

2024 NOIP模拟赛

| 题目名称 | 迷宫 | 玩具 | 权重 | 周长 |
|---------|----------|---------|------------|---------------|
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 目录 | maze | toy | weight | perimeter |
| 可执行文件名 | maze | toy | weight | perimeter |
| 输入文件名 | maze.in | toy.in | weight.in | perimeter.in |
| 输出文件名 | maze.out | toy.out | weight.out | perimeter.out |
| 每个测试点时限 | 1.5秒 | 1.0秒 | 1.0秒 | 1.0秒 |
| 内存限制 | 512M | 512M | 512M | 512M |
| 测试点数目 | 20 | 20 | 20 | 25 |
| 测试点是否等分 | 是 | 是 | 是 | 是 |

提交源程序文件名

| 对于C++语言 | maze.cpp | toy.cpp | weight.cpp | perimeter.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/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。
- 提交的程序代码文件的放置位置请参考各省的具体要求。
- 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。
- 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。
- 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。
- 只提供 Linux 格式附加样例文件。
- 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此为准。

T1 迷宫 (maze)

题目背景

在一个遥远的魔法世界里，有一个名叫小 H 的年轻法师。他住在一座神秘的迷宫城堡中，这座城堡由无尽的走廊和房间构成，墙壁上布满了奇异的符文和闪烁的光点。小 H 非常喜欢探索这个迷宫，然而，这次他的冒险却与众不同。

问题描述

一天，小 H 收到了来自城堡另一端的信件，信中提到他的好友小 M 被困在一个古老的陷阱中，急需帮助。小 H 的心中充满了焦虑，他决定立刻前往救援。然而，迷宫的结构复杂，通道曲折，让他不得不面对艰难的挑战。

小 H 站在迷宫的起点 S 处，四周是一片宁静的黑暗，只有微弱的光线透过石缝洒落在地面上。他深吸一口气，开始了他的冒险。迷宫的墙壁上布满了不可穿越的障碍物（用字符 `#` 表示），而那些可以通行的道路则是光滑的地面（用字符 `.` 表示）。小 H 记得在迷宫的另一端，有一个神秘的终点 T ，那里正是他的好友小 M 被困的地方。

小 H 的移动方式非常灵活，他可以花费 1 的时间选择向上、下、左、右移动一步。然而，迷宫中的障碍物让他每一步都需要谨慎考虑。为了加快速度，小 H 还拥有一种强大的魔法能力，他可以在自己的位置上施展魔法，花费 t 的时间到以距离当前格子不超过 k 的任意位置。这让他面对复杂的迷宫时，能够迅速穿越障碍，寻找出路。

随着小 H 的不断探索，他发现迷宫中的每一个转角都藏着未知的危险与机遇。有时他会遇到迷宫中的守卫，他们是由魔法构成的生物，阻挡着他的去路；有时他会发现隐藏的宝藏，给予他额外的力量来应对接下来的挑战。

小 H 明白，时间紧迫，他必须尽快找到通往终点 T 的道路。他开始思考，怎样才能尽可能花费最短的时间，顺利到达小 M 的身边。每一次魔法的施展都需要谨慎计算，因为他知道，只有在关键时刻才能发挥出最大的效果。

在这场充满挑战的冒险中，小 H 不仅要依靠自己的智慧和勇气，还要善用他的魔法，才能最终打破迷宫的束缚，救出他的好友。时间在流逝，迷宫的秘密等待着他去揭开，而小 H 的旅程才刚刚开始.....

