Representação

Algoritmos em Grafos

Marco A L Barbosa

Conteúdo

Introdução

Representação de grafos

- Lista de adjacências
- Matriz de adjacências
- Atributos
- Exemplos de implementação

Exercícios

Referências

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.

Introdução

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

Representação de grafos

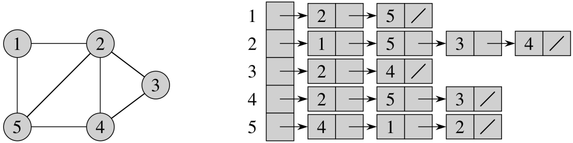
- Existem duas maneiras padrão para representar um grafo $G = (V, E)$
  - Lista de adjacências
  - Matriz de adjacências

Lista 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 $(u, v) \in E$
  - No pseudo do código vamos tratar o arranjo Adj como um atributo do grafo, assim como V e E
    - $G.V$ - conjunto de vértices
    - $G.E$ - conjunto de arestas
    - $G.Adj$ - arranjo com as listas de adjacências

Representação de grafos



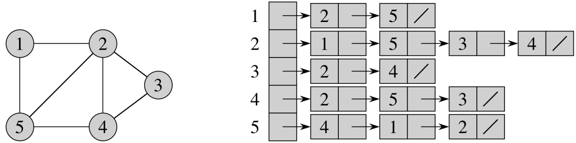
(a) (b)

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

(c)

Figura: 22-1

Representação de grafos

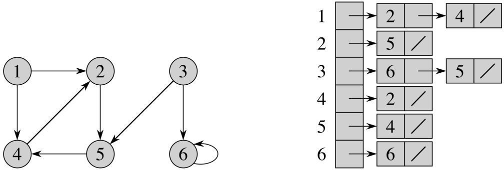


(a) (b)

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

(c)

Figura: 22-1



(a) (b)

	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

(c)

Figura: 22-2

Representação de grafos

- Lista de adjacências
  - Qual é a soma dos comprimentos de todas as listas de adjacências?

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 $|E|$

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 $2|E|$

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?

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? $\Theta(V + E)$

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

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
  - Vantagens

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
  - Vantagens
    - Flexível, é possível adaptar a representação para variantes de grafos
    - A quantidade de memória é assintoticamente ótima

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
  - Vantagens
    - Flexível, é possível adaptar a representação para variantes de grafos
    - A quantidade de memória é assintoticamente ótima
  - Desvantagem

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
  - Vantagens
    - Flexível, é possível adaptar a representação para variantes de grafos
    - A quantidade de memória é assintoticamente ótima
  - Desvantagem
    - Não existe nenhum modo rápido para determinar se uma dada aresta (u, v) está presente no grafo

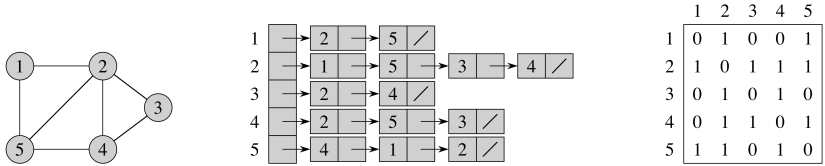
Matriz de adjacências

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

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

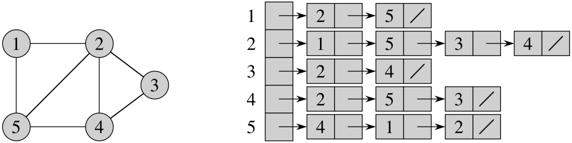
Representação de grafos



(a) (b) (c)

Figura: 22-1

Representação de grafos

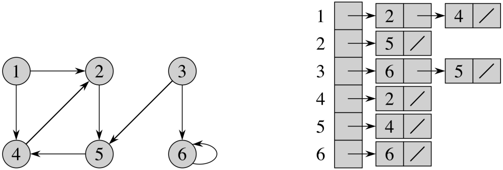


(a) (b)

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

(c)

Figura: 22-1



(a) (b)

	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

(c)

Figura: 22-2

Representação de grafos

- Matriz de adjacências
  - Qual é quantidade de memória requerida?

Representação de grafos

- Matriz de adjacências
  - Qual é quantidade de memória requerida? $\Theta(V^2)$. A quantidade de memória independe de E

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

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

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
  - Vantagens

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
  - Vantagens
    - Simplicidade
    - Permite consultar se uma aresta faz parte do grafo em tempo constante

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
  - Vantagens
    - Simplicidade
    - Permite consultar se uma aresta faz parte do grafo em tempo constante
  - Desvantagens

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
  - Vantagens
    - Simplicidade
    - Permite consultar se uma aresta faz parte do grafo em tempo constante
  - Desvantagens
    - Memória

Atributos

Atributos

- Muitos algoritmos que operam em grafos precisam manter atributos para vértices e/ou arestas

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)

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?

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

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
  - Atributos de arestas podem ficar nos registros que representam as arestas

Exemplos de implementação

Exemplos de implementação

Veja o arquivo 02-representacao-exemplo.zip

Exercícios

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?

Resolução 22.1-1

- Antes de fazer a análise do tempo de execução é necessário escrever o pseudo código do algoritmo

Resolução 22.1-1

```
computa-graus-de-saida(G)
for u in G.V
  u.grau-de-saida = 0
for u in G.V
  for v in G.Adj[u]
      u.grau-de-saida += 1
```

Resolução 22.1-1

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

- Análise do tempo de execução
  - O laço das linhas 2 a 3 demora $\Theta(V)$
  - O laço da linha 4 (sem contar as linhas 5 e 6) demora $\Theta(V)$
  - A cada interação do laço da linha 4, o laço das linhas 5 a 6 é executado $|G.Adj[u]|$ vezes, como o laço da linha 4 é executado uma vez para cada vértices, temos que o laço das linhas 5 a 6 é executado $\sum_{u \in G.V} |G.Adj[u]| = |E|$. Ou seja, o tempo de execução das linha 5 e 6 é $\Theta(E)$
  - Portanto, o tempo de execução do procedimento `computa-graus-saida` é $\Theta(V + E)$

Resolução 22.1-1

```
computa-graus-de-entrada(G)
for u in G.V
  u.grau-de-entrada = 0
for u in G.V
  for v in G.Adj[u]
    v.grau-de-entrada += 1
```

- Análise do tempo de execução
  - Mesmo do procedimento `computa-graus-de-saida`

Referências

Referências

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