

# Pseudocódigo e explicação

## Explicação Geral do Problema

O problema UVA - 705 - *Slash Maze*, apresenta um labirinto desenhado com os caracteres '/' e '\'. Cada caractere forma uma parede diagonal que, quando expandida, cria caminhos e barreiras. O desafio é identificar quantos ciclos fechados existem no labirinto e determinar o tamanho do maior desses ciclos.

Para lidar com as diagonais, uma técnica comum é expandir a matriz original para uma grade com resolução três vezes maior. Dessa forma, cada caractere do labirinto é convertido em uma submatriz 3×3 onde determinadas posições são marcadas como paredes. Com essa representação, o problema se transforma em percorrer uma grade ortogonal (movendo-se para cima, baixo, esquerda e direita) e identificar regiões conectadas de células livres que não tocam a borda da grade. Estas regiões correspondem aos ciclos fechados.

## Explicação da Solução

A solução proposta utiliza os seguintes passos:

### 1. Leitura do Input:

O programa lê os casos de teste a partir do arquivo **input.txt**. Cada caso inicia com dois inteiros que representam a largura (w) e a altura (h) do labirinto. Em seguida, lê-se h linhas que compõem o labirinto. A leitura encerra quando ambos w e h são zero.

### 2. Expansão do Labirinto:

Para facilitar a detecção dos ciclos, o labirinto é expandido para uma grade 3 vezes maior em cada dimensão. Cada caractere '/' ou '\' é transformado em uma submatriz 3×3, onde:

- Para '/', as posições correspondentes (ex.: [0,2], [1,1] e [2,0]) são marcadas como parede.

- Para '\', as posições correspondentes (ex.: [0,0], [1,1] e [2,2]) são marcadas como parede.

Dessa forma, o problema passa a ser o de percorrer essa matriz expandida, tratando células com valor 0 como espaços livres e com valor 1 como paredes.

### 3. Busca em Profundidade (DFS):

A solução percorre a matriz expandida procurando por células livres que ainda não foram visitadas. Quando encontra uma nova célula livre, inicia-se uma busca em profundidade (DFS) para explorar toda a região conectada. Durante a exploração, verifica-se se a região toca a borda da grade. Se tocar, a região não é considerada um ciclo fechado. Caso contrário, a região é um ciclo e seu tamanho é calculado dividindo-se o número de células livres por 3 (devido à expansão).

### 4. Contagem e Impressão dos Resultados:

Após explorar todas as regiões, o programa conta quantos ciclos foram encontrados e determina o tamanho do maior ciclo. Os resultados são, então, formatados conforme o padrão do problema e exibidos.

## Pseudocódigo

```
Ler entrada do arquivo input.txt
Enquanto houver casos de teste:
    Ler w e h (largura e altura)
    Se w == 0 e h == 0, encerrar
    Para cada uma das h linhas:
        Ler a linha do labirinto (sequência de '/' e '\')
    Expandir o labirinto para uma grade 3×3:
        Para cada caractere na posição (i, j):
            Se o caractere for '/':
                Marcar as células (3*i + 0, 3*j + 2), (3*i + 1, 3*j + 1) e (3*i + 2, 3*j + 0) como parede
            Se o caractere for '\':
                Marcar as células (3*i + 0, 3*j + 0), (3*i + 1, 3*j + 1) e (3*i + 2, 3*j + 2) como parede
    Inicializar contador de ciclos e variável para o maior ciclo
```

Para cada célula da grade expandida:

- Se a célula for livre e não tiver sido visitada:
  - Iniciar DFS para explorar a região conectada:
    - Marcar célula como visitada
    - Incrementar contador de células da região
    - Se a célula estiver na borda, marcar região como não ciclo
    - Visitar todas as células vizinhas (movimentos: cima, baixo, esquerda e direita)
  - Se a região não tocou a borda:
    - Incrementar o contador de ciclos
    - Calcular o tamanho do ciclo como (número de células livres da região) / 3
    - Atualizar o maior ciclo, se necessário

Imprimir os resultados formatados para o caso de teste

## Referências

- <https://github.com/morris821028/UVa/blob/master/volume007/705%20-%20Slash%20Maze.cpp>
- [https://github.com/yubinbai/pcuva-problems/tree/master/UVa 705 slash maze](https://github.com/yubinbai/pcuva-problems/tree/master/UVa%20705%20slash%20maze)
- <https://codeforces.com/blog/entry/7466?f0a28=2>
- <https://www.cnblogs.com/NaCl/p/4787514.html>