O estudo utilizando apenas este material **não é suficiente** para o entendimento do conteúdo. Recomendamos a leitura das referências no final deste material e a resolução (por parte do aluno) de todos os exercícios indicados.

Grafos

Representações computacionais

Conteúdo

Introdução

- Geralmente medimos o tamanho de um grafo $G = (V, E)$ em termos do número de vértice $|V|$ e do número de arestas $|E|$
  - Dentro da notação assintótica, o termo V representará $|V|$, e o termo E, representará $|E|$
- Um grafo $G = (V, E)$ é
  - **Esparso** se $|E|$ é muito menor que $|V|^2$
  - **Denso** se $|E|$ está próximo de $|V|^2$

Representação de grafos

- Existem duas maneiras padrão para representar um grafo $G = (V, E)$
  - Lista de adjacências
  - Matriz de adjacências

Representação de grafos

- Lista de adjacências
  - A **representação de lista de adjacências** consiste de um arranjo `adj` de $|V|$ listas, uma para cada vértice
  - Para cada $u \in V$, a lista de adjacências `adj`$[u]$ contém (ponteiros para) todos os vértices v tal que $(u, v) \in E$

Representação de grafos

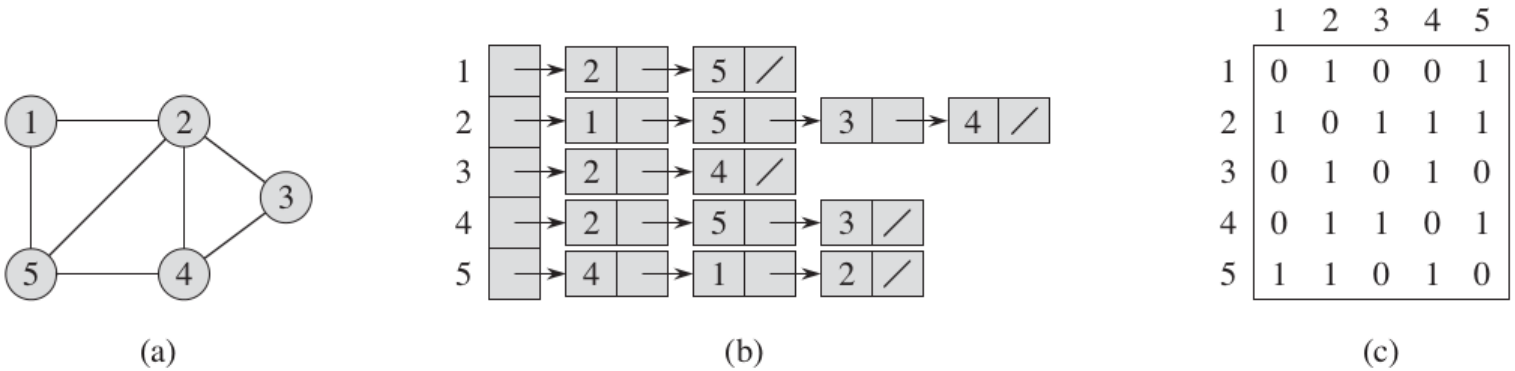

Figure 22.1 Two representations of an undirected graph. **(a)** An undirected graph G with 5 vertices and 7 edges. **(b)** An adjacency-list representation of G. **(c)** The adjacency-matrix representation of G.