输入格式

从文件 `maze.in` 中读入数据。

第一行，四个正整数 n, m, k, t 。

接下来 n 行，每行 m 个字符描述迷宫。

其中 `#` 表示障碍，`.` 表示平地， S 表示起点， T 表示终点。

输出格式

输出到文件 `maze.out` 中。

如果无解，请输出 -1 ，否则输出一个正整数表示最短的时间。

样例1

输入样例

```
1 4 4 3 4
2 S.#.
3 ..#.
4 .#..
5 .#.T
```

输出样例

```
1 7
```

样例2

见选手目录下的 `maze/maze2.in` 与 `maze/maze2.ans`。

这个样例满足测试点 4 的约束条件。

数据范围

对于所有数据，满足 $1 \leq n, m \leq 2000$ ， $1 \leq k \leq 8$ ， $1 \leq t \leq 10^9$ ，保证 S 和 T 只出现一次。

| 测试点 | $n, m \leq$ | 特殊性质 |
|---------|-------------|----------------------|
| 1 ~ 3 | 5 | 无 |
| 4 ~ 6 | 10 | 无 |
| 7 ~ 10 | 100 | 无 |
| 11 ~ 14 | 2000 | $t = 1$ |
| 15, 16 | 2000 | 保证不存在 <code>#</code> |
| 17 | 2000 | 保证 S 和 T 相邻 |
| 18, 19 | 2000 | $t = 10^9$ |
| 20 | 2000 | 无 |

T2 玩具(toy)

问题描述

小 T 买了一个毛毛虫玩具。毛毛虫的构造非常简单，由上下两部分组成：

- 上半部分是毛毛虫的身体，长度为 a ；
- 下半部分是毛毛虫的脚，每只脚占一个单位长度，毛毛虫共有 b 只脚。

一开始毛毛虫的身体在最左边 a 个位置，脚在最左边 b 个位置。最终，毛毛虫的身体要在最右边 a 个位置，脚要在最右边 b 个位置。

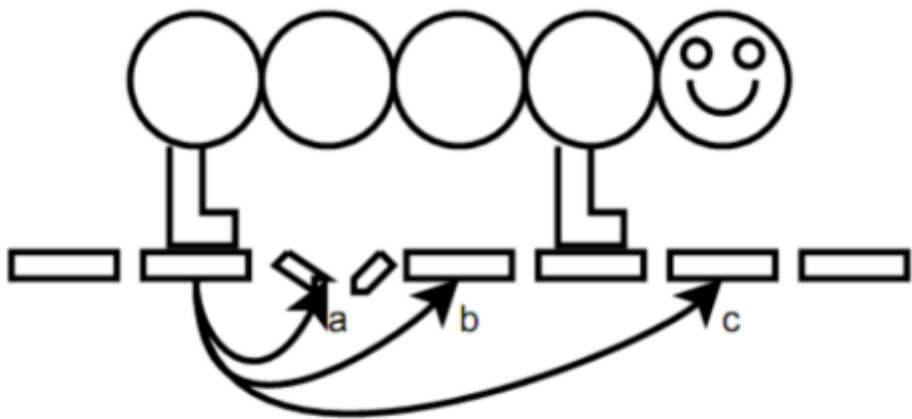


图 1: 示例

现在，毛毛虫要从木板的左边爬到右边。木板的长度为 n ，由于小 T 比较穷，用了很多年的木板在某些位置已经破了。毛毛虫不能将脚放在这些破的位置。

众所周知，毛毛虫是通过蠕动爬行的，因此，毛毛虫的运动分为 2 种操作：

1. 将某一只脚向右移动若干距离，要求最后的落脚点的木板不能是破的，并且，要严格在前一只脚的左边；
2. 将上半部分身体向右移动 1 个距离，要求移动后所有的脚仍然在身体下方。

这两种的操作均需要花费 1 的时间。现在，小 T 希望聪明的你告诉他，毛毛虫爬到最右边（即身体的最右一格在 n 号位置，并且所有的脚也在最右边）最少需要多少时间。当然，有的时候，小 T 家的木板已经残破不堪，毛毛虫无法到达最右边，那么请输出 `IMPOSSIBLE`。

