarrow a_{i,j}$ 。其中 q 为给定的一个排列。

你想知道最终能得到多少种不同的方阵，答案对 998244353 取模。
两个方阵被视作不同的，当且仅当存在一个位置使得两个 bot 的型号不同。

【输入格式】

从文件 `matrix.in` 中读入数据。
本题有多组测试数据。
第一行一个整数 T ，表示数据组数。
对于每组数据：
第一行两个整数 n, m ，表示方阵的行数列数。
接下来一行 n 个整数，表示排列 p 。
接下来一行 m 个整数，表示排列 q 。
接下来 n 行，每行 m 个整数。第 i 行 j 列的数表示当前位置 bot 的型号 $a_{i,j}$ 。

【输出格式】

输出到文件 `matrix.out` 中。
输出共 T 行，每行一个数表示每组数据的答案。

【样例 1 输入】

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

```
8 1 3 2 4
9 1 3 2 4
10 1 2 3 4
11 3 4 1 2
12 1 2 4 1
13 4 3 3 2
14 3 9
15 1 3 2
16 1 6 2 7 3 8 4 9 5
17 1 8 1 1 8 1 1 8 1
18 1 8 8 8 8 8 8 8 1
19 1 1 1 8 8 8 1 1 1
```

【样例 1 输出】

```
1 6
2 96
3 6336
```

【样例 1 解释】

对于第一组数据，能形成的方阵有：

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix}, \begin{bmatrix} 2 & 2 & 1 \\ 1 & 2 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 2 \\ 2 & 2 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 2 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 2 \\ 1 & 2 & 2 \end{bmatrix}, \begin{bmatrix} 2 & 2 & 1 \\ 2 & 2 & 1 \end{bmatrix}$$

【样例 2】

见选手目录下的 *matrix/matrix2.in* 与 *matrix/matrix2.ans*。
该样例满足子任务 1 的限制。

【样例 3】

见选手目录下的 *matrix/matrix3.in* 与 *matrix/matrix3.ans*。
该样例满足子任务 2 的限制。

【样例 4】

见选手目录下的 *matrix/matrix4.in* 与 *matrix/matrix4.ans*。
该样例满足子任务 3 的限制。

【样例 5】

见选手目录下的 *matrix/matrix5.in* 与 *matrix/matrix5.ans*。
该样例满足子任务 4 的限制。

【样例 6】

见选手目录下的 *matrix/matrix6.in* 与 *matrix/matrix6.ans*。
该样例满足子任务 5 的限制。

【样例 7】

见选手目录下的 *matrix/matrix7.in* 与 *matrix/matrix7.ans*。
该样例满足子任务 6 的限制。

【样例 8】

见选手目录下的 *matrix/matrix8.in* 与 *matrix/matrix8.ans*。
该样例满足子任务 7 的限制。

【数据范围】

对于所有数据，保证 $1 \leq T \leq 10^5, 1 \leq \sum nm \leq 10^6, 1 \leq a_{i,j} \leq 10^6$ ， p 为 $1 \sim n$ 的排列， q 为 $1 \sim m$ 的排列。

子任务编号	$\sum nm \leq$	特殊性质	分值
1	40	$nm \leq 8$	15
2	3×10^5	$p_i = i$	10
3		$p_i = (i \bmod n) + 1, q_i = (i \bmod m) + 1$ 且 nm 为奇数	10
4		$p_i = (i \bmod n) + 1, q_i = (i \bmod m) + 1$	15
5		$n = m$ 且 $p = q$	10
6		无	20
7	10^6		20

工厂 (factory)

【题目描述】

有一个工厂，其中共有 n 个仓库和 m 个 **bot**。初始时，第 i 个仓库还能再装下 c_i 个货物。第 i 个 **bot** 有 v_i 个货物，它能在仓库 l_i 和 r_i 之间卸货。

bot 分为两种类型。如果 **bot** 的类型 $t_i = 0$ ，则其卸货范围始终保持不变。而 **bot** 类型 $t_i = 1$ 则可以改变它的范围。具体地，如果将编号为 x 的仓库指定为主仓，则每个类型为 1 的 **bot** 能在仓库 $\min(l_i, x)$ 和 $\max(r_i, x)$ 之间卸货。

