

Trabalho 5: Grafos I

Casos de Teste

Este documento descreve os casos de teste do quinto trabalho da disciplina INF1010 turma 3WB 2018.1:

T5: Implementação de grafos com busca em largura e busca em profundidade.

Instruções para a implementação dos testes foram dadas em sala de aula e podem também ser encontradas na página inicial da plataforma de testes.

Dica: Para testar a sua implementação localmente, direcionando um arquivo para a entrada padrão, adicione por exemplo "< arquivo_entrada.txt" ao comando de executar. Isso pode ser feito no terminal, ou configurado nas IDEs. Da mesma forma, se quiser direcionar a saída para um arquivo, utilize "> arquivo_saida.txt".

Caso 1: Criar e exibir o grafo (2pts)

Teste das funções `insereAresta()` e `mostra()`. Crie grafos sempre de quantidade fixa de vértices, e exiba sua configuração (em texto), após vários comandos de inserção de arestas.

Obs: O exemplo de grafo direcionado dos testes corresponde ao mesmo grafo do item 2 do enunciado.

Entrada:

- **Uma string** indicando o caso de teste: TESTCASE1.
- **Um inteiro** indicando a quantidade de testes que serão feitos (ou grafos que serão criados).
- Para cada teste:
 - **um inteiro N** indicando o número de vértices do grafo.
 - **um inteiro E** indicando o número de arestas do grafo.
 - **um inteiro D** igual a 0 ou 1, indicando se o grafo é direcionado. 0 para não direcionado, 1 para direcionado.
 - **uma sequência de E linhas**, cada uma composta por três inteiros, que são os dois índices dos vértices da aresta, seguidos do peso da aresta.

Saída: Para cada grafo criado, após inseridas todas as arestas, imprimir a sua configuração, no formato exemplificado no item 3 do enunciado do trabalho.

Pule uma linha entre cada teste realizado.

Exemplo de entrada e saída:

Entrada	Saída
TESTCASE1	0: 1(28), 5(10)
2	1: 0(28), 2(16), 6(14)
7	2: 1(16)
7	3: 6(18)
0	4: 5(25), 6(24)
0 1 28	5: 0(10), 4(25)
0 5 10	6: 1(14), 3(18), 4(24)
1 2 16	
1 6 14	0: 5(10)
3 6 18	1: 2(16)
4 5 25	2:
4 6 24	3: 6(18), 4(14)
7	4: 6(24)
6	5: 4(25)
1	6:
0 5 10	
1 2 16	
3 6 18	
3 4 14	
5 4 25	
4 6 24	

Caso 2: Busca em largura (3pts)

Teste a sua implementação da busca em largura - função `bfs()`. Para cada grafo de teste, várias buscas podem ser realizadas, iniciando de vértices diferentes.

Entrada:

- **Uma string** indicando o caso de teste: TESTCASE2.
- **Um inteiro** indicando a quantidade de testes que serão feitos (ou grafos que serão criados).
- Para cada teste:
 - **um inteiro N** indicando o número de vértices do grafo.
 - **um inteiro E** indicando o número de arestas do grafo.
 - **um inteiro D** igual a 0 ou 1, indicando se o grafo é direcionado. 0 para não direcionado, 1 para direcionado.
 - **uma sequência de E linhas**, cada uma composta por três inteiros, que são os dois índices dos vértices da aresta, seguidos do peso da aresta.
 - **um inteiro T** indicando quantas buscas em largura serão realizadas.
 - **uma sequência de T linhas**, cada uma contendo um inteiro *s*, que indica o vértice inicial da busca em largura.

Saída: Para cada busca, imprima os vértices visitados, separados somente por um espaço. Entre cada grafo (e não entre cada `bfs`), salte uma linha.

Exemplo de entrada e saída na próxima página.

