CENTRO DE CIÊNCIA E TECNOLOGIA
LABORATÓRIO DE CIÊNCIAS MATEMÁTICAS
UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE

Grafos

Disciplina: Estrutura de Dados II

Prof. Fermín Alfredo Tang Montané

Curso: Ciência da Computação

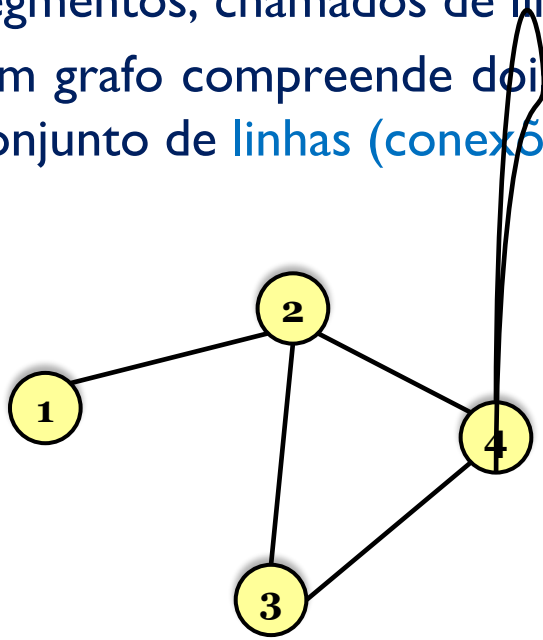
Grafos (*Graphs*)
Definição

- A estrutura de dados grafo, difere de todas as outras pelo seguinte aspecto principal: cada nó pode ter vários predecessores e vários sucessores.
- Os grafos são estruturas muito úteis. Podem ser utilizadas para resolver problemas de roteirização complexos, tais como projetar e roteirizar linhas aéreas entre os aeroportos que elas servem.
- Também podem ser utilizadas para roteirizar mensagens em uma rede de computadores de um nó para outro.

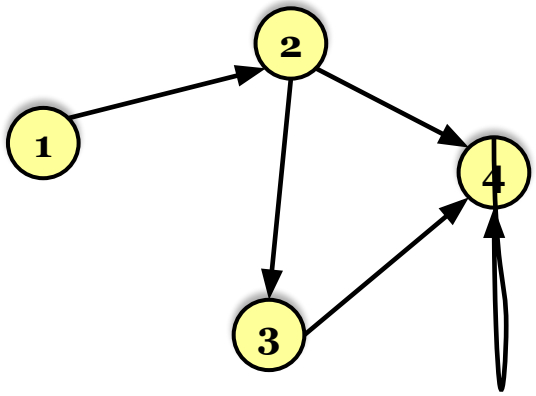
Grafos (*Graphs*)

Definição

- Um grafo é uma coleção de nós, chamados vértices e uma coleção segmentos, chamados de linhas, conectando pares de vértices.
- Um grafo compreende dois conjuntos: um conjunto de vértices e um conjunto de linhas (conexões).



