

搜索：**DFS**、**BFS**、**A\***。题目很多，有简单的，当然也有难得不会做的。代码量似乎都很大。

## 第 26 题 POJ 1324 Holedox Moving

### Holedox Moving

#### Description

During winter, the most hungry and severe time, Holedox sleeps in its lair. When spring comes, Holedox wakes up, moves to the exit of its lair, comes out, and begins its new life.

Holedox is a special snake, but its body is not very long. Its lair is like a maze and can be imagined as a rectangle with  $n \times m$  squares. Each square is either a stone or a vacant place, and only vacant places allow Holedox to move in. Using ordered pair of row and column number of the lair, the square of exit located at (1,1). Holedox's body, whose length is  $L$ , can be represented block by block. And let  $B_1(r_1, c_1)$   $B_2(r_2, c_2)$  ..  $B_L(r_L, c_L)$  denote its  $L$  length body, where  $B_i$  is adjacent to  $B_{i+1}$  in the lair for  $1 \leq i \leq L-1$ , and  $B_1$  is its head,  $B_L$  is its tail.

To move in the lair, Holedox chooses an adjacent vacant square of its head, which is neither a stone nor occupied by its body. Then it moves the head into the vacant square, and at the same time, each other block of its body is moved into the square occupied by the corresponding previous block.

For example, in the Figure 2, at the beginning the body of Holedox can be represented as  $B_1(4,1)$   $B_2(4,2)$   $B_3(3,2)$   $B_4(3,1)$ . During the next step, observing that  $B_1'(5,1)$  is the only square that the head can be moved into, Holedox moves its head into  $B_1'(5,1)$ , then moves  $B_2$  into  $B_1$ ,  $B_3$  into  $B_2$ , and  $B_4$  into  $B_3$ . Thus after one step, the body of Holedox locates in  $B_1(5,1)$   $B_2(4,1)$   $B_3(4,2)$   $B_4(3,2)$  (see the **Figure 3**).

Given the map of the lair and the original location of each block of Holedox's body, your task is to write a program to tell the minimal number of steps that Holedox has to take to move its head to reach the square of exit (1,1).

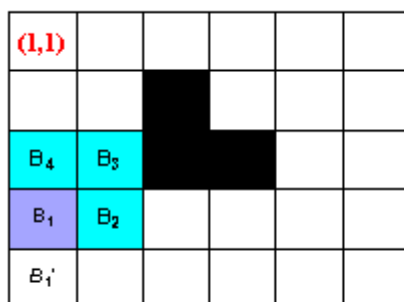


Figure 2 Sample Input

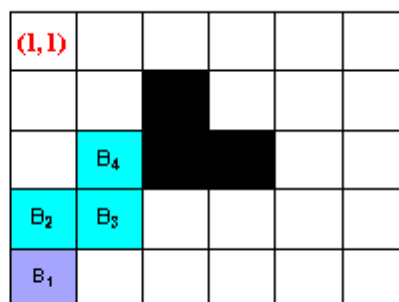


Figure 3 After one step from figure 2

## Input

The input consists of several test cases. The first line of each case contains three integers  $n$ ,  $m$  ( $1 \leq n, m \leq 20$ ) and  $L$  ( $2 \leq L \leq 8$ ), representing the number of rows in the lair, the number of columns in the lair and the body length of Holedox, respectively. The next  $L$  lines contain a pair of row and column number each, indicating the original position of each block of Holedox's body, from  $B1(r1, c1)$  to  $BL(rL, cL)$  orderly, where  $1 \leq ri \leq n$ , and  $1 \leq ci \leq m, 1 \leq i \leq L$ . The next line contains an integer  $K$ , representing the number of squares of stones in the lair. The following  $K$  lines contain a pair of row and column number each, indicating the location of each square of stone. Then a blank line follows to separate the cases.

The input is terminated by a line with three zeros.

