Lista de Exercícios 9 — Grafos

QXD0115 – Estrutura de Dados Avançada – Turma 01A – 2021.1

Prof. Atílio Gomes

12 de agosto de 2021

Aluno: [ ] Matrícula: [ ]

Referências: Cormem et al. Introduction to Algorithms

1. O **grau** de um vértice em um grafo não-direcionado é o número de arestas que nele incidem. Em um grafo direcionado, o **grau de saída** de um vértice é o número de arestas que saem dele, e o **grau de entrada** de um vértice é o número de arestas que entram nele. O **grau** de um vértice em um grafo direcionado é seu grau de entrada mais seu grau de saída.

   Dada uma representação por lista de adjacências de um grafo direcionado:

   (a) Qual o tempo necessário para calcular os graus de saída de todos os vértices?

   (b) Qual o tempo necessário para calcular os graus de entrada?

2. Dê uma representação por lista de adjacências para uma árvore binária cheia com sete vértices. Dê uma representação por matriz de adjacências equivalente. Suponha que os vértices são numerados de 1 até 7 como em um heap binário.

3. O **transposto** de um grafo direcionado $G = (V, E)$ é o grafo $G_T = (V, E_T)$, onde $E_T = \{(v, u) \in V \times V : (u, v) \in E\}$. Assim, G_T é G com todas as suas arestas invertidas. Descreva algoritmos eficientes para calcular G_T a partir de G, para a representação por lista de adjacências e também para a representação por matriz de adjacências de G. Analise os tempos de execução de seus algoritmos.

4. O **quadrado** de um grafo direcionado $G = (V, E)$ é o grafo $G_2 = (V, E_2)$ em que $(u, v) \in E_2$ se e somente se G contiver um caminho que tenha no máximo duas arestas entre u e v. Descreva algoritmos eficientes para calcular G_2 a partir de G para uma representação por lista de adjacências e para uma representação por matriz de adjacências de G. Analise os tempos de execução de seus algoritmos.

5. A maioria dos algoritmos em grafos que adota uma representação por matriz de adjacências como entrada exige o tempo $\Omega(V^2)$, mas há algumas exceções. Mostre como determinar se um grafo direcionado G contém um **sumidouro** (isto é, um vértice com grau de entrada $|V| - 1$ e grau de saída 0) no tempo $O(V)$, dada uma matriz de adjacências para G.

6. A **matriz de incidência** de um grafo direcionado $G = (V, E)$ sem nenhum laço é uma matriz $B = (b_{ij})$ de dimensão $|V| \times |E|$ tal que:

$$b_{ij} = \begin{cases} -1, & \text{se} \;\; \text{a aresta } j \text{ sai do vértice } i; \\ 1, & \text{se} \;\; \text{a aresta } j \text{ entra no vértice } i; \\ 0, & \text{se} \;\; \text{caso contrário.} \end{cases}$$

   Descreva o que representam as entradas do produto de matrizes BB_T, onde B_T é a transposta de B.

7. Dada a matriz de adjacências de uma grafo com N vértices, faça um algoritmo que determine se esse grafo é direcionado ou não-direcionado.

8. Um **multigrafo** é um grafo que possui arestas paralelas e/ou laços. Dada uma representação por lista de adjacências de um multigrafo $G = (V, E)$, descreva um algoritmo de tempo $O(V + E)$ para calcular a representação por lista de adjacências do grafo não direcionado equivalente $G' = (V, E')$, onde E' consiste nas arestas em E onde todas as arestas múltiplas entre dois vértices foram substituídas por uma aresta única e onde todos os laços foram removidos.

9. Um grafo **bipartido** $G = (V, E)$ é um grafo cujo conjunto de vértices V pode ser particionado em dois conjuntos disjuntos A e B tais que toda aresta conecta um vértice em A a um vértice em B. Em outras palavras, $G = (V, E)$ é bipartido se e somente se:

   (a) $V = A \cup B$,

   (b) $A \cap B = \emptyset$,

   (c) e para toda aresta $e = \{u, v\} \in E$, temos que $u \in A$ e $v \in B$.

