Ciências da Computação
UFPA

Grafos

Representações Computacionais de Grafos

19 de abril de 2023

Representando Grafos

Grafos podem ser representados computacionalmente utilizando dois tipos de estruturas:

- Matrizes
- Listas de Adjacências

Representando Grafos

Grafos podem ser representados computacionalmente utilizando dois tipos de estruturas:

- Matrizes
- Listas de Adjacências

Obs.:

- A escolha da representação mais apropriada depende de fatores como veremos adiante;
- Existem outros tipos de representações.

Grafo Denso e Esparso

Antes de apresentarmos as representações precisamos definir o que são grafos densos e esparsos. Em geral, podemos dizer que:

Grafo Denso

Possui muitas arestas

Grafo Denso e Esparso

Antes de apresentarmos as representações precisamos definir o que são grafos densos e esparsos. Em geral, podemos dizer que:

Grafo Denso

Possui muitas arestas

Grafos Esparso

Possui poucas arestas

Quantificando Grafos Denso/Esparso

Mais especificamente, vamos definir o que são grafos densos e esparsos de acordo com a seguinte fórmula:

Densidade do Grafo:

Média do grau dos vértices

$$\text{Densidade} = \frac{2A}{V}$$

Quantificando Grafos Denso/Esparso

Mais especificamente, vamos definir o que são grafos densos e esparsos de acordo com a seguinte fórmula:

Densidade do Grafo:

$$\text{Média do grau dos vértices}$$

$$\text{Densidade} = \frac{2A}{V}$$

Assim, podemos afirmar que:

Grafo Denso: A média dos graus dos vértices é proporcional a V.

Quantificando Grafos Denso/Esparso

Mais especificamente, vamos definir o que são grafos densos e esparsos de acordo com a seguinte fórmula:

Densidade do Grafo:

$$\text{Média do grau dos vértices}$$
$$\text{Densidade} = \frac{2A}{V}$$

Assim, podemos afirmar que:

Grafo Denso: A média dos graus dos vértices é proporcional a V.

Grafo Esparso: Contrário de grafo denso.

Representação através de Matriz de Adjacência

Pode ser utilizada tanto para grafos não orientados como para dígrafos.

Seja um grafo $G = (V, A)$ e a matriz $A = a_{ij}$, $n \times n$, onde n é o número de vértices:

$$
\begin{array}{ll}
a_{ij} = 1 & \text{, se } v_i, v_j \in A \\
a_{ij} = 0 & \text{, caso contrário}
\end{array}
$$

$$
A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}
$$

Representação através de Matriz de Adjacência

- Se o grafo não for direcionado, a matriz A é simétrica $A = A^T$;
- Se o grafo for direcionado (dígrafo), a matriz A não é simétrica $A \neq A^T$;
- O espaço de armazenamento necessário em qualquer caso é n^2.

Representação através de Matriz de Adjacência

- Se o grafo não for direcionado, a matriz A é simétrica $A = A^T$;
- Se o grafo for direcionado (dígrafo), a matriz A não é simétrica $A \neq A^T$;
- O espaço de armazenamento necessário em qualquer caso é n^2.
- Grau de um vértice - número de 1's na linha ou coluna correspondente ao vértice;
- Permutações de linhas/colunas resultam em uma representação de um grafo isomorfo (Verifique !).

Representação através de Matriz de Adjacência

Devantagens

- Grafos grandes e esparsos $\rightarrow$ matriz com muitos zeros;
- Consome muito espaço de armazenamento desnecessário;
- Difícil de inserir/deletar vértices $\rightarrow$ a matriz precisa ser alterada.

Representação através de Listas de Adjacências

Ideia:

Associar a cada vértice uma lista contendo os vértices adjacentes.

Representação através de Listas de Adjacências

Ideia:

Associar a cada vértice uma lista contendo os vértices adjacentes.

Definição:

Seja um grafo $G = (V, A)$, a lista de adjacências do vértice v (i.e. $adj(v)$) contém os vértices adjacentes a v.

Ou seja: $adj(v) = \{w | (v, w) \in A\}$

Representação através de Listas de Adjacências

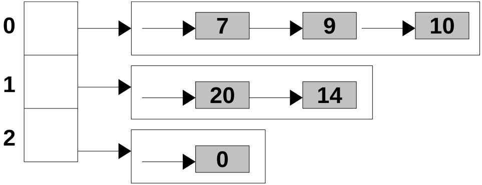
- Grafos não direcionados $\rightarrow$ cada aresta possui duas entradas.
  Ex: Seja (u, v), então $v \in adj(u) \wedge u \in adj(v)$

Representação através de Listas de Adjacências

- Grafos não direcionados $\rightarrow$ cada aresta possui duas entradas. Ex: Seja (u, v), então $v \in adj(u) \wedge u \in adj(v)$
- Dígrafos $\rightarrow$ cada aresta possui uma entrada. Ex: Seja (u, v), então $v \in adj(u)$

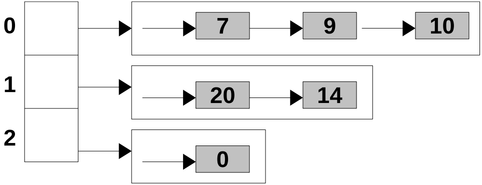
Representação através de Listas de Adjacências

Podemos implementar a lista de adjacências através de um vetor de listas. Por exemplo:



Representação através de Listas de Adjacências

Podemos implementar a lista de adjacências através de um vetor de listas. Por exemplo:



- Em geral as listas são utilizadas na representação de grafos esparsos

Listas de Adjacência x Matrizes

A representação de um grafo também depende do tipo de grafo utilizado. Por exemplo, se o grafo possuir arestas paralelas, não podemos representá-lo através de uma matriz de adjacências.

Listas de Adjacência x Matrizes

A representação de um grafo também depende do tipo de grafo utilizado. Por exemplo, se o grafo possuir arestas paralelas, não podemos representá-lo através de uma matriz de adjacências.

Resumo:

Estrutura	Espaço	Adicionar aresta v-w	Verificar se w é adj. a v	Iterar através de vértices adj. a v
Matriz de Adj.	V^2	1	1	V
Listas de Adj.	Propor. $E+V$	1	$grau(v)$	$grau(v)$

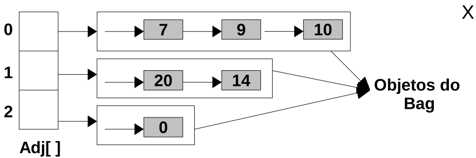
Representando Listas de Adjacências através de Bags

Bags são estruturas de dados destinadas a coleção de elementos em que **a ordem dos elementos é irrelevante** e **não há suporte a remoção de elementos**. Tal estrutua tem como objetivo colecionar objetos e iteragir através da coleção.

Representando Listas de Adjacências através de Bags

Bags são estruturas de dados destinadas a coleção de elementos em que **a ordem dos elementos é irrelevante** e **não há suporte a remoção de elementos**. Tal estrutua tem como objetivo colecionar objetos e iteragir através da coleção.

Pode ser implementado através de uma pilha ou uma lista.



Bibliografia

- NETTO, Paulo O. B. **Teoria e Modelos e Algoritmos**, 4ª. ed. Edgard Blücher. São Paulo, 2006;
- SEDGEWICK, Robert; Wayne K. **Algorithms**, Fourth Edition, Pearson Education, 2011;
- SEDGEWICK, Robert. **Algorithms in Java, Part 5: Graph Algorithms**, 3rd Edition, Addison-Wesley Professional, 2003;