

Algoritmos e Estruturas de Dados 2

2024/1 - Trabalho 2

Enunciado

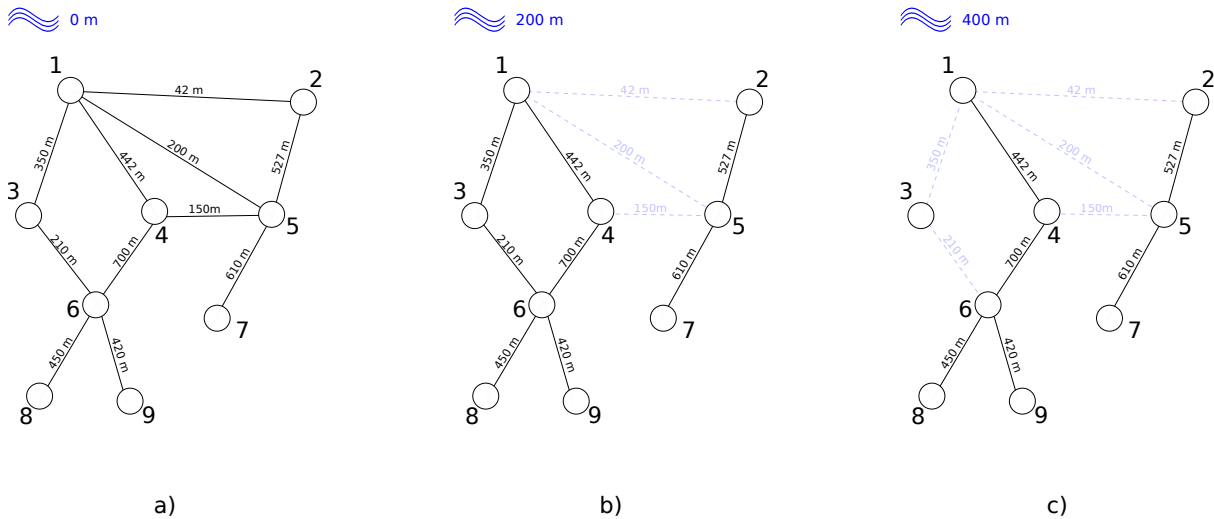
Touledow é um pequeno país turístico localizado no Oceano Pacífico. O país é composto por várias ilhas e algumas pontes entre elas. Cada ponte do país liga duas ilhas, pode ser atravessada nos dois sentidos, e está a alguma altura em relação ao nível do mar.

Por estar em uma região de clima muito instável, o nível da água varia muito em Touledow, podendo aumentar e diminuir com frequência. Quando o nível da água é *igual ou maior* do que a altura de uma ponte, esta ponte fica *submersa* e não pode mais ser utilizada. Por isso, algumas ilhas podem ficar “isoladas” de outras, mesmo havendo pontes, dependendo do nível da água no momento.

Um *arquipélago* é um conjunto de ilhas no qual é possível ir de qualquer ilha para qualquer outra ilha nele, com o nível da água atual.

Como exemplo, a figura abaixo apresenta:

- Um conjunto de ilhas e pontes, com suas respectivas alturas. Com o nível da água em 0 m, todas as pontes podem ser utilizadas e, logo, há 1 (um) arquipélago com todas as 9 ilhas;
- O nível da água subiu para 200 m. Agora, algumas pontes estão submersas e não podem mais ser usadas. Neste momento, há 2 arquipélagos: um com as ilhas 1, 3, 4, 6, 8 e 9, e outro com as ilhas 2, 5 e 7;
- O nível da água subiu para 400 m. Neste momento há 3 arquipélagos: um com as ilhas 1, 4, 6, 8 e 9, outro com as ilhas 2, 5 e 7, e outro com apenas a ilha 3.



Escreva um programa em C que lê do usuário a descrição das ilhas e pontes do país e uma sequência de comandos, e os processa de acordo com o descrito a seguir.

Entrada e Saída

A entrada começa com dois inteiros N e M , o número de ilhas e de pontes, respectivamente. As próximas M linhas da entrada descrevem uma ponte cada. Cada ponte é descrita por três inteiros A , B e h , indicando que a ponte liga as ilhas A e B e está a h metros do nível do mar.

Como exemplo, o conjunto de ilhas e pontes dado na figura acima é descrito pela seguinte entrada:

```
9 11
1 2 42
1 3 350
4 1 442
4 5 150
7 5 610
5 2 527
6 8 450
3 6 210
6 4 700
6 9 420
1 5 200
```

Inicialmente, considere que o nível da água é igual a 0 (zero).