Grafo



Digrafo

Grafos (*Graphs*)
Conceitos Básicos

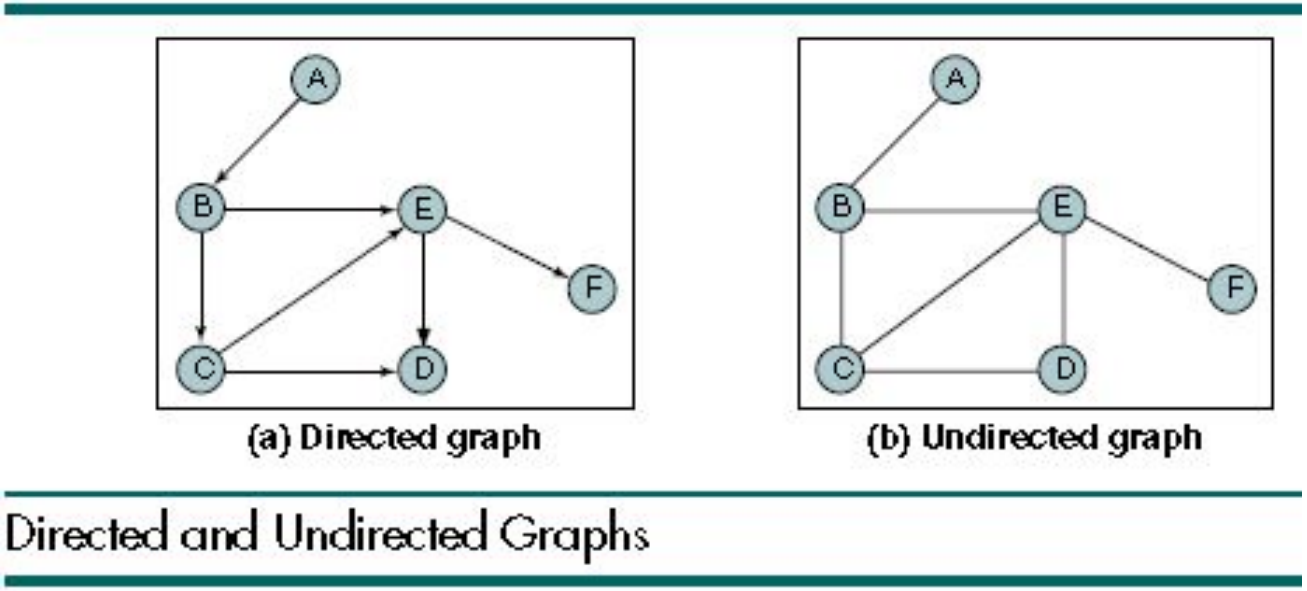
- Os grafos podem ser classificados em dois tipos:
  - grafos direcionados ou digrafos (*directed graphs or digraphs*) e
  - não-direcionados (*undirected graphs*).

- Em um grafo direcionado (ou Digrafo) as linhas (relações entre os objetos) possuem uma direção que serve para indicar o sucessor de um objeto (uma seta na representação gráfica). Neste caso os objetos são chamados de **nós** (nodes) e as linhas entre esses objetos são chamados de **arcos** (arcs). O fluxo entre dois nós deve seguir necessariamente a direção do arco.

- Em um grafo não-direcionado as linhas (relações entre os objetos) não possuem qualquer direção. Neste caso os objetos são chamados de **vértices** (*vertices*) e as linhas são chamados de **arestas** (*edges*). O fluxo entre dois vértices pode seguir qualquer direção.

Grafos (Graphs)
Conceitos Básicos

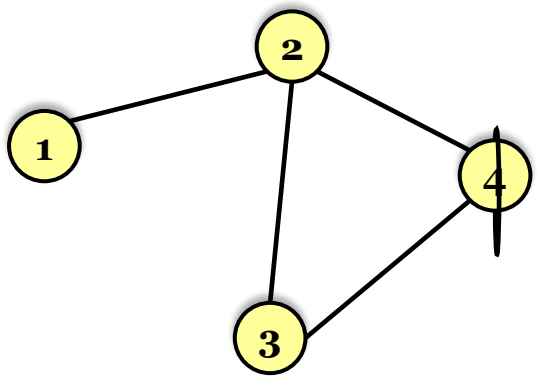
- Grafos Direcionados e não Direcionados



Directed and Undirected Graphs

Grafos (Graphs)
Conceitos Básicos

- Um **grafo** é um modelo matemático que representa as relações entre objetos de determinado conjunto.
- Um grafo G(V, A) é definido em termos de dois conjuntos:
  - Um conjunto V de vértices, que são os itens (objetos) representados em um grafo.
  - Um conjunto A de arestas, que são utilizadas para conectar qualquer par de vértices. Neste caso, dois vértices são conectados segundo critério previamente estabelecido.