Note:  $B_i$  is always adjacent to  $B_{i+1}$  ( $1 \leq i \leq L-1$ ) and exit square  $(1,1)$  will never be a stone.

## Output

For each test case output one line containing the test case number followed by the minimal number of steps Holedox has to take. "-1" means no solution for that case.

## Sample Input

```
5 6 4
4 1
4 2
3 2
3 1
3
2 3
3 3
3 4
```

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

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

```
0 0 0
```

## Sample Output

```
Case 1: 9
```

```
Case 2: -1
```

题意：

在  $n*m$  的地图上，给出长度为  $L$  的蛇身体各个节位置，以及有  $k$  个点 is 墙，问蛇从初始位置走到  $(1, 1)$  点的最小步数。蛇不能撞墙，不能撞自己的身体。

提示：A\*，广搜（深搜）+ 状态压缩

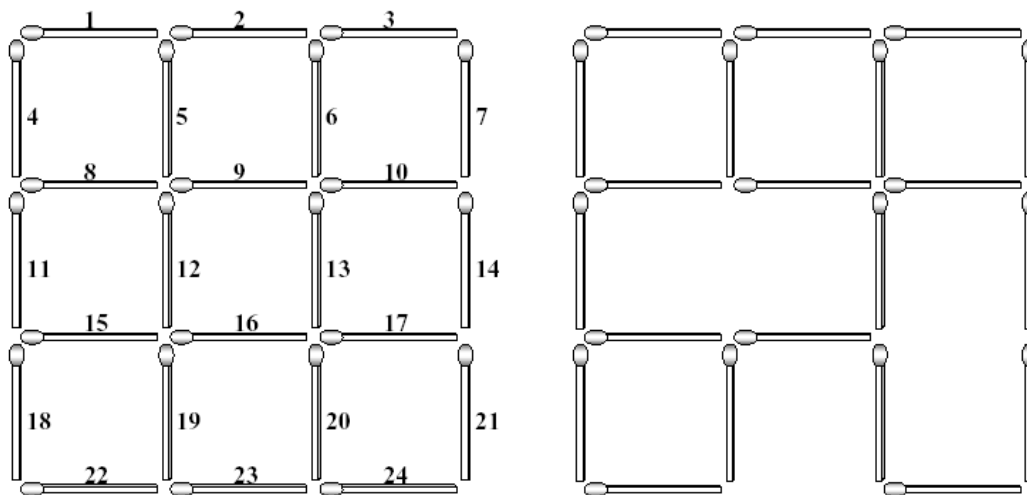
## 第 27 题 POJ 1084 Square Destroyer

### Square Destroyer

#### Description

The left figure below shows a complete 3\*3 grid made with  $2 \times (3 \times 4) (=24)$  matchsticks. The lengths of all matchsticks are one. You can find many squares of different sizes in the grid. The size of a square is the length of its side. In the grid shown in the left figure, there are 9 squares of size one, 4 squares of size two, and 1 square of size three.

Each matchstick of the complete grid is identified with a unique number which is assigned from left to right and from top to bottom as shown in the left figure. If you take some matchsticks out from the complete grid, then some squares in the grid will be destroyed, which results in an incomplete 3\*3 grid. The right figure illustrates an incomplete 3\*3 grid after removing three matchsticks numbered with 12, 17 and 23. This removal destroys 5 squares of size one, 3 squares of size two, and 1 square of size three. Consequently, the incomplete grid does not have squares of size three, but still has 4 squares of size one and 1 square of size two.



As input, you are given a (complete or incomplete)  $n \times n$  grid made with no more than  $2n(n+1)$  matchsticks for a natural number  $5 \leq n$ . Your task is to compute the minimum number of matchsticks taken out to destroy all the squares existing in the input  $n \times n$  grid.

## Input

The input consists of  $T$  test cases. The number of test cases ( $T$ ) is given in the first line of the input file.