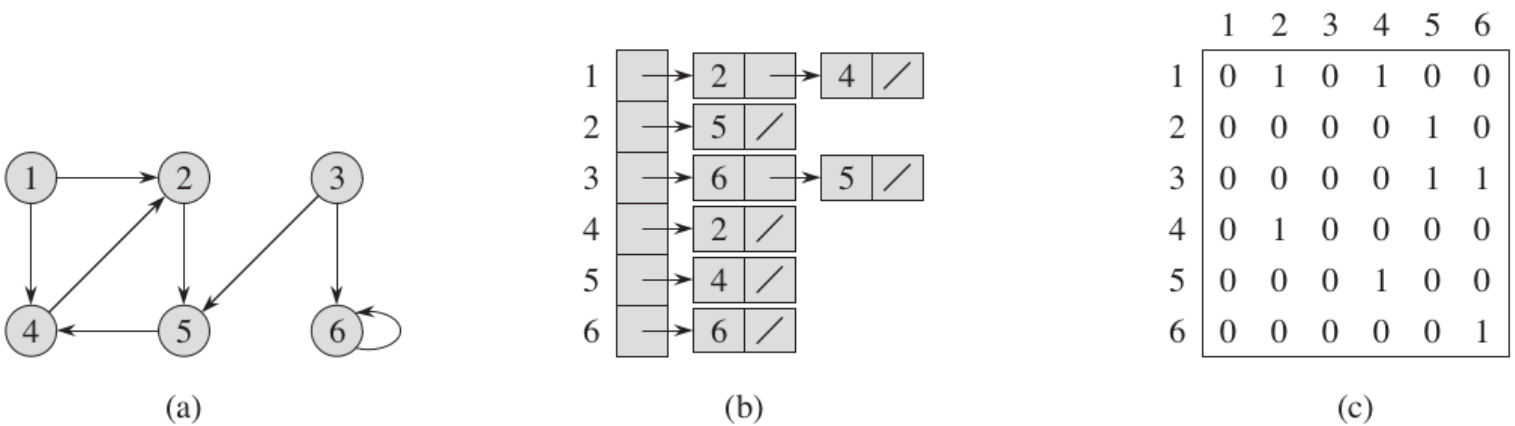

Figure 22.2 Two representations of a directed graph. **(a)** A directed graph G with 6 vertices and 8 edges. **(b)** An adjacency-list representation of G. **(c)** The adjacency-matrix representation of G.

Representação de grafos

- Lista de adjacências
  - Qual é a soma dos comprimentos de todas as listas de adjacências?
    - Se G é um grafo orientado, a soma é $|E|$
    - Se G é um grafo não orientado, a soma é $2|E|$
  - Qual é a quantidade de memória requerida? $\Theta(V + E)$
  - Adequada para grafos esparsos
  - Desvantagem
    - Não existe nenhum modo rápido para determinar se uma dada aresta (u, v) está presente no grafo

Representação de grafos

- Matriz de adjacências
  - Na **representação de matriz de adjacências**, supomos que os vértices são numerados $1, 2, \ldots, |V|$
  - A representação consiste em uma matriz $|V| \times |V| A = (a_{ij})$ tal que

$$a_{ij} = \begin{cases} 1 \text{ se } (i,j) \in E \\ 0 \text{ caso contrário} \end{cases}$$

  - Esta representação permite consultar se uma aresta faz parte do grafo em tempo constante

Representação de grafos

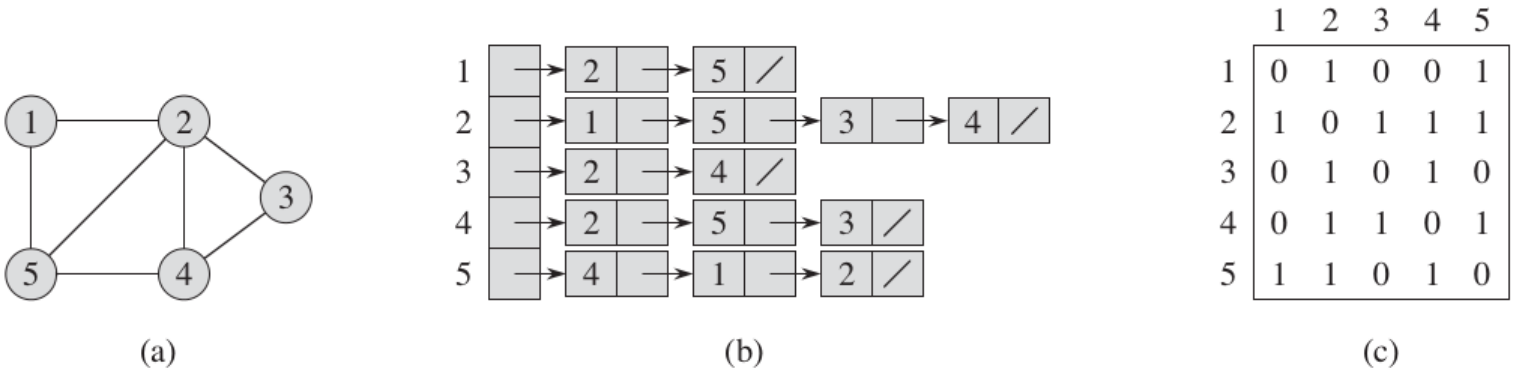

Figure 22.1 Two representations of an undirected graph. **(a)** An undirected graph G with 5 vertices and 7 edges. **(b)** An adjacency-list representation of G. **(c)** The adjacency-matrix representation of G.

