GRAPHS

ELEMENTOS

Um grafo G = V,E é definido por um par de dois conjuntos: um conjunto finito não vazio V de itens chamados vértices e um conjunto E de pares desses itens chamados arestas. . Um grafo G é chamado não direcionado se todas as arestas nele são sem direção. O grafo cujas arestas são direcionadas é chamado de direcionado, também chamado de dígrafo.Um grafo com cada par de seus vértices conectados por uma aresta é chamado de completo.
Um grafo com relativamente poucas arestas possíveis faltando é chamado de denso; um grafo com poucas arestas em relação ao número de seus vértices é chamado esparso.

REPRESENTAÇÃO

Os grafos para algoritmos de computador geralmente são representados de duas maneiras: a matriz de adjacência e as listas de adjacência. a adjacência
matriz de um grafo com n vértices é uma matriz booleana n × n com uma linha e uma coluna para cada um dos vértices do grafo, em que o elemento da i-ésima linha e a j-ésima coluna é igual a 1 se houver uma aresta do i-ésimo vértice ao j-ésimo vértice, e igual a 0 se não houver tal aresta.
As listas de adjacência de um grafo ou digrafo são uma coleção de listas encadeadas, um para cada vértice, que contém todos os vértices adjacentes ao vértice da lista (ou seja, todos os vértices conectados a ele por uma aresta).

DFS

Em cada iteração, o algoritmo prossegue para um vértice não visitado que é adjacente àquele em que está atualmente. (Se houver vários desses vértices, um empate podem ser resolvidos arbitrariamente. Por uma questão prática, qual das áreas adjacentes não visitadas candidatos é escolhido é ditado pela estrutura de dados que representa o gráfico. Em nossos exemplos, sempre desempatamos pela ordem alfabética dos vértices.) Isso
o processo continua até um beco sem saída - um vértice sem vértices adjacentes não visitados - é encontrado. Em um beco sem saída, o algoritmo faz o backup de uma aresta para o vértice
de onde veio e tenta continuar visitando vértices não visitados a partir daí. O algoritmo eventualmente para depois de retornar ao vértice inicial, com o último sendo um beco sem saída. Até então, todos os vértices no mesmo componente conectado como o vértice inicial foram visitados. Se os vértices não visitados ainda permanecerem, a profundidade primeiro
a pesquisa deve ser reiniciada em qualquer um deles.

BFS/TOPOSORT

Se a busca em profundidade for uma travessia para os corajosos (o algoritmo vai tão longe de "casa" como pode), a busca em largura é uma travessia para os cautelosos. Ele procede em de maneira concêntrica, visitando primeiro todos os vértices adjacentes a um ponto inicial vértice, então todos os vértices não visitados a duas arestas dele, e assim por diante, até que todos
os vértices no mesmo componente conectado como o vértice inicial são visitados.
Se ainda restarem vértices não visitados, o algoritmo deve ser reiniciado em um vértice arbitrário de outro componente conexo do grafo.
É conveniente usar uma fila (observe a diferença da pesquisa em profundidade!) para rastrear a operação de busca em largura.Em termos de
este dígrafo, a questão é se podemos listar seus vértices em tal ordem que para cada aresta no grafo, o vértice onde a aresta começa é listado antes do vértice onde termina a aresta. (Você pode encontrar tal ordenação do dígrafo vértices?) Este problema é chamado de ordenação topológica.