问题保证最开始的最开始的 b 个位置和最后的最后的 b 个位置木板都是好的。

输入格式

从文件 `toy.in` 中读入数据。

第一行包含三个正整数 b, a, n ，分别表示身体长度、脚的长度和木板长度。

第二行包含 n 个 `01` 字符，其中 `0` 表示当前位置木板是破的，`1` 表示当前位置木板完好。

输出格式

输出到文件 `toy.out` 中。

一行一个数，表示毛毛虫花费的最少时间，无解输出 `IMPOSSIBLE`。

样例1

输入样例

```
1 1 3 5
2 11011
```

输出样例

```
1 5
```

样例2

见选手目录下的 `toy/toy2.in` 与 `toy/toy2.ans`。

这个样例满足测试点 12 的约束条件。

样例3

见选手目录下的 `toy/toy3.in` 与 `toy/toy3.ans`。

这个样例满足测试点 16 的约束条件。

数据范围

对于所有数据，满足 $1 \leq b \leq a \leq n \leq 3 \times 10^6$ 。

| 测试点 | $n \leq$ | 特殊性质 |
|---------|-----------------|---------|
| 1 ~ 4 | 10 | 无 |
| 5 ~ 8 | 3×10^6 | $b = 1$ |
| 9 ~ 11 | 3×10^6 | 没有破地板 |
| 12 ~ 16 | 10^4 | 无 |
| 17 ~ 20 | 3×10^6 | 无 |

T3 权重(weight)

题目背景

夜里，应该已经过了 12 点，你因为之前小睡了一会，现在非常精神，于是决定再做点什么。回想起白天老师对你的建议：想清楚自己的目标。

于是你决定来总结一下。

问题描述

你根据自己掌握的技能之间的关系，绘制出了一棵 n 个节点的技能树，树的节点代表一个技能，而树上的任意一条简单路径则代表了一个发展方向。

每个点有一个权重 w_i ，代表这个点的技能的效果。一条路径的价值则是其包含的点的权重按位异或的结果。

现在你需要根据这幅图决定发展方向的起点。受困于现实情况，你认为每个点会在以他为端点的 n 条路径中随机选取一条，称为该点的实际路径，这条路径的价值就是这个点的实际价值。你将通过接下来一个阶段的研究搞明白每个点的实际价值，并从中选择任意一个实际价值最大的点作为发展的起点，该点的实际价值就是这棵技能树的价值。

虽然你现在还不知道每个点的具体实际价值，但是你假设每个点的实际路径将在 n 个候选方案中等概率随机。你希望知道最终技能树的价值大于等于 t 的概率。

请输出概率乘 n^n 后对 998244353 取模的结果。

输入格式

从文件 `weight.in` 中读入数据。

输入第一行，包含一个正整数 n ，表示树的节点个数。

接下来一行，每行两个正整数 u, v ，表示树中的一条边 (u, v) 。

接下来一行 n 个非负整数，第 i 个是 w_i 代表节点 i 的权重。

接下来一行一个非负整数 t ，含义如题目描述所示。

输出格式

输出到文件 `weight.out` 中。

输出一个非负整数，表示概率乘 n^n 对 998244353 取模的结果。

样例1

输入样例

| | |
|---|---------|
| 1 | 4 |
| 2 | 1 2 |
| 3 | 2 3 |
| 4 | 1 4 |
| 5 | 1 3 2 0 |

6 | 4

输出样例

1 | 0

样例1解释

没有任何路径的价值 ≥ 4 ，因此概率为 0。

样例2

输入样例

1 | 4
2 | 1 2
3 | 2 3
4 | 1 4
5 | 1 3 2 2
6 | 3

输出样例

1 | 148

样例2解释

考虑从每个点出发的路径：

- 1. 结点 1 价值 $\geq t$ 概率： $\frac{1}{4}$ （到 4）；
- 2. 结点 2 价值 $\geq t$ 概率： $\frac{1}{4}$ （到 2）；
- 3. 结点 3 价值 $\geq t$ 概率： $\frac{0}{4}$ ；
- 4. 结点 4 价值 $\geq t$ 概率： $\frac{1}{4}$ （到 1）。

