



Análise e Projeto de Algoritmos

Aula 6

Thiago Cavalcante – thiago.cavalcante@penedo.ufal.br

6 de dezembro de 2019

Universidade Federal de Alagoas – UFAL
Campus Arapiraca
Unidade de Ensino de Penedo

Representação abstrata para descrever:

- Sistemas de transporte
- Interações humanas
- Redes de telecomunicações

Uma mesma definição serve para modelar várias estruturas

$$G = (V, E)$$

$$\left\{ \begin{array}{l} V = \text{conjunto de vértices} \\ E = \text{conjunto de pares de vértices} \\ \text{ou arestas} \end{array} \right.$$

Grafos podem modelar essencialmente **qualquer** relação

- Redes de rodovias, onde os vértices são cidades e as arestas são as estradas
- Circuitos elétricos, onde os vértices são as junções e as arestas são os componentes
- Código-fonte, onde os vértices são as linhas de código e as arestas representam o fluxo entre elas
- Interações humanas, onde os vértices são as pessoas e as arestas representam conexões

A chave para a solução de muitos problemas de algoritmos está em interpretá-los **em termos de grafos**

“Linguagem” para descrever propriedades das relações (problemas complexos → descrição e soluções simples em termos de grafos)

- Criar novos algoritmos de grafos é uma **tarefa complexa**
- Melhor forma: modelar o problema de forma que um **algoritmo já existente** possa ser utilizado
- É importante se familiarizar com problemas de algoritmos com grafos (cada problema tem um algoritmo mais adequado)

Dirigido \times Não-dirigido

- Não-dirigido: para um grafo $G = (V, E)$, a aresta $(x, y) \in E$ implica $(y, x) \in E$
- Exemplo: rodovias \times ruas
- Maioria dos grafos de interesse teórico são não-dirigidos

Ponderado \times Não-ponderado

- Em um grafo ponderado, cada aresta possui um valor numérico associado chamado de peso (pode introduzir um cálculo de custo)
- Exemplo: rodovias
- Exemplo: caminho mais curto

Simplex × Multigrafo

- Um grafo simples não possui laços ou múltiplas arestas conectando os mesmos vértices
- Essas conexões requerem um cuidado especial na hora da implementação de um algoritmo
- Em geral, “grafo” = “grafo simples”

Esparso \times Denso

- Definição vaga
- Em geral, os grafos esparsos possuem uma quantidade de arestas proporcional a V e os grafos densos possuem uma quantidade proporcional a V^2
- Exemplos: ruas e circuitos

Cíclico × Acíclico

- Um grafo acíclico não contém nenhum ciclo
- Exemplo: árvores e DAGs (agendamento)

Embutido \times Topológico

- Qualquer grafo para o qual são atribuídas posições geométricas aos vértices e arestas é um grafo embutido
- Um grafo topológico é definido pela “regra” subjacente
- Exemplos: caixeiro-viajante, grade de pontos

Implícito × Explícito

- Um grafo implícito é construído a medida que vai sendo utilizado

Rotulado × Não-rotulado

- Cada vértice em um grafo rotulado possui um nome ou identificador único, para distinguí-lo de todos os outros vértices
- Exemplo: cidades

- Exemplo de modelagem
- Vértices: pessoas
- Arestas: relação de amizade
- Esses grafos são chamados de redes sociais

Direção: se eu sou seu amigo, isso significa que você é meu amigo? (outro exemplo: grafo de “ouvir falar”)

Peso: o quão próximo você é do seu amigo?

Simplicidade: eu sou meu próprio amigo? (multigrafo: diferentes relações de amizade)

Densidade: quem tem mais amigos?

Geometria: meus amigos moram perto de mim? (redes sociais requerem um grafo embutido)

Implícito/Explícito: você também conhece aquela pessoa? (sites de redes sociais × vida real)

Rótulos: você é um indivíduo ou parte de uma multidão? (em geral, estudo de redes sociais não se importa com nomes)

- Matriz de adjacência
- Lista de adjacência

Matriz de adjacência

Podemos representar um grafo G com uma matriz M de tamanho $n \times n$ (onde n é o número de vértices), de forma que $M(i, j) = 1$ se a aresta (i, j) é parte de G e $M(i, j) = 0$, caso contrário

Essa estrutura permite uma rápida resposta para a pergunta “a aresta (i, j) está presente em G ?”, e também uma rápida atualização do grafo (inserção e remoção de arestas)

Pode desperdiçar espaço em um gráfico com muitos vértices e poucas arestas

Lista de adjacência

Grafos esparsos podem ser representado de forma mais eficiente com listas encadeadas, as quais armazenam os vértices vizinhos de um determinado vértice

Implementação da lista de adjacência

Problema: visitar cada aresta e cada vértice em um grafo de forma sistemática

Comparação: grafo e labirinto

Um labirinto pode ser naturalmente representado por um grafo, onde os vértices denotam uma junção do labirinto e as arestas denotam os caminhos. Sendo assim, um algoritmo para atravessar um grafo deve ser poderoso o suficiente para nos tirar de um labirinto.

Comparação: grafo e labirinto

- **Eficiência:** evitar visitar os mesmos lugares repetidamente
- **Correção:** travessia sistemática que garanta a saída do labirinto, cada vértice e aresta devem ser visitados

Idéia principal: categorizar cada vértice como

- Não descoberto: estado inicial
- Descoberto: foi encontrado, mas suas arestas não foram checadas
- Processado: foi encontrado e todas as suas arestas foram visitadas