Each test case consists of two lines: The first line contains a natural number  $n$ , not greater than 5, which implies you are given a (complete or incomplete)  $n \times n$  grid as input, and the second line begins with a nonnegative integer  $k$ , the number of matchsticks that are missing from the complete  $n \times n$  grid, followed by  $k$  numbers specifying the matchsticks. Note that if  $k$  is equal to zero, then the input grid is a complete  $n \times n$  grid; otherwise, the input grid is an incomplete  $n \times n$  grid such that the specified  $k$  matchsticks are missing from the complete  $n \times n$  grid.

## Output

Print exactly one line for each test case. The line should contain the minimum number of matchsticks that have to be taken out to destroy all the squares in the input grid.

## Sample Input

```
2
2
0
3
3 12 17 23
```

## Sample Output

```
3
3
```

题意：给一个由  $(n+1) \times n \times 2$  根火柴棒构成的边长为  $n$  正方形，每根火柴棒都有编号，已经删去了  $k$  根火柴棒，问至少再删多少根火柴棒可以破坏掉所有正方形。

## 第 28 题 POJ 2449 Remmarguts' Date

### Remmarguts' Date

#### Description

"Good man never makes girls wait or breaks an appointment!" said the mandarin duck father. Softly touching his little ducks' head, he told them a story.

"Prince Remmarguts lives in his kingdom UDF – United Delta of Freedom. One day their neighboring country sent them Princess Uyuw on a diplomatic mission."

"Erenow, the princess sent Remmarguts a letter, informing him that she would come to the hall and hold commercial talks with UDF if and only if the prince go and meet her via the K-th shortest path. (in fact, Uyuw does not want to come at all)" Being interested in the trade development and such a lovely girl, Prince Remmarguts really became enamored. He needs you - the prime minister's help!

DETAILS: UDF's capital consists of N stations. The hall is numbered S, while the station numbered T denotes prince' current place. M muddy directed sideways connect some of the stations. Remmarguts' path to welcome the princess might include the same station twice or more than twice, even it is the station with number S or T. Different paths with same length will be considered disparate.

#### Input

The first line contains two integer numbers N and M ( $1 \leq N \leq 1000$ ,  $0 \leq M \leq 100000$ ). Stations are numbered from 1 to N. Each of the following M lines contains three integer numbers A, B and T ( $1 \leq A, B \leq N$ ,  $1 \leq T \leq 100$ ). It shows that there is a directed sideways from A-th station to B-th station with time T.

The last line consists of three integer numbers S, T and K ( $1 \leq S, T \leq N$ ,  $1 \leq K \leq 1000$ ).

#### Output

A single line consisting of a single integer number: the length (time required) to welcome Princess Uyuw using the K-th shortest path. If K-th shortest path does not exist, you should output "-1" (without quotes) instead.

#### Sample Input

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

#### Sample Output

```
14
```

题目大意：给出一个图，然后给出一个起点一个终点，求这两点间的第  $K$  短路。

提示：可以走重复的路的，如果图中有环的话，无论求第几短路都是存在的。

## 第 29 题 POJ 1475 Pushing Boxes

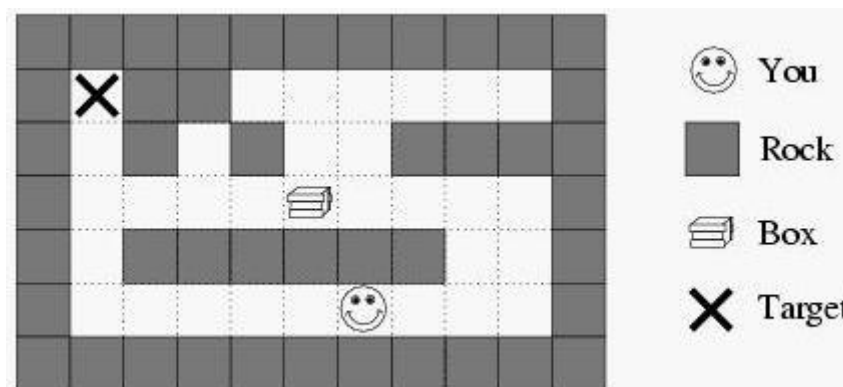
### Pushing Boxes

#### Description

Imagine you are standing inside a two-dimensional maze composed of square cells which may or may not be filled with rock. You can move north, south, east or west one cell at a step. These moves are called walks.

