

Universidade Federal Rural de
Pernambuco
Unidade Acadêmica de Garanhuns

Grafos

Busca em Largura
Busca em Profundidade

Igor Medeiros Vanderlei

Igor.vanderlei@gmail.com

Busca em Largura



Dado um grafo $G=(V,E)$ e um vértice de origem distinta s , a busca em largura explora sistematicamente as arestas de G até “descobrir” cada vértice acessível a partir de s .

- O algoritmo calcula a distância (menor número de arestas) desde s até todos os vértices acessíveis.
- Ele também produz uma “árvore primeiro na extensão” com a raiz em s que contém todos os vértices acessíveis.

Busca em Largura

-

Para qualquer vértice v acessível a partir de s , o caminho na árvore de s até v corresponde a um “caminho mais curto” de s até v em G .

- O algoritmo funciona sobre grafos orientados ou não orientados.
- O algoritmo descobre todos os vértices à distância k de s , antes de descobrir os vértices à distância $k + 1$.

Busca em Largura

-

Cada vértice é pintado de branco, cinza ou preto.

- No início, todos os vértices são brancos.
- Quando um vértice é descoberto pela primeira vez, ele passa para a cor cinza e

entra na fila para que seus vértices adjacentes possam ser visitados.

- Após a visita de todos os nós adjacentes ao vértice v , v passa para a cor preta.

Busca em Largura

-

Desta forma, os vértices cinza formam uma fronteira entre os brancos e os pretos.

- A busca em largura constrói uma árvore “primeiro na extensão”, tendo como raiz o vértice inicial s .

- Sempre que um vértice branco v é descoberto durante a varredura dos vizinhos de u já descoberto, o vértice v e a aresta (u, v) são adicionados à árvore e dizemos que u é o predecessor de v .

Busca em Largura

Busca em Largura

Busca em Largura

-

O procedimento de busca em largura constrói a árvore primeiro na extensão, à medida que pesquisa o grafo. A árvore é representada pelo campo Π de cada vértice. Formalmente, define-se o subgrafo predecessor de G como:

Busca em Largura

-

Para imprimir um caminho mais curto entre o nó inicial s e algum nó acessível, basta percorrer a árvore

criada pelo algoritmo de busca em largura.

Busca em Profundidade

-

A estratégia da busca em profundidade consiste em procurar “mais fundo”, ou seja, mais distante do vértice inicial, sempre que possível.

- As arestas são exploradas a partir do vértice mais recentemente descoberto v que ainda tem arestas inexploradas.

- Quando se esgotam todas as arestas de v , a busca regressa para explorar as arestas do vértice a partir do qual v foi descoberto.

Busca em Profundidade

-

Esse processo continua até descobrirmos todos os vértices acessíveis a partir do vértice de origem inicial. Se restarem vértices não descobertos, um deles será selecionado como nova origem e a busca se repetirá a partir dele.

- O processo inteiro se repete até que não haja mais nenhum vértice descoberto.

Busca em Profundidade

-

De forma similar à busca em largura, sempre que um vértice v é descoberto a partir da varredura da lista de adjacências de um vértice já descoberto u , registra-se u como pai de v .

- Como já vimos, o subgrafo predecessor da busca em largura forma uma árvore.
- Na busca em profundidade, o subgrafo predecessor pode ser composto por várias árvores.

Busca em Profundidade

-

O subgrafo predecessor de uma busca em profundidade forma uma floresta primeiro na profundidade, composta por árvores primeiro na profundidade.

- Os vértices também serão coloridos para indicar o seu status.
- Ao iniciar a busca, todos os vértices recebem a cor branca.
- Quando um vértice é descoberto ele passa para a cor cinza.

- Ele passa para a cor preta quando sua lista de adjacência é completamente examinada, garantindo que cada vértice só aparecerá em uma árvore.

Busca em Profundidade

-

Além das cores, o algoritmo de busca em profundidade identifica cada vértice com dois carimbos de tempo: o primeiro $d[v]$ registra quando o vértice foi descoberto pela primeira vez; o segundo $f[v]$ registra quando o vértice é terminado (pintado de preto).

- O vértice v é branco antes do tempo $d[v]$, é cinza entre $d[v]$ e

$f[v]$, e é preto após o tempo $f[v]$.

Busca em Profundidade

Busca em Profundidade

Busca em Profundidade

-

Classificação de arestas.

- A busca em profundidade pode ser utilizada para classificar as arestas do

grafo de entrada. Essa classificação pode reunir informações importantes sobre os grafos.

- São definidos 4 tipos de arestas:
 - Arestas de árvore;
 - Arestas de retorno;
 - Arestas direta;
 - Arestas cruzadas;

Busca em Profundidade

-

O algoritmo de busca em profundidade pode ser modificado para classificar as arestas a medida

que as encontra. Aa idéia básica é que a classificação de uma aresta (u, v) pode ser realizada baseada na cor do vértice v .

- Branco: aresta de árvore
- Cinza: aresta de retorno
- Preto: aresta direta ou cruzada (não tem como diferenciar apenas pela cor).
 - Aresta direta se $d[u] < d[v]$
 - Aresta cruzada se $d[u] > d[v]$
- Arestas diretas e cruzadas nunca ocorrem em uma busca em profundidade de um grafo não orientado.

Aplicação: Ordenação Topológica

-

Uma ordenação topológica de um grafo acíclico orientado é uma ordenação linear de todos os seus vértices de tal modo que se G possui uma aresta (u, v) , então u aparece antes de v na ordenação.

- Pode ser vista como uma ordenação dos vértices em uma linha horizontal de tal modo que todas as arestas “apontam” para a direita.

Aplicação: Ordenação Topológica

-

Grafos acíclicos orientados podem ser utilizados para modelar processos nos quais os eventos possuem uma ordem de precedência, ex:

- Gerencia de projetos de desenvolvimento de software
- Linha de montagem de fábrica
- Uma aresta orientada (u, v) significa que o evento u deve ser realizado antes do v .

Aplicação: Ordenação Topológica

Aplicação: Ordenação Topológica

-

Implementação: