

DP，搜索，网络流的强化，很有竞赛的味道

Translate:USACO/latin

All Latin Squares 拉丁正方形

描述

一种正方形的数字编排

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

是一个 5×5 的拉丁幻方，即每个 1 到 5 的整数在每行每列都出现且出现一次。

写个程序计算 $N \times N$ 的拉丁幻方的总数且要求第一行是：

```
1 2 3 4 5.....N
```

$2 \leq N \leq 7$

格式

PROGRAM NAME: latin

INPUT FORMAT

一行包含一个整数 N

OUTPUT FORMAT

只有一行，表示拉丁正方形的个数，且拉丁正方形的第一行为 1 2 3 . . . N.

SAMPLE INPUT (file latin.in)

```
5
```

SAMPLE OUTPUT (file latin.out)

解题思路:

题目要求计算出一个 $n \times n$ 的拉丁方所有的数目, 限制条件是第一行是 $1, 2, 3 \dots N$ 。单纯的 DFS 对于 7 以上的测试点很慢, 我们要考虑一些剪枝:

1. 假设一个 $n \times n$ 的拉丁方在固定第一行和第一列的情况下的构造数目是 $L[n]$, 那么解除第一列的固定后总数目是 $L[n] * (n-1)!$ 。理由是第二行到最后一行排列组合也可以重新组成新的拉丁方。这样也就不必要对第一列构造了, 令第一列也是 $1, 2 \dots N$ 即可, 也就是实质上我们的搜索是从 $[2, 2]$ 开始的。

2. 最后一行不用构造, 如果搜索完 $N-1$ 行, 到达第 N 行, 那么一定存在解。

3. 置换群优化, 如下:

ROW

1 12345

2 32154 ----> 存在两个置换 (1, 2, 3) 和 (4, 5)

1 12345

2 21453 ----> 存在两个置换 (1, 2) 和 (3, 4, 5)

上面两个置换最后得到的解的总数是一样的, 有了这个理论后, 我们就可以用个 Hash 来存储不同的置换群, 以后每次 DFS 时先找到前两行的置换是否有搜索过的, 如果有则直接累加就可以了。

Translate:USACO/tour

Canada Tour 周游加拿大

译 by Felicia Crazy

描述

你赢得了一场航空公司举办的比赛, 奖品是一张加拿大环游机票。旅行在这家航空公司开放的最西边的城市开始, 然后一直自西向东旅行, 直到你到达最东边的城市, 再由东向西返回, 直到你回到开始的城市。除了旅行开始的

城市之外，每个城市只能访问一次，因为开始的城市必定要被访问两次（在旅行的开始和结束）。

当然不允许使用其他公司的航线或者用其他的交通工具。

给出这个航空公司开放的城市的列表，和两两城市之间的直达航线列表。找出能够访问尽可能多的城市的路线，这条路线必须满足上述条件，也就是从列表中的第一个城市开始旅行，访问到列表中最后一个城市之后再返回第一个城市。

格式

PROGRAM NAME: tour

INPUT FORMAT

Line 1: 航空公司开放的城市数 N 和将要列出的直达航线的数量 V 。 N 是一个不大于 100 的正整数。 V 是任意的正整数。

Lines 2.. $N+1$: 每行包括一个航空公司开放的城市名称。城市名称按照自西向东排列。不会出现两个城市在同一条经线上的情况。每个城市的名称都是一个字符串，最多 15 字节，由拉丁字母表上的字母组成；城市名称中没有空格。

Lines $N+2$.. $N+2+V-1$: 每行包括两个城市名称（由上面列表中的城市名称组成），用一个空格分开。这样就表示两个城市之间的直达双程航线。

OUTPUT FORMAT

Line 1: 按照最佳路线访问的不同城市的数量 M 。如果无法找到路线，输出 1。

SAMPLE INPUT (file tour.in)

```
8 9
Vancouver
Yellowknife
Edmonton
Calgary
Winnipeg
Toronto
Montreal
Halifax
Vancouver Edmonton
Vancouver Calgary
Calgary Winnipeg
Winnipeg Toronto
Toronto Halifax
Montreal Halifax
Edmonton Montreal
Edmonton Yellowknife
Edmonton Calgary
```

SAMPLE OUTPUT (file tour.out)

7 (也就是: Vancouver, Edmonton, Montreal, Halifax, Toronto, Winnipeg, Calgary, 和 Vancouver (回到开始城市, 但是不算在不同城市之内)。)

题目解答:

问题的大意是找到一个回路使得从起点出发必须经过终点然后再次返回起点, 途中要求经过尽量多的城市, 每个城市只能访问一次。到此为止问题的模型是哈密顿回路, NP 难。不过问题又给出了到达终点和返回终点的方向是给定的。这也就是使得问题可以划分阶段, 去掉了后效性。我们可以用 DP 来解决这个问题。