One of the empty cells contains a box which can be moved to an adjacent free cell by standing next to the box and then moving in the direction of the box. Such a move is called a push. The box cannot be moved in any other way than by pushing, which means that if you push it into a corner you can never get it out of the corner again.

One of the empty cells is marked as the target cell. Your job is to bring the box to the target cell by a sequence of walks and pushes. As the box is very heavy, you would like to minimize the number of pushes. Can you write a program that will work out the best such sequence?



#### Input

The input contains the descriptions of several mazes. Each maze description starts with a line containing two integers  $r$  and  $c$  (both  $\leq 20$ ) representing the number of rows and columns of the maze.

Following this are  $r$  lines each containing  $c$  characters. Each character describes one cell of the maze. A cell full of rock is indicated by a '#' and an empty cell is represented by a '.'. Your starting position is symbolized by 'S', the starting position of the box by 'B' and the target cell by 'T'.

Input is terminated by two zeroes for  $r$  and  $c$ .



## Output

For each maze in the input, first print the number of the maze, as shown in the sample output. Then, if it is impossible to bring the box to the target cell, print ``Impossible.".

Otherwise, output a sequence that minimizes the number of pushes. If there is more than one such sequence, choose the one that minimizes the number of total moves (walks and pushes). If there is still more than one such sequence, any one is acceptable.

Print the sequence as a string of the characters N, S, E, W, n, s, e and w where uppercase letters stand for pushes, lowercase letters stand for walks and the different letters stand for the directions north, south, east and west.

Output a single blank line after each test case.

## Sample Input

```
1 7
SB....T
1 7
SB..#.T
7 11
#####
#T##.....#
#.#.#..####
#....B....#
#.######..#
#.....S...#
#####
8 4
....
.##.
.#..
.#..
.#.B
.##S
....
###T
0 0
```

## Sample Output

Maze #1

EEEEEE

Maze #2

Impossible.

Maze #3

eennwWwWWWeeeeeesswwwwwnNN

Maze #4

swwwnnnnnnneeeessSSS

假设只有一个箱子。游戏在一个 R 行 C 列的由单位格子组成的区域中进行，每一步，可以移动到相邻的四个格子中的一个，前提是那个格子是空的；或者，如果你在箱子旁边，你也可以推动箱子前进一格，当然不能推到区域外面。初始时你在其中某个格子内，你要把箱子推到指定格子。又由于箱子很重，所以你要用尽量少的推动次数。

## 第 30 题 POJ 1077 Eight

### Eight

#### Description

The 15-puzzle has been around for over 100 years; even if you don't know it by that name, you've seen it. It is constructed with 15 sliding tiles, each with a number from 1 to 15 on it, and all packed into a 4 by 4 frame with one tile missing. Let's call the missing tile 'x'; the object of the puzzle is to arrange the tiles so that they are ordered as:

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 x

where the only legal operation is to exchange 'x' with one of the tiles with which it shares an edge. As an example, the following sequence of moves solves a slightly scrambled puzzle:

1 2 3 4    1 2 3 4    1 2 3 4    1 2 3 4

5 6 7 8    5 6 7 8    5 6 7 8    5 6 7 8

9 x 10 12    9 10 x 12    9 10 11 12    9 10 11 12

13 14 11 15    13 14 11 15    13 14 x 15    13 14 15 x

r->

d->

r->

The letters in the previous row indicate which neighbor of the 'x' tile is swapped with the 'x' tile at each step; legal values are 'r','l','u' and 'd', for right, left, up, and down, respectively.