Após ler a descrição das ilhas e pontes, seu programa deverá ler da entrada uma sequência de comandos, onde cada comando pode ser:

- **agua** H : o nível da água mudou para H metros. Imprima uma linha contendo **Nível da água: H m** ;
- **arq**: imprima informações sobre os arquipélagos atuais. Para cada arquipélago, imprima uma linha contendo $[i_1 i_2 \dots]$ (**I ilhas, P pontes**), onde $i_1 i_2 \dots$ é a lista de ilhas que estão no arquipélago, e I e P é o número de ilhas e de pontes no arquipélago, respectivamente. Após imprimir todos os arquipélagos, imprima uma linha contendo **Total: A arquipelagos**, onde A é o número total de arquipélagos. Como exemplo, os arquipélagos mostrados no item b) da figura acima seriam impressos da seguinte forma:

```
[ 1 3 6 8 4 9 ] (6 ilhas, 6 pontes)
[ 2 5 7 ] (3 ilhas, 2 pontes)
Total: 2 arquipelagos
```

Os arquipélagos e as ilhas em cada um deles podem ser impressos em qualquer ordem;

- **caminho** A B : determine o *menor* caminho (em número de pontes) da ilha A para a ilha B . O caminho não pode utilizar nenhuma ponta submersa. Imprima uma linha contendo $i_1 \rightarrow i_2 \rightarrow \dots$ (**P pontes**), onde i_1, i_2, \dots é a sequência de ilhas no caminho, e P é o número total de pontes no mesmo. Como exemplo, um caminho da ilha 9 para a ilha 1, na situação mostrada no item b) da figura acima, seria impresso da seguinte forma: **9 \rightarrow 6 \rightarrow 3 \rightarrow 1 (3 pontes)**. Se houver mais de um caminho com o menor número de pontes possível, imprima qualquer um. Se não houver nenhum caminho possível no momento, imprima **sem caminho entre A e B** .
- **fim**: termina a execução do programa.

Imprima uma linha em branco após cada comando.

Confira o seguinte exemplo completo de entrada e saída.

Exemplo de entrada	Exemplo de saída
9 11 1 2 42 1 3 350 4 1 442 4 5 150 7 5 610 5 2 527 6 8 450 3 6 210 6 4 700 6 9 420 1 5 200 arq caminho 1 7 agua 200 arq caminho 1 7 caminho 9 1 caminho 1 3 agua 400 arq caminho 1 7 caminho 9 1 caminho 1 3 agua 500 caminho 9 1 arq agua 0 caminho 6 2 arq fim	[1 2 5 4 6 8 3 9 7] (9 ilhas, 11 pontes) Total: 1 arquipelagos 1 -> 5 -> 7 (2 pontes) Nivel da agua: 200 m [1 3 6 8 4 9] (6 ilhas, 6 pontes) [2 5 7] (3 ilhas, 2 pontes) Total: 2 arquipelagos sem caminho entre 1 e 7 9 -> 6 -> 3 -> 1 (3 pontes) 1 -> 3 (1 pontes) Nivel da agua: 400 m [1 4 6 8 9] (5 ilhas, 4 pontes) [2 5 7] (3 ilhas, 2 pontes) [3] (1 ilhas, 0 pontes) Total: 3 arquipelagos sem caminho entre 1 e 7 9 -> 6 -> 4 -> 1 (3 pontes) sem caminho entre 1 e 3 Nivel da agua: 500 m sem caminho entre 9 e 1 [1] (1 ilhas, 0 pontes) [2 5 7] (3 ilhas, 2 pontes) [3] (1 ilhas, 0 pontes) [4 6] (2 ilhas, 1 pontes) [8] (1 ilhas, 0 pontes) [9] (1 ilhas, 0 pontes) Total: 6 arquipelagos Nivel da agua: 0 m 6 -> 3 -> 1 -> 2 (3 pontes) [1 2 5 4 6 8 3 9 7] (9 ilhas, 11 pontes) Total: 1 arquipelagos

Implementação

O trabalho deve obrigatoriamente:

- utilizar a *lista de adjacência* como estrutura de dados para representar o grafo;
- utilizar o algoritmo da *Busca em Profundidade (DFS)* para determinar os arquipélagos. Você pode escolher entre implementar a DFS recursiva ou iterativa, como preferir. Caso opte pela implementação iterativa, seu trabalho deve conter e incluir uma biblioteca de Pilha;
- utilizar o algoritmo da *Busca em Largura (BFS)* para determinar os caminhos entre as ilhas. Para tal, seu trabalho deve obrigatoriamente conter e incluir uma biblioteca de Fila.

Certifique-se que toda memória alocada por seu programa é desalocada ao final do mesmo. Você pode assumir que haverá no máximo 100 ilhas no país.

Orientações

- O trabalho pode ser feito por equipes de *até 2* (dois) estudantes;
- Submeta, via *Moodle*, um pacote (zip ou tar.gz) contendo todo o código-fonte necessário para compilar e executar seu trabalho, além de um arquivo de texto (txt) onde conste:
 - O nome de todos os integrantes da equipe;
 - Toda informação que a equipe julgar relevante para a correção (como *bugs* conhecidos, detalhes de implementação, escolhas de projeto, etc.)
- Comente adequadamente seus códigos para facilitar a correção.
- Atenção: a correção será parcialmente automatizada, e a saída do programa será testada com outras entradas além das fornecidas como exemplo. *Siga **fielmente** o formato de saída dado nos exemplos*, sob pena de grande redução da nota;
- Certifique-se que seu programa compila e funciona antes de submetê-lo;
- O trabalho deve ser entregue até **16 de Junho de 2024, 23:59**, apenas via *Moodle*. Trabalhos entregues por outros meios ou fora do prazo não serão aceitos. É suficiente que o trabalho seja submetido por apenas um estudante da equipe;
- Trabalhos detectados como cópia/plágio (de colegas, da internet ou de ferramentas de IA) ou comprados receberão **todos** a nota 0 (**ZERO**) e estarão sujeitos a abertura de Processo Administrativo Disciplinar Discente.