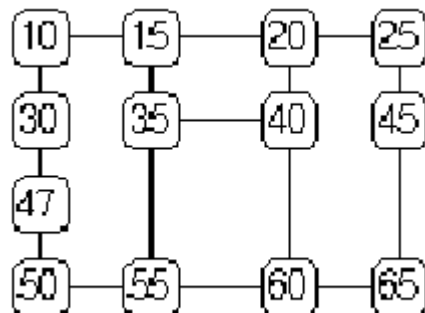


A Node Too Far – UVA 336

Para evitar o problema potencial de mensagens de rede (pacotes) circulando para sempre dentro de uma rede, cada mensagem inclui um campo de tempo de vida – *Time To Live* (TTL). Este campo contém o número de nós (estações, computadores, etc) que podem transmitir a mensagem, encaminhando-a rumo ao seu destino, antes que a mensagem seja descartada sem cerimônia. Cada vez que uma estação recebe uma mensagem, ela decrementa o campo TTL em uma unidade. Se o destino da mensagem é a estação corrente, então o valor do campo TTL é ignorado. Entretanto, se a mensagem deve ser encaminhada, e o campo TTL decrementado contém zero, então a mensagem não é encaminhada.

Neste problema você terá a descrição de várias redes, e para cada rede pede-se para determinar o número de nós que não são alcançáveis dado um nó inicial e um valor para o campo TTL. Considere a seguinte rede como exemplo:



Se uma mensagem com campo TTL igual a 2 foi enviada do nó 35 ela pode alcançar os nós 15, 10, 55, 50, 40, 20 e 60. Ela não pode alcançar os nós 30, 47, 25, 45 ou 65, visto que o campo TTL seria ajustado para zero quando a mensagem chegasse nos nós 10, 20, 50 e 60. Se aumentamos o valor inicial do campo TTL para 3, começando do nó 35 uma mensagem poderia alcançar todos os nós, menos o 45.

Especificação de Entrada e Saída

Haverá múltiplas configurações de rede fornecidas na entrada. Cada descrição de rede começa com um inteiro NC especificando o número de conexões entre nós da rede. Um valor de NC igual a zero marca o fim dos dados de entrada. Após NC , haverá NC pares de inteiros positivos. Estes pares identificam os nós que estão conectados por uma linha de comunicação. Não haverá mais do que uma linha de comunicação (direta) entre qualquer par de nós, e nenhuma rede irá conter mais de 30 nós. Após cada configuração de rede, haverá várias consultas quanto ao número de nós que não são acessíveis dado um nó inicial e a configuração do campo TTL. Estas consultas são dadas como um par de números inteiros, o primeiro identifica o nó de partida e o segundo a configuração de campo TTL inicial. As consultas são finalizadas por um par de zeros.

Para cada consulta, exiba uma única linha mostrando o número do caso de teste (numerados sequencialmente a partir de um), o número de nós não alcançável, o número do nó de partida, e a configuração inicial do campo TTL. O exemplo de entrada e saída abaixo ilustram o formato.

Exemplo de Entrada

16
10 15 15 20 20 25 10 30 30 47 47 50 25 45 45 65
15 35 35 55 20 40 50 55 35 40 55 60 40 60 60 65
35 2 35 3 0 0

14
1 2 2 7 1 3 3 4 3 5 5 10 5 11
4 6 7 6 7 8 7 9 8 9 8 6 6 11
1 1 1 2 3 2 3 3 0 0

0

Exemplo de Saída

Case 1: 5 nodes not reachable from node 35 with TTL = 2.
Case 2: 1 nodes not reachable from node 35 with TTL = 3.
Case 3: 8 nodes not reachable from node 1 with TTL = 1.
Case 4: 5 nodes not reachable from node 1 with TTL = 2.
Case 5: 3 nodes not reachable from node 3 with TTL = 2.
Case 6: 1 nodes not reachable from node 3 with TTL = 3.

Knight Moves – UVA 439

Um amigo seu esta pesquisando o *Problema do Cavaleiro Viajante* onde você deve encontrar o menor percurso fechado de movimentos de cavalo que visitam cada casa, de um conjunto de n casas, em um tabuleiro de xadrez exatamente uma única vez. Ele acha que a parte mais difícil do problema é determinar o menor número de movimentos de cavado entre duas casas dadas e que, uma vez que você tenha feito isto, encontrar o percurso seria fácil.

É claro que você sabe que é vice-versa. Então você se oferece para escrever um programa que resolva a parte "difícil".

Seu trabalho é escrever um programa que receba duas casas a e b como entrada e então determine o número de movimentos de cavalo em uma rota mais curta de a para b .

Especificação de Entrada

O arquivo de entrada conterá um ou mais casos de teste. Cada caso de teste consiste de uma linha contendo duas casas separadas por um espaço. Uma casa é uma "string" consistindo de uma letra (a-h) representando a coluna e um dígito (1-8) representando a linha do tabuleiro de xadrez.

Especificação de Saída

Para cada caso de teste, escreva uma linha dizendo: "To get from xx to yy takes n knight moves."

Exemplo de Entrada

```
e2 e4
a1 b2
b2 c3
a1 h8
a1 h7
h8 a1
b1 c3
f6 f6
```

Exemplo de Saída

```
To get from e2 to e4 takes 2 knight moves.
To get from a1 to b2 takes 4 knight moves.
To get from b2 to c3 takes 2 knight moves.
To get from a1 to h8 takes 6 knight moves.
To get from a1 to h7 takes 5 knight moves.
To get from h8 to a1 takes 6 knight moves.
To get from b1 to c3 takes 1 knight moves.
To get from f6 to f6 takes 0 knight moves.
```

Dungeon Master – UVA 532

Você está preso em um calabouço 3D e precisa encontrar o caminho mais rápido para fora! O calabouço é composto de cubos unitários que podem ou não estar cheios de rocha. Demora um minuto para mover uma unidade norte, sul, leste, oeste, para cima ou para baixo. Você não pode se mover na diagonal e o labirinto está rodeado por rocha sólida em todos os lados.

É possível uma fuga? Se sim, quanto tempo vai demorar?

Especificação de Entrada

O arquivo de entrada consiste de um número de calabouços. Cada calabouço começa com uma linha contendo três inteiros L , R e C (todos limitados ao tamanho 30). L é o número de níveis que formam o calabouço. R e C são o número de linhas e colunas que formam o plano de cada nível.

Então segue-se L blocos de R linhas cada contendo C caracteres. Cada caractere descreve uma célula do calabouço. Uma célula cheia de rocha é indicada por um '#' e uma célula vazia é representada por um '.'. Sua posição inicial é indicada por 'S' e a saída pela letra 'E'. Existe uma única linha em branco após cada nível. Entrada é terminada por três zeros para L , R e C .

Especificação de Saída

Cada labirinto gera uma linha de saída. Se é possível alcançar a saída, imprima uma linha da forma: Escaped in x minute(s).

onde x é substituída por o menor tempo levado para escapar.

Se não é possível escapar, imprima a linha:

Trapped!

Exemplo de Entrada

```
3 4 5
S....
.###.
.##..
###.#
```

```
#####
#####
##.##
##...
```

```
#####
#####
#.###
####E
```

```
1 3 3
S##
#E#
###
```

```
0 0 0
```

Exemplo de Saída

```
Escaped in 11 minute(s).
Trapped!
```