总概率为 $\frac{37}{64}$ ，故答案为 148。

样例3

输入样例

1 | 5
2 | 5 3
3 | 2 1
4 | 4 2
5 | 1 3
6 | 2827309828 1072955396 267871571 670894092 2402751778
7 | 2218918831

输出样例

1 | 3053

样例4

见选手目录下的 `weight/weight4.in` 与 `weight/weight4.ans`。

这个样例满足测试点 5 的约束条件。

样例5

见选手目录下的 `weight/weight5.in` 与 `weight/weight5.ans`。

这个样例满足测试点 18 的约束条件。

数据范围

对于所有测试数据保证： $1 \leq n \leq 3 \times 10^5, 1 \leq u, v \leq n$ 。保证输入为一棵树，即满足无环、连通。
 $0 \leq w_i, t < 2^{32}$ 。

| 测试点 | $n \leq$ | 特殊性质 |
|---------|-----------------|------|
| 1 | 3 | 无 |
| 2, 3 | 10^2 | 无 |
| 4 | 10^3 | A |
| 5, 6 | 10^3 | 无 |
| 7, 8 | 5×10^4 | A |
| 9 ~ 11 | 5×10^4 | 无 |
| 12, 13 | 10^5 | A |
| 14 | 3×10^5 | A |
| 15 ~ 17 | 3×10^5 | B |
| 18 ~ 20 | 3×10^5 | 无 |

- 特殊性质 A ：保证对于任意 $1 \leq i < n$ ，满足 i 和 $i + 1$ 之间有边；
- 特殊性质 B ：保证对于任意 $1 < i \leq n$ ，满足 1 和 i 之间有边。

T4 周长

问题描述

有 n 位同学站在一个 $w \times h$ 的二维平面上，第 i 个同学在 (X_i, Y_i) 。

对于第 i 个同学，他可以选择进行以下 4 种操作之一：

1. 摧毁二维平面 $x < X_i$ 的部分；
2. 摧毁二维平面 $x > X_i$ 的部分；
3. 摧毁二维平面 $y < Y_i$ 的部分；
4. 摧毁二维平面 $y > Y_i$ 的部分。

请最大化所有同学完成操作后平面剩余部分的周长。

输入格式

从文件 `perimeter.in` 中读入数据。

第一行包括三个整数 n ， w 和 h 。

接下来的 n 行，每行两个整数 X_i 和 Y_i ，表示第 i 个同学的坐标。

输出格式

输出到文件 `perimeter.out` 中。

输出一个整数，表示所有同学完成操作后平面剩余部分的最大周长。

样例1

输入样例

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

输出样例

```
1 | 12
```

样例2

见选手目录下的 `perimeter/perimeter2.in` 与 `perimeter/perimeter2.ans`。

这个样例满足测试点 4 的约束条件。

样例3

见选手目录下的 `perimeter/perimeter3.in` 与 `perimeter/perimeter3.ans`。

这个样例满足测试点 13 的约束条件。

数据范围

对于所有测试数据保证： $1 \leq w, h \leq 10^8$, $1 \leq n \leq 3 \times 10^5$, $0 \leq X_i \leq w$, $0 \leq Y_i \leq h$ 。

| 测试点 | $n \leq$ | 特殊性质 |
|---------|-----------------|------|
| 1 | 500 | AB |
| 2 | 500 | A |
| 3 | 500 | B |
| 4, 5 | 500 | 无 |
| 6, 7 | 5000 | A |
| 8, 9 | 5000 | B |
| 10 ~ 12 | 5000 | 无 |
| 13 ~ 15 | 10^5 | 无 |
| 16, 17 | 3×10^5 | AB |
| 18 ~ 20 | 3×10^5 | A |
| 21 ~ 24 | 3×10^5 | B |
| 25 | 3×10^5 | 无 |

- 特殊性质 A : $1 \leq w, h \leq 100$;
- 特殊性质 B : 保证 $X_i < \frac{w}{2}$ 。