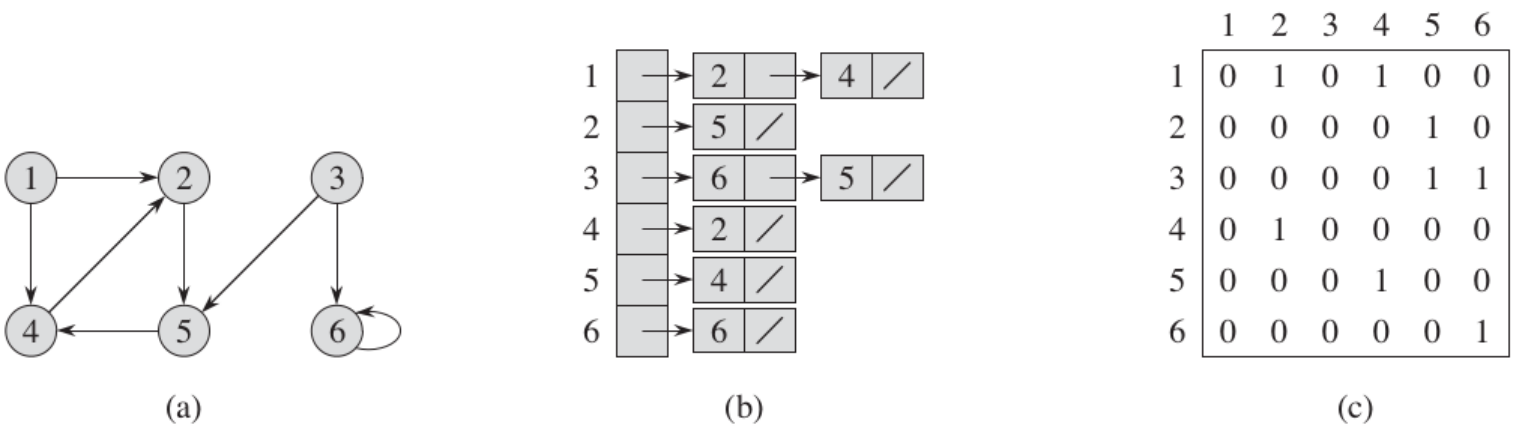

Figure 22.2 Two representations of a directed graph. **(a)** A directed graph G with 6 vertices and 8 edges. **(b)** An adjacency-list representation of G. **(c)** The adjacency-matrix representation of G.

Representação de grafos

- Matriz de adjacências
  - Qual é quantidade de memória requerida? $\Theta(V^2)$. A quantidade de memória independe de E
  - Em um grafo não orientado, a matriz é igual a sua transposta, desta forma é possível usar apenas os elementos abaixo (ou acima) da diagonal principal
  - Adequada para grafos densos

Atributos

- Muitos algoritmos que operam em grafos precisam manter atributos para vértices e/ou arestas
- Nos códigos, indicamos os atributos como
  - `v.d`, atributo `d` do vértice `v`
  - `(u, v).f`, atributo `f` da aresta `(u, v)`
- Como estes atributos podem ser implementados?
  - Depende da linguagem de programação, algoritmo, etc
  - Os atributos podem ser armazenado diretamente na lista ou matriz de adjacência
  - Se os vértices são enumerados de $1..|V|$ os atributos podem ser representados em arranjos, tais como $d[1..|V|]$
  - Atributos de vértices podem ficar nos registros que representam os vértices

Exercícios

- 22.1-1 a 22.1-8

Exercícios

22.1-1 Dada uma representação de lista de adjacências de um grafo orientado, qual o tempo necessário para computar o grau de saída de todo o vértice? Qual o tempo necessário para computar os graus de entrada?