   Sabemos que a matriz de adjacência de um grafo simples desperdiça muito espaço de armazenamento pois a diagonal principal é sempre zero e a matriz é simétrica, ou seja, a informação do triângulo inferior é replicada no triângulo superior. Dado um grafo simples G qualquer, se provarmos que ele é bipartido podemos diminuir consideravelmente o espaço de armazenamento necessário para a matriz de adjacência. Como? Determine um modo de fazer isso. Identifique na Figura 1 (última página) quais grafos são bipartidos e represente-os utilizando essa forma mais compacta que você encontrou. A seguir, calcule a porcentagem de espaço (em número de células) economizada em relação a representação por matriz de adjacência.

Exercícios de Programação

10. Implemente o TAD Grafo utilizando uma matriz de adjacências para armazenar os vértices e arestas.

11. Implemente o TAD Grafo utilizando uma lista de adjacências para armazenar os vértices e arestas.

12. Dizemos que um grafo simples é **k-regular** se todos os seus vértices possuem grau igual a k. Escreva uma função que receba como parâmetro um dos grafos que você programou anteriormente e responda categoricamente (SIM ou NÂO) se o grafo em questão é k-regular. A função deve ainda retornar o valor de k em caso afirmativo.

13. O **complemento** de um grafo G é um grafo $\overline{G}$ tal que $V(\overline{G}) = V(G)$ e tal que dois vértices de $\overline{G}$ são adjacentes se e somente se eles não são adjacentes em G. Escreva uma função que receba como entrada a matriz de adjacência de um grafo simples G qualquer e retorne a matriz de adjacência do grafo $\overline{G}$.

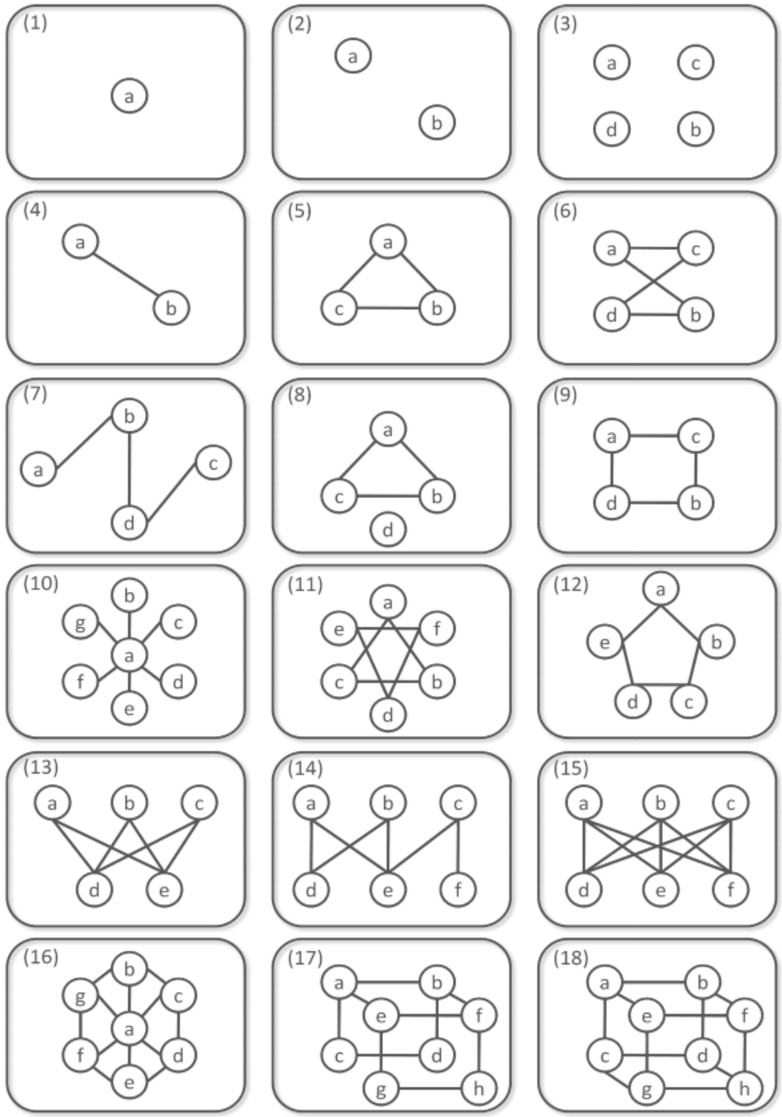

Figura 1: Exemplos de grafos

Referências

[1] Cormen, Thomas H. and Leiserson, Charles E. and Rivest, Ronald L. and Stein, Clifford. Introduction to Algorithms, Third Edition. *The MIT Press (2009)*