

CSP 2019 模拟题

题目名称	扫雷	缘分	阅读
英文名	mine	fate	book
时间限制	1s	1s	1s
空间限制	512MB	512MB	512MB
测试点个数	20	20	10

注意：评测时开启O2优化。

扫雷

题目描述

你是倒序开题的。

此时此刻，你已经轻松AK了这场考试，并在没有看过题目描述的情况下切掉了本题。

百无聊赖的你决定在剩下的三个小时里做些什么有意义的事，你想到了扫雷。但是，久经沙场的你已经厌倦了Windows自带的扫雷游戏，于是你自己手写了一个扫雷。然而全随机摆雷随机性太大，为了降低难度，你选择保证其中 k 个特定的位置必须是雷。

你发现，你的游戏体验与初始地雷的「布局」关系十分密切。如果雷太少，随便点击几下就能无脑轻松通过；而雷太多，又会令你频繁陷入2选1，3选1的局面，毫无游戏体验可言。而且有些布局的难度波动太小，千篇一律，这也不是你想要看到的。

于是你开始对这样一件事情兴致勃勃——究竟怎样的「布局」才是最好玩的？

在这里，我们规定一种「布局」是指规定网格的大小为 n 行 m 列，总共有 w 个雷，并且有 k 个雷的位置已经确定，其余的雷全随机安排。及所有的 $\binom{nm-k}{w-k}$ 种情况的概率相等。

你发现，每个格子中的数字之和⁽¹⁾能很好地衡量一场扫雷的质量，数字越大，代表雷越多，逻辑推理量越大。而波动程度越大，越能刺激玩家的探索欲望，保留玩家的游戏热情。（以上都是在乱扯）。

你挑选了你认为最好玩的几个布局，并想知道这些布局的「难度」和「波动程度」分别如何。

形式化的，我们说一个「布局」的每个格子中数字之和为随机变量 ξ ，我们定义一个布局的「难度」为 ξ 的期望，即 $E(\xi)$ ；定义「波动程度」为 ξ 的方差，即 $D(\xi)$ ⁽²⁾。

你发现求出精确的分数值非常繁琐，退而求其次，你希望求出 $E(\xi)$ 和 $D(\xi)$ 对998244353取模的结果。

输入格式

第一行四个整数 n, m, w, k ，分别表示网格的行数，列数，总雷数，已确定的雷数。

接下来 k 行，每行两个整数 x_i, y_i ，表示第 x 行第 y 列必然是雷。

输出格式

一行两个整数 $E(\xi)$ 和 $D(\xi)$ ，用空格隔开。

样例

输入样例 1

```
1 | 2 2 2 1
2 | 2 2
```

输出样例 1

```
1 | 4 0
```

样例1解释

有 2×2 的棋盘，右下角必是雷，此外另三处有且仅有一处是雷，且三种情况的概率相同。并且不论那种情况，两个空地上的数字必然都是2，所以 $\xi \equiv 4$, $E(\xi) = 4$, $D(\xi) = 0$ 。

输入样例 2

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

输出样例 2

```
1 | 665496239 554580197
```

样例2解释

有 2×3 的棋盘，有一处是雷。假如在角上，数字之和为3，否则为5。因此

$$E(\xi) = \frac{1}{3} \times 5 + \frac{2}{3} \times 3 = \frac{11}{3}$$
$$D(\xi) = \frac{1}{3} \times (5 - \frac{11}{3})^2 + \frac{2}{3} \times (3 - \frac{11}{3})^2 = \frac{8}{9}$$

容易验证它们对998244353取模的分别为665496239和554580197。

输入样例 3

```
1 | 6 6 10 5
2 | 5 2
3 | 3 1
4 | 1 2
5 | 3 2
6 | 5 4
```

输出样例 3

```
1 | 461553889 995270246
```

数据范围

测试点编号	$n \leq$	$m \leq$	$nm \leq$	$k \leq$
1	2	3	-	-
2	2	3	-	-
3	5	5	-	-
4	5	5	-	-
5	5	10	-	-
6	5	10	-	-
7	5	1000	-	-
8	5	1000	-	-
9	-	-	2500	-
10	-	-	2500	-
11	-	-	2500	-
12	-	-	2500	-
13	1	-	-	0
14	1	-	-	-
15	2	-	-	0
16	2	-	-	-
17	-	-	400000	-
18	-	-	400000	-
19	-	-	400000	-
20	-	-	400000	-

上表中「-」表示没有特殊限制

对所有数据，满足 $1 \leq nm \leq 4 \times 10^5$ ， $0 \leq k \leq w \leq nm$ ， $1 \leq x \leq n$ ， $1 \leq y \leq m$ 。

且保证 $\forall i \neq j, x_i \neq x_j$ 或 $y_i \neq y_j$ 。

提示

- (1). 一个格子上的数字是指与这个格子**八连通**的格子中的雷数，但注意本题中如果该格有雷，此格上的数字算作0。你可以通过自己电脑上的扫雷来理解，也可以看下图：



- (2). $D(\xi) = E\left((\xi - E(\xi))^2\right)$

缘分

题目描述

ν 居住在一个神奇的国度，这个国家由 n 个城市组成，城市和城市之间有 $n - 1$ 条铁路连接，而且构成一个树形结构，有趣的是，每条铁路的长度是一样的。

ν 过着两点一线的平淡生活，每天上午从 a 市出发去 b 市上班，下午回家。然而某一天，他的生活迎来了转机。一天上班时，他在 e 转车时遇到了分别多年的老同学 v ，两人在火车上相谈甚欢。可是美好的时光总是有限的，若干站路后，他们在 f 市再次分道扬镳。