Exercícios - Solução 22.1-1

```
computar-graus-de-saida(G)
1 for v in G.V
2   v.grau-de-saida = 0
3 for v in G.V
4   for u in G.adj[v]
5     v.grau-de-saida += 1
```

```
computar-graus-de-entrada(G)
1 for v in G.V
2   v.grau-de-entrada = 0
3 for v in G.V
4   for u in G.adj[v]
5     u.grau-de-entrada += 1
```

Conforme discutido em sala, ambos algoritmos tem tempo de execução $\Theta(V + E)$.

Exercícios

22.1-2 Forneça uma representação de lista de adjacências para uma árvore binária completa sobre 7 vértices. Forneça uma representação de matriz de adjacências equivalente. Suponha que os vértices estejam numerados de 1 até 7 como em um heap binário.

22.1-3 A **transposta** de um grafo orientado $G = (V, E)$ é o grafo $G^T = (V, E^T)$, onde $E^T = \{(v, u) \in V \times V : (u, v) \in E\}$. Deste modo, G^T é G com todas as suas arestas invertidas. Descreva algoritmos eficientes para calcular G^T a partir de G, para a representação de lista de adjacências e também para a representação de matriz de adjacências de G. Analise os tempos de execução de seus algoritmos.

Exercícios

22.1-4 Dada uma representação de lista de adjacências de um multigrafo $G(V, E)$, descreva um algoritmo de tempo $O(V + E)$ para calcular a representação de lista de adjacência do grafo não orientado "equivalente" $G' = (V, E')$, onde E' consiste nas arestas em E com todas as arestas múltiplas entre dois vértices substituídas por uma aresta única e com todos os autoloops removidos.

Exercícios - Solução 22.1-4

```
transformar-multigrafo-em-grafo(G)
 1 G' = (G.V, 0) // mesmo conjunto de vértices
                 // conjunto vazio de arestas
 2 for v in G.V
 3   v.presente = nil
 4 for u in G.V
 5   for v in G.adj[u]
 6     if v.presente != u and v != u
 7       v.presente = u  // marca a aresta (u, v)
                         // como presente em G'
 8       G'.adj[u].add(v)
```

Conforme discutido em sala, o tempo de execução é $\Theta(V + E)$.

Exercícios

22.1-5 O **quadrado** de um grafo orientado $G = (V, E)$ é o grafo $G^2 = (V, E^2)$ tal que $(u, w) \in E^2$ se e somente se, para algum $v \in V$, tem-se $(u, v) \in E$ e também $(v, w) \in E$. Ou seja, G^2 contém uma aresta entre u e w sempre que G contém um caminho com exatamente duas arestas entre u e w. Descreva algoritmos eficientes para calcular G^2 a partir de G para uma representação de lista de adjacências e para uma representação de matriz de adjacências de G. Analise os tempos de execução de seus algoritmos.

Exercícios

22.1-6 Quando uma representação de matriz de adjacências é usada, a maioria dos algoritmos de grafos exige $\Omega(V^2)$, mas existem algumas exceções. Mostre que detectar se um grafo orientado G contém um **sorvedor universal** – um vértice com grau de entrada $|V| - 1$ e grau de saída 0 – é uma operação que pode ser realizada no tempo $O(V)$, dada uma matriz de adjacências de G.

Exercícios

22.1-7 A **matriz de incidência** de um grafo orientado $G = (V, E)$ é uma matriz $|V| \times |E| B = (b_{ij})$ tal que

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

Descreva o que representa as entrada do produto de matrizes BB^T, onde B^T é a transposta de B.

Exercícios

22.1-8 Suponha que, em vez de uma lista ligada, cada entrada de arranjo `adj[u]` é uma tabela hash contendo os vértices v para os quais $(u, v) \in E$. Se todas as pesquisas de arestas forem igualmente prováveis, qual será o tempo esperado para determinar se uma aresta está no grafo. Que desvantagens esse esquema apresenta? Sugira uma estrutura de dados alternativa para cada lista de arestas que resolva estes problemas. Sua alternativa tem desvantagens em comparação com a tabela hash?

Referências

- Thomas H. Cormen et al. Introduction to Algorithms. 3rd edition. Capítulo 22.1.