状态转移: $DP[i][j] = \max\{DP[i][k] + 1\}$ (如果 k 和 j 两个城市连通, 同时 $DP[i][k]$ 非 0, 也就是存在可行的路径), 方程的意义是: 我们让两个路线 (往返) 同时从起点走, 从起点开始到达 i 和到达 j 共访问了多少个城市。最后输出的结果就是如果 i 和终点 n 存在路径连通那么 $DP[i][n]$ 中数字最大的那个 i 就是最终结果了。这个方程的时间复杂度是 $O(n^3)$ 。

边界条件是 $DP[1][1] = 1$ 。那么基本上问题就被解决了, 现在有个疑问是怎么避免两条路径出现交叉。首先有个性质 $DP[i][j] = DP[j][i]$ 。这样我们计算一边就可以了, 让 $j \geq i$ 。不可以飞向同一个点, 这也不符合每个城市访问一次的限制条件, 因此不能有等于号, 变成 $j > i$ 。因为方向是确定的, 自西向东走, 每个阶段都不会走到同一个城市上去也就是 $DP[i][i]$ 恒为 0, 也不能自己飞向自

已 $k \neq j$ 。这样 k 的取值应当是 $1 \leq k < j$ 同时 $k \neq i$ 。这样也就避免了两条路径交叉的情况。使得程序正常工作。

Translate:USACO/charrec

Character Recognition 字符识别

译 by Felicia Crazy

描述

这个问题需要你写一个程序完成字符识别的任务。

每个完整的字符图案有 20 行，20 位。每个位是“0”或“1”。图 1a 对应着输入文件中的符号图案。

文件 [font.in](#) 包括了 27 个字符图案的信息，以这样的顺序记录：

`_abcdefghijklmnopqrstuvwxyz`

其中 `_` 表示空格字符。每个完整字符长 20 行。

输入文件包含一个或多个可能损坏的字符图案。一个字符图案可能以这些方式被损坏。

- 最多有一行可能被复制了（就接在原来那一行的下面）
- 最多有一行可能丢失了
- 有些“0”可能被改成“1”
- 有些“1”可能被改成“0”

不会有任何一个字符图案既多余了一行并且又丢失了一行。在测试数据的任何一个字符图案中，“0”和“1”的被改变率不超过 30%。

被复制的那一行中，原来的行和多余的行可能都损坏了，而且损坏的部分可能并不相同。

写一个程序，使用 `font.in` 提供的字体，在输入文件提供的图象中识别出一个或多个的字符序列。

使用提供的字体图象来识别字符的时候，要找出最佳的多余行或遗漏行，使找出的所有“0”和“1”的变化数量最小。在所有可能的多余行中，只按损坏数据最少的那一行计算。你必须确定和输入序列最接近的字符序列（就是损坏数据最少的那一个）。每个测试数据有唯一的最优解。

正确的解答必须使用到输入文件中的所有数据。

格式

PROGRAM NAME: charrec

INPUT FORMAT (both input files)

两个输入文件都以一个整数 N 开始 ($19 \leq N < 1200$)，表示接下去的行数。

N
(位 1) (位 2) (位 3) ... (位 20)
(位 1) (位 2) (位 3) ... (位 20)
...

每行有 20 位的宽度。在 0 和 1 之间没有空格分开。

文件 `font.in` 描述字体。它共有 541 行。它在每个测试数据中可能不同。

OUTPUT FORMAT

你的程序必须建立一个输出文件，包含一个识别出来的字符串。它的格式是一个单行的 **ASCII** 文本。输出文件中不应该包含任何分隔符。如果你的程序没有识别出一个特定的字符，就必须在适当的位置输出一个“?”。(按照原文翻译，请忽略这句话，最后数据也没有要求输出“?”的)

SAMPLE INPUT

例: `font.in` 的开头 (不完全的), 有“空格”和“a” 例: `charrec.in` 显示了一个损坏的“a”

(——某郁闷的菜鸟: 其实 **Sample** 里面的 **font.in** 和上文给出的 **font.in** 是不同的...)

font.in

540

[illegible]

00000000000000000000
00000000000000000000
00000011100000000000
0000011111011000000
0000111111001100000
0000111000110010000
0000110000110001000
0000110000010001000
0000010000010001000
0000001000000011000
0000000100000111000
0000111111111111000
0000111111111111000
0000111111111100000
0000100000000000000
0000000000000000000
0000000000000000000
0000000000000000000
0000000000000000000

charrec.in

19

00000000000000000000
00000000000000000000
00000000000000000000
00000011100000000000
0010011101101100000
0000111111100110000
0000111000110010000
0000110000110001000
0000110000010001000
0000010000010001000
0000001000000011000
0000111101111111000
0000111111111111000
0000111111111100000
0000100001000000000
0000000000000000000
0000000000000100000
0000000000000000000
0000000000000000000