令人意想不到的，下午 ν 又在 f 遇到了 v ，而且更巧的是，从家出发到在 e 相遇的时间，恰好等于从公司出发到在 f 相遇的时间。

ν 和 v 发现，之所以会出现这样的情况，是因为 v 的家和公司分别为 c, d ，而恰好有 $\text{dis}(a, e) = \text{dis}(c, e) = \text{dis}(b, f) = \text{dis}(d, f)$ 。

ν 在感慨之虞，也想知道还有哪些城市的人能享有这样奇妙的缘分。 v 提议用一个**无序**的四元组 (a, b, c, d) 来表示一组有缘的城市， a, b, c, d **两两不同**。一个四元组是有缘的，当且仅当存在一种这四个元素的排列 $(\iota, \psi, \tau, \zeta)$ 并满足以下性质：

1. 路径 $\langle \iota, \tau \rangle$ 和 $\langle \psi, \zeta \rangle$ 有交。我们取它们的交为为路径 $\langle \epsilon, \delta \rangle$ 。
2. ι 和 ψ 在 ϵ 的一侧， τ 和 ζ 在 δ 的一侧
3. $\text{dis}(\iota, \epsilon) = \text{dis}(\psi, \epsilon) = \text{dis}(\tau, \delta) = \text{dis}(\zeta, \delta)$ 。

现在 ν 和 v 想知道，在所有的 $\binom{n}{4}$ 种可能的城市组合中，总共有多少组有缘的城市。由于答案可能很大，你只需算出它对998244353取模的结果。

输入格式

第一行一个整数 n ，表示城市的个数。

接下来 $n - 1$ 行每行两个整数 u, v ，表示 u, v 之间有一条铁路连接。保证将给出一棵无根树。

输出格式

一行一个整数，表示有缘的城市四元组个数。

样例

输入样例 1

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

输出样例 1

1	4
---	---

样例1解释

合法的四元组有四个：（后面给出了一种可能的解释）

a, b, c, d	$\iota, \psi, \tau, \zeta, \epsilon, \delta$
2, 3, 5, 6	2, 3, 5, 6, 1, 1
2, 3, 4, 7	2, 3, 4, 7, 1, 6
2, 4, 5, 7	2, 5, 4, 7, 1, 6
3, 4, 5, 7	3, 5, 4, 7, 1, 6

输入样例 2

1	20
2	2 1
3	3 1
4	4 1
5	5 4
6	6 1
7	7 6
8	8 5
9	9 6
10	10 4
11	11 1
12	12 7
13	13 1
14	14 6
15	15 1
16	16 1
17	17 1
18	18 1
19	19 6
20	20 5

输出样例 2

1 | 516

数据范围

- 对于20%的数据， $n \leq 50$ 。
- 对于40%的数据， $n \leq 300$ 。
- 对于60%的数据， $n \leq 3000$ 。
- 对于80%的数据， $n \leq 100000$ 。
- 对于100%的数据， $1 \leq n \leq 400000$ 。

提示

- 本题中出现的希腊字母的念法：

ν	υ	ι	ψ	τ	ζ	ϵ	δ
nu	upsilon	iota	psi	tau	zeta	epsilon	delta

阅读

题目描述

你的一位朋友 M 有十分奇怪的阅读癖好。

这一次， M 又新买了一些书，但碰巧他的书架垮了，他只能把书成摞的堆在地上。总共有 n 摞书，第 i 摞书有 l_i 本。每一本书有一个封面颜色，我们用小写英文字母表示。

M 每天会看一次书，他将从**不同的**几摞书中可选出一本。看书之前，需要先找到对应的那一摞，并把他想看的书上面所有的书拿下来。此后他将把所有书恢复原来的摆放方式。

此外为了防止混淆， M 不希望从每一摞书上拿下来的部分（包括他要看的书）之间不存在某一条颜色序列为另一段的子串的情况。

M 希望在满足上述要求的情况下，以最短的时间看完所有的书至少一遍。作为他的朋友，你必须写一个程序帮他求出这一最短时间。

输入格式

第一行一个正整数 n ，表示书的摞数。

接下来 n 行，每行一个小写字母字符串 s_i ，从上到下描述第 i 摞书每本的颜色。

输出格式

一行一个正整数，表示看完所有书的最小天数。

样例

输入样例 1

```
1 | 3
2 | aba
3 | aa
4 | baa
```

输出样例 1

```
1 | 4
```

样例1解释

我们用 (i, j) 表示第 i 摞从上往下的第 j 本。用符号 \sim 表示两本书不能放在一天看，显然可以得到 $(1, 1) \sim (2, 1), (1, 3) \sim (3, 2)$ 等。存在一种4天看完的方案：

第一天	第二天	第三天	第四天
$(1, 1)$	$(2, 1), (3, 1)$	$(1, 2), (2, 2), (3, 2)$	$(1, 3), (3, 3)$

可以证明不存在少于4天看完的方案。

输入样例 2

```
1 | 4
2 | h1cu
3 | oeas
4 | psc
5 | est
```

输出样例 2

```
1 | 4
```

数据范围

用 $\sum l$ 表示所有书的总本数。

对于10%的数据, $n \leq 5$, $\sum l \leq 20$ 。

对于20%的数据, $\sum l \leq 500$ 。

对于50%的数据, $\sum l \leq 5000$ 。

对于100%的数据, $1 \leq n \leq \sum l \leq 10^6$ 。