Entrada	Saída
TESTCASE2	0 1 5 2 6 4 3
2	6 1 3 4 0 2 5
7	
7	1 2
0	2
0 1 28	5 4 6
0 5 10	
1 2 16	
1 6 14	
3 6 18	
4 5 25	
4 6 24	
2	
0	
6	
7	
6	
1	
0 5 10	
1 2 16	
3 6 18	
3 4 14	
5 4 25	
4 6 24	
3	
1	
2	
5	

Caso 3: Busca em profundidade (3pts)

Teste a sua implementação da busca em profundidade - função `dfs()`. Para cada grafo de teste, várias buscas podem ser realizadas, iniciando de vértices diferentes.

Entrada:

- **Uma string** indicando o caso de teste: `TESTCASE3`.
- **Um inteiro** indicando a quantidade de testes que serão feitos (ou grafos que serão criados).
- Para cada teste:
 - **um inteiro N** indicando o número de vértices do grafo.
 - **um inteiro E** indicando o número de arestas do grafo.
 - **um inteiro D** igual a 0 ou 1, indicando se o grafo é direcionado. 0 para não direcionado, 1 para direcionado.
 - **uma sequência de E linhas**, cada uma composta por três inteiros, que são os dois índices dos vértices da aresta, seguidos do peso da aresta.
 - **um inteiro T** indicando quantas buscas em profundidade serão realizadas.
 - **uma sequência de T linhas**, cada uma contendo um inteiro `s`, que indica o vértice inicial da busca em profundidade.

Saída: Para cada busca, imprima os vértices visitados, separados somente por um espaço. Entre cada grafo (e não entre cada `dfs`), salte uma linha.

Exemplo de entrada e saída na próxima página.

Entrada	Saída
TESTCASE3	0 1 2 6 3 4 5
2	6 1 0 5 4 2 3
7	
7	1 2
0	2
0 1 28	5 4 6
0 5 10	
1 2 16	
1 6 14	
3 6 18	
4 5 25	
4 6 24	
2	
0	
6	
7	
6	
1	
0 5 10	
1 2 16	
3 6 18	
3 4 14	
5 4 25	
4 6 24	
3	
1	
2	
5	

Caso 4: Caminhos entre vértices (2pts)

Resolva o problema de listar todos os caminhos simples possíveis entre um dado par de vértices.

Um caminho simples S é uma sequência de vértices, tal que:

- Se u e v são vértices consecutivos em S , então a aresta $u-v$ pertence ao grafo.
- S não possui arestas repetidas.
- S não possui vértices repetidos.

Por exemplo, no grafo do item 2 do enunciado do trabalho, a sequência 0, 5, 4, 6, 3 é um caminho entre os vértices 0 e 3. A sequência 0, 1, 6, 3 é um outro caminho entre os mesmos vértices.

Implemente um algoritmo que, dado um par de vértices u e v de um grafo NÃO DIRECIONADO, imprima todos os caminhos simples possíveis entre u e v .

Entrada:

- **Uma string** indicando o caso de teste: TESTCASE4.
- **Um inteiro** indicando a quantidade de testes que serão feitos (ou grafos que serão criados).
- Para cada teste:
 - **um inteiro N** indicando o número de vértices do grafo.
 - **um inteiro E** indicando o número de arestas do grafo.
 - **uma sequência de E linhas**, cada uma composta por três inteiros, que são os dois índices dos vértices da aresta, seguidos do peso da aresta.
 - **um inteiro T** indicando quantos pares de vértices terão seus caminhos consultados.
 - **uma sequência de T linhas**, cada uma contendo um inteiro dois inteiros, u e v , indicando os vértices inicial e final, respectivamente, cujos caminhos serão listados.

Saída: Para cada consulta T , imprima os caminhos encontrados separando os vértices somente por um espaço. Entre cada grafo (e não entre cada caminho), salte uma linha.

Exemplo de entrada e saída na próxima página.

Entrada	Saída
TESTCASE4	0 1 2
2	0 5 4 6 1 2
7	5 0 1
7	5 4 6 1
0 1 28	3 6 1 2
0 5 10	3 6 4 5 0 1 2
1 2 16	
1 6 14	0 1 2 4 5
3 6 18	0 1 3 4 5
4 5 25	
4 6 24	
3	
0 2	
5 1	
3 2	
6	
6	
0 1 1	
1 2 1	
1 3 1	
2 4 1	
3 4 1	
4 5 1	
1	
0 5	