**一、目录树收缩显示**

在一个目录树中（假设都是目录），过深的目录路径不容易展示，为了提升用户体验，需要对目录进行收缩展示，求收缩后某一深度的目录个数。

如图所示，原目录树及收缩后示意：



**收缩规则：**

1、若某目录仅有一个子目录，则把这个子目录收缩到其父目录，展示为一个新目录。如图所示，i 收缩到 X 变成新的目录 X/i。

2、所有符合上述条件的均需收缩，收缩后符合上述条件的继续收缩，直到无法收缩。如图所示，目录B、C、E收缩为新的目录 B/C/E

现给定一个原目录树 orgTree，请按照收缩规则展示为一个新目录树，求新目录树中深度值为 depth（根节点深度为 1）的节点个数。

如图所示，收缩后深度为 2 的节点有 2 个（B/C/E、F），深度为 3 的节点有 4 个（N、M、H、X/i）。

**输入**

一个整数 num，表示父子节点对的数量，1 <= num <= 300

接下来 num 行表示 orgTree，每行一个父子节点对，格式为父节点 子节点，节点名称仅含字母或数字，长度 [1,10]

最后一行一个整数 depth，1 <= depth <= 300

树只有一个根，首个节点对的父节点为根。

树上各节点名称是全局唯一的。

每个节点下的子节点不超过10个。

**输出**

一个整数，表示收缩后深度为 depth 的节点个数

**样例 1：**

**输入：**

9

root B

root F

C E

B C

E N

F H

F X

E M

X i

3

**输出：**4

**提示：**输入数据表示的原目录树，及收缩示意如题面图示，深度为 3 的节点有 4 个（N、M、H、X/i）

**示例 2：**

**输入：**

3

1 B123456789

B123456789 c

1 b123456789

3

**输出：**0

**提示：**原目录树收缩后，不存在深度为 3 的节点，因此返回 0

**示例 3：**

**输入：**

4

A B

B C

C D

D E

1

**输出：**1

**提示：**收缩显示为一个节点 A/B/C/D/E，深度 1 的节点数为 1

**二、港口卸货**

智能港口需实现一个卸货机制：有若干艘都装有 6 个集装箱的货轮即将到达港口，所有货轮到达港口的时刻 t 记录于一维升序数组 time 中（非严格递增）。

港口有 num 个起重机，每个起重机每单位时间可完成一个集装箱的卸货。

若一艘货轮在时刻 t 到达港口：

1、如果在 t + limit 时刻前无法完成其所有集装箱的卸货，则该货轮无需卸货，直接进入驳船区等待人工卸货；

2、否则尽早完成该艘货轮的卸货；一旦开始卸货，所用的起重机不允许被抢占。

请问有多少艘货轮需进入驳船区等待人工卸货。

**注意：** 货轮仅在到达港口后才可卸货；多个起重机可同时对一个货轮进行卸货工作。

**示例 1：**

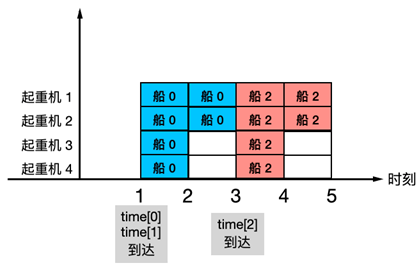
**输入：**

num = 4

time = [1,1,3]