SAMPLE OUTPUT

a

解题报告：

这道题目分析起来很有难度，程序的核心模块实现起来也很繁杂。重点是要想清楚。下面分别从分析和程序实现两个部分来讨论这道题目。

分析：

首先本题目的题意大致是给你一个字体图集，然后从输入文件中去匹配最相似的字体并输出。但是有很多额外条件，例如多一行少一行，每行可能有些损坏如此等等。一个可行的方案是我们枚举所有可能的情况选择不同程度最小的那一种作为匹配，假设不存在多一行和少一行的情况，显然这就是一道统计题目。这样的话本题目就有了最优子结构，可以用 DP 求解。考虑一个方程 $DP[i]$ 表示匹配到第 i 行时的最小不同程度，那么状态转移方程就可以是 $DP[i] = \min\{DP[i-19]+COST[i-18][19], DP[i-20]+COST[i-19][20], DP[i-21]+COST[i-20][21]\}$ ，其中 $COST[i][j]$ 的含义是从 i 行开始连续匹配 j 行的最小情况，同时考虑了多一行和少一行的情况。

边界条件 $DP[0] = 0$ 。

程序实现：

读入时把字体中的字母分开，这样以后处理时会很方便。

$COST[i][j]$ 的计算方法，枚举所有字母：

对于 $j = 19$ ：假设 $T[k]$ 表示 1 到 20 行中去掉第 k 行 ($1 \leq k \leq 20$) 后的总区别值 $COST[i][j] = \min\{T[k]\}$ 。

对于 $j = 20$ ：直接枚举统计判断。

对于 $j = 21$ ：类似 19，但是注意多余的一行不需要加入统计中也就是枚举连续的 21 行，统计其中 20 行。

记录 $ANSL[i][j]$ 表示从第 i 行开始连续匹配 j 行后最优的匹配字母。

为了实现方便和加速程序，提前预处理一个数组 $dif[i][j][k]$ 表示字体 i 第 j 行与测试数据中第 k 行的不同程度。

注意要统计方案，一个是在计算 $COST$ 的时候，要知道对于 j 分别选择了哪个最优匹配的字符，另一个是求 DP 时，要记录是从上一个阶段的 3 个情况中的哪一个中继承来的。那么最佳方案就是记录在 $DP[n]$ 中的，这样每次选择的字母也就知道了。

Translate:USACO/betsy

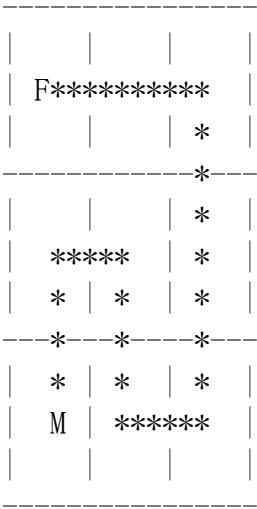
Betsy's Tour 漫游小镇

Don Piele

译 by Felicia Crazy

翻译

一个正方形的镇区分为 $N*N$ 个小方块 ($1 \leq N \leq 7$)。农场位于方格的左上角，集市位于左下角。贝茜穿过小镇，从左上角走到左下角，刚好经过每个方格一次。当 $N=3$ 时，贝茜的漫游路径可能如下图所示：



写一个程序，对于给出的 N 值，计算贝茜从农场走到集市有多少种唯一的路径。

格式

PROGRAM NAME: betsy

INPUT FORMAT

行 1: 一个整数 N ($1 \leq N \leq 7$)

OUTPUT FORMAT 只有一行。输出一个整数表示唯一路径的数量。

SAMPLE INPUT (file betsy.in)

本问题的考点也是剪枝，有以下两种情况：

如图所示，孤立区域也就是被划分成了两个域的情况，这个时候无论走哪个区域另一个区域都无法访问，那么一个剪枝就可以是判断当前有无孤立区域的形成，这个产生条件比较多，有一种基本情况就是如果当前点上下（包括边界）被访问过了，而左右没被访问则必产生孤立，另一个情况则正好相反。理由是当前点上下如果被访问了，而这两个访问过的点都是由起点走过来的，也就是都连通到起点，意义就是这两个点本身是连通的。我们不能穿过被访问过的路径走交叉路线，所以说这样无论走哪里都会被包住再也出不来了。这是一个情况。