Not all puzzles can be solved; in 1870, a man named Sam Loyd was famous for distributing an unsolvable version of the puzzle, and frustrating many people. In fact, all you have to do to make a regular puzzle into an unsolvable one is to swap two tiles (not counting the missing 'x' tile, of course).

In this problem, you will write a program for solving the less well-known 8-puzzle, composed of tiles on a three by three arrangement.

## Input

You will receive a description of a configuration of the 8 puzzle. The description is just a list of the tiles in their initial positions, with the rows listed from top to bottom, and the tiles listed from left to right within a row, where the tiles are represented by numbers 1 to 8, plus 'x'. For example, this puzzle

1 2 3

x 4 6

7 5 8

is described by this list:

1 2 3 x 4 6 7 5 8

## Output

You will print to standard output either the word ``unsolvable'', if the puzzle has no solution, or a string consisting entirely of the letters 'r', 'l', 'u' and 'd' that describes a series of moves that produce a solution. The string should include no spaces and start at the beginning of the line.

## Sample Input

2 3 4 1 5 x 7 6 8

## Sample Output

ullddrurdllurdruldr

题目大意：一个 3\*3 的棋盘上有 8 个格子写着 1~8 的数字，还有一个空格，给定局面，要求给出一个空格的移动序列，使得达到目标状态：

1 2 3

4 5 6

7 8 x（空格）

提示：刚学过 A\*

## 选做 POJ 1184 聪明的打字员

### 聪明的打字员

#### Description

阿兰是某机密部门的打字员，她现在接到一个任务：需要在一天之内输入几百个长度固定为 6 的密码。当然，她希望输入的过程中敲击键盘的总次数越少越好。

不幸的是，出于保密的需要，该部门用于输入密码的键盘是特殊设计的，键盘上没有数字键，而只有以下六个键：Swap0, Swap1, Up, Down, Left, Right，为了说明这 6 个键的作用，我们先定义录入区的 6 个位置的编号，从左至右依次为 1, 2, 3, 4, 5, 6。下面列出每个键的作用：

**Swap0:** 按 Swap0，光标位置不变，将光标所在位置的数字与录入区的 1 号位置的数字（左起第一个数字）交换。如果光标已经处在录入区的 1 号位置，则按 Swap0 键之后，录入区的数字不变；

**Swap1:** 按 Swap1，光标位置不变，将光标所在位置的数字与录入区的 6 号位置的数字（左起第六个数字）交换。如果光标已经处在录入区的 6 号位置，则按 Swap1 键之后，录入区的数字不变；

**Up:** 按 Up，光标位置不变，将光标所在位置的数字加 1（除非该数字是 9）。例如，如果光标所在位置的数字为 2，按 Up 之后，该处的数字变为 3；如果该处数字为 9，则按 Up 之后，数字不变，光标位置也不变；

**Down:** 按 Down，光标位置不变，将光标所在位置的数字减 1（除非该数字是 0），如果该处数字为 0，则按 Down 之后，数字不变，光标位置也不变；

**Left:** 按 Left，光标左移一个位置，如果光标已经在录入区的 1 号位置（左起第一个位置）上，则光标不动；

**Right:** 按 Right，光标右移一个位置，如果光标已经在录入区的 6 号位置（左起第六个位置）上，则光标不动。

当然，为了使这样的键盘发挥作用，每次录入密码之前，录入区总会随机出现一个长度为 6 的初始密码，而且光标固定出现在 1 号位置上。当巧妙地使用上述六个特殊键之后，可以得到目标密码，这时光标允许停在任何一个位置。现在，阿兰需要你的帮助，编写一个程序，求出录入一个密码需要的最少的击键次数。

#### Input

仅一行，含有两个长度为 6 的数，前者为初始密码，后者为目标密码，两个密码之间用一个空格隔开。

#### Output

仅一行，含有一个正整数，为最少需要的击键次数。

## Sample Input

123456 654321

## Sample Output

11

提示：双向搜索