

Atividade Prática de Programação - AP11

Busca em Grafo Orientado

Algoritmos e Estruturas de Dados II

Tempo limite: 1s

Instruções:

1. E/S: tanto a entrada quanto a saída de dados devem ser “secas”, ou seja, não devem apresentar frases explicativas.
2. Identificadores de variáveis: escolha nomes apropriados;
3. Documentação: inclua cabeçalho, comentários e indentação no programa;
4. Plágios serão punidos com zero na atribuição da nota da AP;
5. 4. Submeta o programa no sistema Judge

Existem muitas aplicações práticas envolvendo algoritmos de busca em teoria dos grafos, como problemas na aviação e ferroviários, redes de computadores, jogos, entre outros. São também muito usados nas escolhas de melhores rotas de transporte, distribuição de dados, agrupamento de informações, etc.

Para resolver esses problemas, foram desenvolvidas diversas técnicas para a extração e manipulação de informações de um grafo. Dentre essas técnicas destacam-se a busca de elementos através da exploração dos nós em um grafo. Essa exploração pode ser realizada de diversas formas, todavia as mais comuns são a Busca em Largura (*Breadth-First Search* - BFS) e a Busca em Profundidade (*Depth-First Search* – DFS) (Celes et al, 2016; Cormen et al, 2012).

Deve-se escrever um programa que exiba a sequência da Busca em Largura e da Busca em Profundidade de um determinado **grafo orientado**.

Entrada

A entrada para cada teste se inicia informando o nó/vértice onde será iniciada a busca. O nó/vértice inicial tem valor **V** ($0 \leq V \leq 100$), depois deve-se informar o número total de vértices **N** ($0 \leq N \leq 100$) e o número de arestas **M** ($1 \leq M \leq 200$). Nas próximas **M** linhas devem ser fornecidas as arestas que serão criadas. Os vértices são representados por valores numéricos que variam de 0 até 100. Considerar **a ordem crescente** quando for inserido um novo vértice na lista de adjacência. Por exemplo, considere que o nó/vértice 1 aponta (tem aresta de ligação) para 3, 5 e 8. Caso insira uma nova aresta 1 -> 2, o nó/vértice 2 deve ser inserido antes do nó/vértice 3 na lista de adjacência. Caso insira a aresta 1 -> 4, então o nó/vértice 4 deve ser inserido antes do nó/vértice 5 e depois do nó/vértice 3 na lista de adjacência.

Saída

A saída do programa é exibida em duas linhas, onde a primeira linha exibe a Busca em Largura (BFS) do grafo, e a segunda linha exibe a Busca em Profundidade (DFS) do grafo. Com base no nó/vértice onde foi iniciada a busca.

Dicas Restrições:

- Pode-se utilizar as estruturas de dados do tipo Pilha e Fila como mecanismos auxiliares;
- O programa deve ser escrito em C/C++ ou Java;
- Deve-se ter no código as funções de: inicialização, inserção e busca;
- Deve-se utilizar uma **Lista de Adjacência** como estrutura de dados para organização do Grafo.
- Não existirão laços (arestas que apontam para o mesmo vértice nas duas extremidades) e arestas duplas idênticas (onde ambas apontam para os mesmos vértices);
- Não existem nós que não possam ser alcançados (nós que não são ligados por nenhuma aresta).

Exemplo de Entrada	Exemplo de Saída
0 8 8 0 1 0 3 1 4 1 5 4 6 6 0 5 2 2 7	0 1 3 4 5 6 2 7 0 1 4 6 5 2 7 3

Exemplo de Entrada	Exemplo de Saída
2 4 5 0 1 1 2 2 0 0 3 3 1	2 0 1 3 2 0 1 3

Exemplo de Entrada	Exemplo de Saída
7 8 9 1 0 2 0 3 0 4 1 6 3 5 2 7 4 7 6 7 5	7 4 5 6 1 2 3 0 7 4 1 0 5 2 6 3

Referências:

CORMEN, Thomas H et al. Algoritmos: teoria e prática. Rio de Janeiro: Campus, 2002. 916 p. 2.ed.

CELES FILHO, Waldemar; CERQUEIRA, Renato Fontoura de Gusmão; RANGEL NETO, José Lucas Mourão. Introdução a estruturas de dados: com técnicas de programação em C., 2016. 410 p. 2ª ed.