对于 $1 \sim n$ 的每个 x ，你需要求出当仓库 x 为主仓时，**bot** 们最多能卸下多少货物。

【输入格式】

从文件 *factory.in* 中读入数据。

本题有多组测试数据。

第一行一个整数 T ，表示数据的组数。

对于每组数据：

第一行两个整数 n 和 m ，分别表示仓库和 **bot** 的数量。

接下来一行 n 个整数 c_1, c_2, \dots, c_n ，表示仓库的容量。

接下来 m 行，每行四个整数 l_i, r_i, v_i, t_i ，表示 **bot** 的设定范围、初始数量和 **bot** 类型。

【输出格式】

输出到文件 *factory.out* 中。

对于每组数据，输出一行共 n 个整数，第 i 个数表示 $x = i$ 时的答案。

【样例 1 输入】

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

【样例 1 输出】

1 8 7 7 8

【样例 1 解释】

当 $x = 1$ 时, 3 号 bot 设定范围改为 $[1, 2]$ 。

最优方案为: 1 号 bot 给 1 号仓库 2 个, 2 号 bot 给 3 号仓库 2 个, 3 号 bot 给 1 号仓库 1 个, 3 号 bot 给 2 号仓库 3 个, 答案为 8。

当 $x = 2$ 时, 3 号 bot 设定范围不变, 答案为 7。

当 $x = 3$ 时, 3 号 bot 设定范围改为 $[2, 3]$, 答案为 7。

当 $x = 4$ 时, 3 号 bot 设定范围改为 $[2, 4]$, 答案为 8。

【样例 2】

见选手目录下的 *factory/factory2.in* 与 *factory/factory2.ans*。

该样例满足子任务 1 的限制。

【样例 3】

见选手目录下的 *factory/factory3.in* 与 *factory/factory3.ans*。

该样例满足子任务 2 的限制。

【样例 4】

见选手目录下的 *factory/factory4.in* 与 *factory/factory4.ans*。

该样例满足子任务 3 的限制。

【样例 5】

见选手目录下的 *factory/factory5.in* 与 *factory/factory5.ans*。

该样例满足子任务 4 的限制。

【样例 6】

见选手目录下的 *factory/factory6.in* 与 *factory/factory6.ans*。

该样例满足子任务 5 的限制。

【样例 7】

见选手目录下的 *factory/factory7.in* 与 *factory/factory7.ans*。

该样例满足子任务 6 的限制。

【样例 8】

见选手目录下的 *factory/factory8.in* 与 *factory/factory8.ans*。
该样例满足子任务 7 的限制。

【样例 9】

见选手目录下的 *factory/factory9.in* 与 *factory/factory9.ans*。
该样例满足子任务 8 的限制。

【样例 10】

见选手目录下的 *factory/factory10.in* 与 *factory/factory10.ans*。
该样例满足子任务 9 的限制。

【样例 11】

见选手目录下的 *factory/factory11.in* 与 *factory/factory11.ans*。
该样例满足子任务 10 的限制。

【样例 12】

见选手目录下的 *factory/factory12.in* 与 *factory/factory12.ans*。
该样例满足子任务 11 的限制。

【样例 13】

见选手目录下的 *factory/factory13.in* 与 *factory/factory13.ans*。
该样例满足子任务 12 的限制。

【数据范围】

对于所有数据，保证 $1 \leq T \leq 2 \times 10^5, 1 \leq n, m \leq 2 \times 10^5, 0 \leq c_i \leq 10^9, 1 \leq l_i \leq r_i \leq n, 0 \leq v_i \leq 10^9, t_i \in \{0, 1\}$ 。

子任务编号	$\sum n \leq$	$\sum m \leq$	特殊性质	分值
1	100	100	$m = 1$	10
2			无	5
3	2000	2000		5
4	2×10^4	200		5
5	10^5	2000		5
6	2×10^4	2×10^4	$t_i = 1$	10
7	10^5	10^5	$l_i \leq l_{i+1}, r_i \leq r_{i+1}, t_i = 1$	5
8			$t_i = 1$	10
9			对于所有 $t_i = 0$ 的 bot , $r_i \leq 50$ 或 $l_i > n - 50$	10
10			$a_i = 1$	5
11			无	15
12				15