

Universidade Federal Rural de
Pernambuco
Unidade Acadêmica de Garanhuns

Grafos

Igor Medeiros Vanderlei

Igor.vanderlei@gmail.com

Introdução

-

Muitos problemas do cotidiano podem ser formulados em termos de **objetos** e

das **conexões** entre eles.

- **Por exemplo:** Dados as rotas de ônibus intermunicipais do Brasil. Poderíamos fazer perguntas como: “Qual é a forma mais rápida de viajar entre Barreirinhas/MA e Itabuna/BA?”
- Para responder esta pergunta nós precisamos apenas informações sobre as **conexões** (rotas de ônibus) entre os **objetos** (cidades).

Introdução

-

Considere um circuito elétrico, formado por resistores, indutores, capacitores e transistores, interligados por conectores. Podemos modelar este sistema no

computador e responder perguntas como: “Todos os componentes estão conectados?” ou “Se este circuito for construído, vai funcionar?”.

- A primeira pergunta depende apenas da análise do cabeamento (conexão) entre os componentes elétricos (objetos).

Introdução

-

Dado um processo de manufatura, no qual as tarefas precisam ser feitas em uma determinada ordem, considere o problema de agendamento das tarefas: “Em que

momento cada tarefa precisa ser realizada?”

- Neste caso, as tarefas são os objetos e as conexões indicam quando uma tarefa precisa ser realizada antes das outras.

Grafos

-

Grafo é um conceito matemático que modela adequadamente essas situações.

- Listas encadeadas e árvores podem ser consideradas instâncias de grafos.
- Teoria dos grafos é um ramo da matemática combinatória.

Nesta disciplina, estudaremos apenas os aspectos algorítmicos.

Conceitos Básicos

-

Um **grafo** é uma coleção de **vértices** e **arestas**.

- Vértices são objetos simples que podem ter nomes e outras propriedades.
- Uma aresta é uma conexão entre dois vértices.

Conceitos Básicos

-

Podemos representá-los graficamente desenhando pontos para os vértices e linhas que ligam esses pontos para as arestas.

Conceitos Básicos

-

Definimos esse grafo pelo seu conjunto de vértices $\{A, B, C, D, E, F, G, H\}$ e o seu conjunto de arestas $\{AB, AC, AG, AF, GE, ED, EF, JM, JL, JK, ML, IH\}$

Conceitos Básicos

-

Em algumas aplicações (como o exemplo do transporte) não faz sentido reorganizar o posicionamento dos vértices. Em outras (o circuito) esta organização geométrica é indiferente.

Conceitos Básicos

-

Um **caminho** entre os vértices x e y é uma lista de vértices no qual os vértices sucessivos são conectados por arestas.

- Ex: BAFEG é um caminho entre B e G

Conceitos Básicos

-

Um grafo é **conectado** se para cada vértice existir um caminho para todos os outros vértices.

- Um grafo que não é conectado é formado por **componentes conectados**.

- No exemplo utilizado temos 3 componentes conectados

Conceitos Básicos

-

Um **caminho simples** é um caminho no qual nenhum vértice se repete. Ex: BAFEGAC não é simples.

- Um **ciclo** é um caminho simples, exceto pelo fato primeiro e o último vértice é o mesmo. Ex: AFEGA é um ciclo.

Conceitos Básicos

-

Seja V o número de vértices e E o número de arestas de um grafo.

Observe que E pode variar entre 0 e $1/2V(V-1)$.

- Grafos que contêm todas as arestas possíveis são chamados de **completos**.
- Grafos com poucas arestas ($< V \log V$) são chamados de **esparsos**.
- Grafos com muitas arestas são chamados de **densos**.

Conceitos Básicos

-

Podemos também classificar os grafos em direcionados ou não-direcionados.

- Em aplicações mais complexas, as arestas podem ter pesos.

Representação

-

Para processar os grafos em programas de computador, inicialmente precisamos decidir de que forma vamos representá-los.

- Existem duas formas de representação comumente utilizadas: **matriz de adjacências** ou **lista de adjacências**. A melhor escolha entre as representações depende se o grafo é denso ou esparsos.

Representação

-

O primeiro passo consiste em mapear os nomes dos vértices em números inteiro entre 0 e $V - 1$.

- A matriz de adjacência é então construída como um array bidimensional ($V \times V$) de valores booleanos.

- $a[x][y]$ assumirá o valor TRUE se existir aresta entre os vértices x e y . Caso contrário assumirá o valor FALSE.

Representação

Matriz de Adjacências

-

Por simplicidade, considere que cada nome de grafo é formado por apenas uma letra do alfabeto e que a i -ésima letra do alfabeto será mapeada para o inteiro $i-1$.

Representação

Matriz de Adjacências

-

Como estamos implementando grafos não-direcionados, a aresta conectando x e y é representada por dois bits $a[x][y]$ e $a[y][x]$.

- Note também que algumas vezes é conveniente assumir que existe um “aresta” conectando cada vértice a si mesmo, então $a[x][x] = 1$, para $0 \leq x < V$.

Representação - Exercício

-

Escreva um programa que recebe como entrada o caminho de um arquivo contendo a descrição de um grafo, constrói sua matriz de adjacência e imprime a matriz na saída padrão.

- A primeira linha do arquivo contém dois números inteiros (V e E), informando o número de vértices e arestas, respectivamente.
- Logo em seguida, o arquivo contém V linhas, cada um com o nome (apenas uma letra cada) de um vértice.
- Por fim, o arquivo contém E linhas, representando as

arestas.

Representação - Exercício

-

Exemplo de entrada / saída

Entrada

3 2

A

D

E

AD

DE

A
D
E
A

1
1
0
D
1
1
1
E
0
1
1

Saída

Representação

-

Matrizes de adjacências são adequadas para representar grafos densos. Em grafos esparsos, a inicialização da matriz pode ser o fator dominante no tempo de execução do algoritmo.

- Na representação por lista de adjacências, cada vértice possui uma lista encadeada de vértices com os quais existe uma aresta que os conecta.

Representação

Lista de Adjacências

-

Observe que novamente cada aresta é representada duas vezes.

Representação - Exercício

-

Escreva um programa que recebe como entrada o caminho de um arquivo contendo a descrição de um grafo, constrói sua lista de adjacência e imprime a lista na saída padrão.

- A primeira linha do arquivo contém dois números inteiros (V e E), informando o número de vértices e arestas, respectivamente.
- Logo em seguida, o arquivo contém V linhas, cada um com o nome (apenas uma letra cada) de um vértice.

- Por fim, o arquivo contém E linhas, representando as arestas.

Representação - Exercício

-

Exemplo de entrada / saída

Entrada

3 2

A

D

E

AD

DE

Saída

A: D

D: A E

E: D