limit = 2

**输出：**1

**解释：**如下图所示  


时刻 1 船 0 到，完成一艘货轮卸货的时刻为 3，未超过时间限制；

时刻 1 再到 船 1，完成这艘货轮的卸货需要到时刻 4，会超出 1+2 的时间限制，因此这一艘货轮需要进入驳船区；

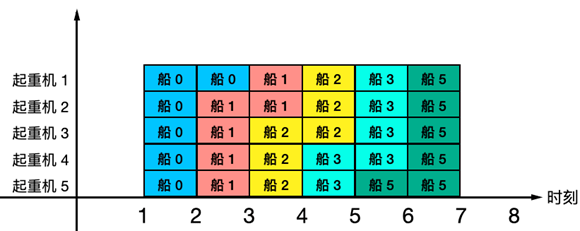
时刻 3 到达 船 2，完成这艘货轮卸货的时刻为 5，未超过时间限制；

因此共有 1 艘货轮需要进入驳船区，返回 1。

**示例 2：**

**输入：**  
num = 5  
time = [1,2,2,3,3,4,5,5]  
limit = 3

**输出：**3

**解释：**如下图所示，  


时刻 1 到达 1 艘货轮，完成 船0 卸货的时刻为 3；

时刻 2 到达 2 艘货轮，完成 船1 卸货的时刻为 4，完成 船2 卸货的时刻为 5；

时刻 3 到达 2 艘货轮，完成 船3 卸货的时刻为 6；船4 无法在规定时间内完成卸货，因此需要进入驳船区；

时刻 4 到达 1 艘货轮，完成 船5 卸货的时刻为 7；

时刻 5 到达 2 艘货轮，完成任意一艘卸货的时刻都为 9，超过时间限制；因此 2 艘货轮均需进入驳船区；

因此共有 3 艘货轮需要进入驳船区，返回 3。

**示例 3：**

**输入：**  
num = 20  
time = [3,3,5,5,5,5,5,5]  
limit = 2

**输出：**0

**提示：**

1 <= num <= 20

1 <= limit <= 100

1 <= time.length <= 1000

1 <= time[i] <= time[i+1] <= 10^4

非严格递增：对于数组arr，都有 arr[i] <= arr[i+1]。

**三、光通信激光穿透材料测试**

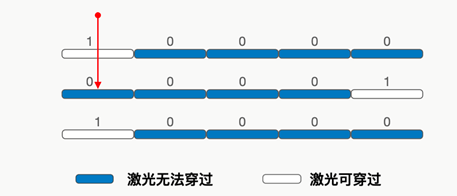
材料实验室研究员正在研究激光在新型透光玻璃中传输时的衰减效果。

给定多层上下叠放的介质，每层介质等长，并划分成若干个单位长度为 1 的格子。planks记录了从上到下每层的介质信息，planks[i] 表示第 i 层介质从左到右的透光玻璃放置情况，其中：

1 表示该位置有透光玻璃，可穿过激光；每层至少包含了一块透光玻璃；

0 表示该位置为不透光介质，无法穿过激光；

图中有三层介质，如下所示：



研究员从最上方、任意位置垂直向下发射激光，每次移动可以将任意一层介质左移（或右移）一个单位长度。

请问研究员最少需要移动多少次介质，可使激光能垂直穿过所有介质（即每层介质都需要被穿过）。

**示例 1：**

**输入：**planks = [[1,0,0,0,0],[0,0,0,0,1],[1,0,0,0,0]]

**输出：**4

**解释：**planks 的行表示层、列表示位置：从上往下依次为第 0 层、第 1 层、第 2 层，如上图所示。

**最佳方案为：**第 1 层介质向左移动 4 次。激光从位置 0 （即第 0 层未移动前的下标）向下发射，可穿过这三层介质。

**注：**将第 1 层介质向右移动 1 次的方案是不可行的，因为此时任意位置都无法穿过所有介质。

**示例 2：**

**输入：**planks = [[0,1,0,0,0],[1,0,0,0,0],[0,0,1,0,0],[0,0,0,1,1]]

**输出：**4

**解释：**

一种可行方案：

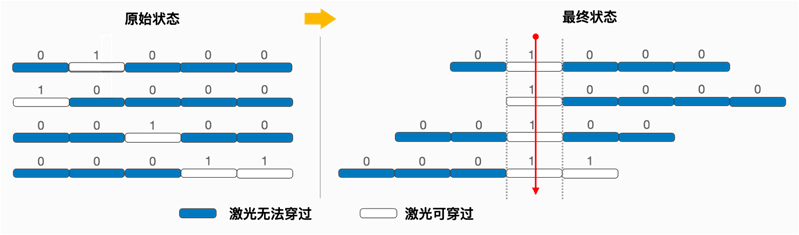
第 0 层介质右移 1 次；

第 1 层介质右移 2 次；

第 2 层介质不移动；

第 3 层介质左移 1 次；

一共移动 4 次，激光从位置 2 （即第 0 层未移动前的下标）向下发射，可穿过所有介质。



**示例 3：**

**输入：**planks = [[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1]]

**输出：**0

**提示：**

2 <= planks[i].length <= 1000

2 <= planks.length <= 1000

0 <= planks[i][j] <= 1

温馨提醒：纯暴力解法通过用例不多，请考虑高效的解法