我们考虑类似围棋中的气，每个格子不需要一进一出，也就是至少有两个度，如果只有一个度则走进进去后就再也出不来了，我们不必要碰到度为一的点再去回溯。而是一开始就避免度为一的点产生。剪枝方式是：除去起点和终点外如果当前点的周围未被访问的格子中有度为一的点（也就是连通到另外一个未被访问的格子），那么这个点必须要访问，理由是这个点如果被访问，就相当于他的一个入度被利用了，如果不访问则就浪费掉了。浪费之后他的度为 1，以后则会只进不出。因此必须要访问，所以可以推出另一个关于度的剪枝，如果存在两个以上的度为 1 的点则回溯，因为不能两个同时都选，选了一个，另一个必然是死胡同。

The figure consists of two 5x5 grids. In both, a yellow path starts at (1,1) and ends at (5,5). Obstacles are represented by red and yellow squares.

Left Grid:

●				

Path: (1,1) → (1,2) → (2,2) → (2,3) → (3,3) → (3,4) → (4,4) → (5,4) → (5,5)

Right Grid:

●				

Path: (1,1) → (1,2) → (1,3) → (1,4) → (1,5) → (2,5) → (3,5) → (3,4) → (3,3) → (3,2) → (3,1) → (4,1) → (4,2) → (4,3) → (4,4) → (4,5) → (5,5)

Translate:USACO/telecow

Telecommunication 奶牛的电信

译 by Maigo Akisame

描述

农夫约翰的奶牛们喜欢通过电邮保持联系，于是她们建立了一个奶牛电脑网络，以便互相交流。这些机器用如下的方式发送电邮：如果存在一个由 c 台电脑组成的序列 $a_1, a_2, \dots, a(c)$ ，且 a_1 与 a_2 相连， a_2 与 a_3 相连，等等，那么电脑 a_1 和 $a(c)$ 就可以互发电邮。

很不幸，有时候奶牛会不小心踩到电脑上，农夫约翰的车也可能碾过电脑，这台倒霉的电脑就会坏掉。这意味着这台电脑不能再发送电邮了，于是与这台电脑相关的连接也就不可用了。

有两头奶牛就想：如果我们两个不能互发电邮，至少需要坏掉多少台电脑呢？
请编写一个程序为她们计算这个最小值和与之对应的坏掉的电脑集合。

以如下网络为例：

$$\begin{array}{c} 1* \\ / \\ 3 - 2* \end{array}$$

这张图画的是有 2 条连接的 3 台电脑。我们想要在电脑 1 和 2 之间传送信息。电脑 1 与 3、2 与 3 直接连通。如果电脑 3 坏了，电脑 1 与 2 便不能互发信息了。

格式

PROGRAM NAME: telecow

INPUT FORMAT 第一行四个由空格分隔的整数：N,M,c1,c2.N 是电脑总数 ($1 \leq N \leq 100$)，电脑由 1 到 N 编号。M 是电脑之间连接的总数 ($1 \leq M \leq 600$)。最后的两个整数 c1 和 c2 是上述两头奶牛使用的电脑编号。连接没有重复且均为双向的(即如果 c1 与 c2 相连，那么 c2 与 c1 也相连)。两台电脑之间至多有一条连接。电脑 c1 和 c2 不会直接相连。第 2 到 M+1 行 接下来的 M 行中，每行包含两台直接相连的电脑的编号。

OUTPUT FORMAT

输出共有两行。第一行是使电脑 c1 和 c2 不能互相通信需要坏掉的电脑数目的最小值。第二行是排好序的坏掉的电脑的编号列表。注意 c1 和 c2 都不能坏掉。如果有多种可能情况，输出第一个数最小的一种，如果第一个数相同，则输出第二个数最小的一种，依此类推。

样例输入(文件 telecow.in)

```
3 2 1 2
1 3
2 3
```

样例输出(文件 telecow.out)

1
3

解题报告：

本题目的大意是截断几个点之后从源到汇不连通，这就可以转化为一个网络流问题，求最小割方案。因为最大流等于最小割，我们建模的时候考虑到点的流量是 1，这样最大流就恰好是最少需要的点。为了方便处理问题我们可以拆点来做，把点 x 拆成 x_1 和 x_2 ，然后 x_1 到 x_2 的容量是 1。然后如果 x 和 y 相连，那么我们就建立有 x_2 到 y_1 ， y_2 到 x_1 两个容量无穷的边，这样，求出最大流第一问就解决了。

具体的方案可以枚举每一个点，删掉后查看流量的变化，如果减少 1 那么就加入输出队列，并永久删除这个点，直到流量为 0 时，最小割方案就有了。