**一、影像之结构化特征（搜索）**

**(graph.pas/c/cpp)**

**【问题描述】**

在影像对比中，有一种办法是利用影像中的边缘（edge）资讯，计算每个边缘资讯中具有代表性的结构化特征。以作为比对两张影像是否相似的判断标准。Water-filling方法是从每个边缘图的一个端点开始，绕着相连的边缘点走并依序编号。若走到某一步时，遇到一个以上不同的连接点，则分成不同路径同时继续走，知道没有任何连接点为止。如果一个点和另一个点为上下左右相邻，就称为连接。

|  |  |
| --- | --- |
|  |  |
|  |  |

例如，在图1的影象中包含三个边缘图，每个边缘图由一些互相连接的边缘点构成。途中以黑色的方块代表边缘点，白色的方块代表背景。在water-filling方法中，首先，从第一列（row）开始，由左至右，由上至下，先找到第一个黑色并编号为1。接下来，找1的下一个尚未编号的连接点并编号为2.依此方法继续往下一个点前进依次编号。在编号6的点之后有两个尚未编号的连接点，此时，则分为两条线路，并同时编号为7继续往下走。当走到没有任何的连接点时，则结束现有边缘图的编号，并继续对影象中的其他边缘图编号。走完图1所有边缘图后所得到的编号如图2所示。所以，走完这三个边缘图所需要的步数分别为12、7和3；所以12、7和3可以作为代表此张影象的结构化特征。请注意：位于斜对角上的两点不能算做连接，如

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  | 1 | 2 | 3 | 4 |  |  |  |  |
|  |  |  |  |  | 5 |  |  |  |  |
|  |  | 9 | 8 | 7 | 6 | 7 |  |  |  |
|  |  | 10 |  | 8 | 7 |  | 1 |  |  |
|  |  | 11 |  |  | 8 |  | 2 | 3 |  |
|  |  | 12 | 11 | 10 | 9 |  |  | 4 |  |
|  | 1 |  |  |  | 10 |  |  | 5 |  |
|  | 2 |  |  |  |  |  | 7 | 6 |  |
|  | 3 |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

图1 图2

请写一个程序计算每个影像中，以water-filling方法走完其中所有的边缘图后，将每个边缘图需走的步数依次走访的顺序列出。

**【输入格式】**

输入文件包含一个正方形的影像。每组影像以图的宽度n开头（1<=n<=1000）。接下来n的n行代表影像的内容：0表示背景的白点，1表示黑色的边缘点。

**【输出格式】**

以water-filling方法走完所有的边缘图后，先输出此张影像中共有几个边缘图。接着，将每个边缘图需走的步数按升序列出。

**【输入输出样例】**

|  |  |
| --- | --- |
| **graph.in** | **graph.out** |
| 10  0000000000  0011110000  0000010000  0011111000  0010110100  0010010110  0011110010  0100010010  0100000110  0100000000 | 3  3  7  12 |

**二、阅览室（模拟）**

**(reading.pas/c/cpp)**

**【问题描述】**

一个阅览室每天都要接待大批读者。阅览室开门时间是0，关门时间是T。每位读者的到达时间都不一样，并且想要阅读的刊物不超过5本。每位读者心里对自己想看的刊物都有一个排位，到达之后他会先去找自己最想看的刊物，如果找不到则去找其次想去看的刊物。如果找不到任何他想看的刊物，他会开始等待，直到有一本以上的他想看的刊物被人放回原处。当然，他会先去拿其中自己最想看的刊物。当他看完某一本刊物后，就把它放回原处，接着去找自己没看过的最想看的刊物。如此下去，直到看完所有他想看的刊物为止。矛盾出现在两个人同时想要拿同一本刊物的时候。阅览室为了避免读者之间出现争执，作了一个规定，读者每次在开始等待时先去服务台做一次登记。如果两个人都同时想要一本刊物，那么先登记的读者将得到这本刊物。如果两个人同时登记，那么先到达阅览室的读者将得到刊物。没得到的人就只能去找其他的刊物看。阅览室关门时，所有读者都将被强迫离开阅览室，不再允许继续阅读。

现在阅览室想做一个统计调查，你被要求写一个程序来模拟这个过程计算出所有刊物被阅读的总次数。当某个读者开始阅读某本刊物时，该刊物的被阅读次数就加1，无论这本刊物最后有没有被读完。

**【输入格式】**

测试数据开头是两个整数T（1<=T<=1000）和n（1<=n<=100），分别表示图书馆关门时间和读者总数。接下来按照读者的到达时间先后依次给出了每位读者的具体描述。每个读者描述开头是一个整数t（0<=t<T），表示该读者到达时间。接下来一行开头是一个整数k（1<=k<=5），表示该读者想要看的刊物数目。之后跟着2k个整数按照读者想要阅读的刊物的顺序依次给出了刊物的描述。其中第2i-1个整数表示刊物的编号s（0<=s<1000），第2i个整数表示该读者读完这本刊物所需的时间。

**【输出格式】**

输出所有刊物被阅读的总次数。

**【输入输出样例】**

|  |  |
| --- | --- |
| **reading.in** | **reading.out** |
| 10 4  1  2 1 4 2 5  3  1 2 4  7  3 2 2 1 3 3 2  9  1 4 2 | 5 |