Grafo

$$G = (V, A)$$

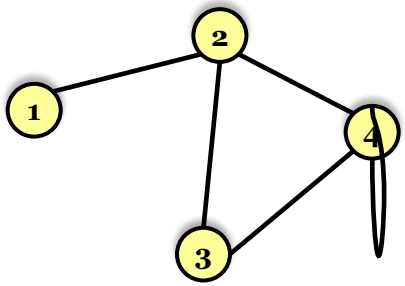
$$V = \{1, 2, 3, 4\}$$

$$A = \{(1,2); (2,3); (2,4); (3,4); (4,4); (2,1); (3,2); (4,2); (4,3);\}$$

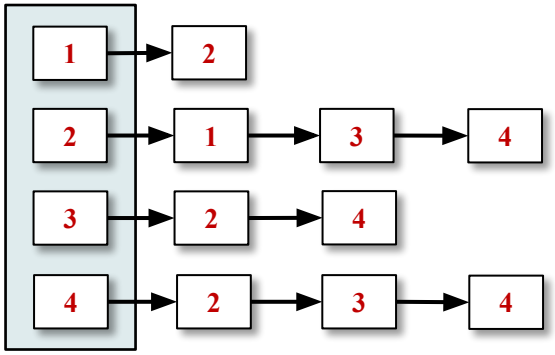
Representação de Grafos

- Ao se modelar um problema utilizando um grafo, surge a questão: como representar este grafo no computador? Existem **duas abordagens** muito utilizadas para representar um grafo no computador:
  - Matriz de adjacência.
  - Lista de adjacência.

- A representação escolhida para um grafo depende da aplicação. Não existe uma representação que seja melhor que a outra em todas os casos.

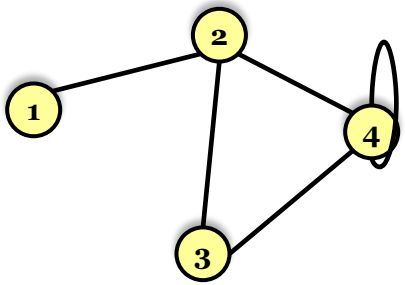


	1	2	3	4
1	0	1	0	0
2	1	0	1	1
3	0	1	0	1
4	0	1	1	1



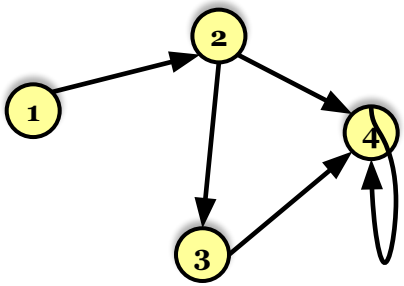
Representação de Grafos
Matriz de Adjacência

- A representação de um grafo por **matriz de adjacência** faz uso de uma simples matriz para descrever as relações entre os vértices.
- Neste tipo de representação, um grafo contendo N vértices utiliza uma matriz de ordem NxN, com N linhas e N colunas para armazenar o grafo.
- Uma aresta ligando dois vértices é representada por uma marca na posição (i, j) da matriz, sendo i o vértice inicial e j o vértice final da aresta (p.e. 1 se existe aresta, e 0 caso não exista).



Grafo

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$



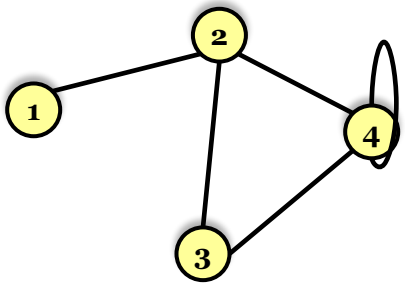
Digrafo

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Representação de Grafos

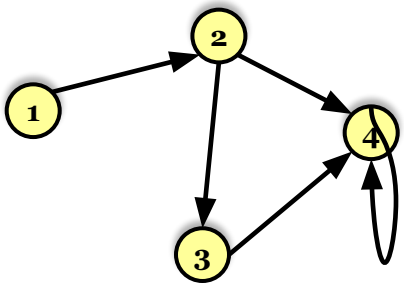
Matriz de Adjacência

- A representação de um grafo por matriz de adjacência possui um alto custo computacional, $O(N^2)$. Além disso, não é indicada para um grafo que possui muitos vértices mas poucas arestas ligando-os.
- Operações como encontrar todos os vértices adjacentes a um vértice exigem que se pesquise em toda a linha da matriz.
- No entanto, se a matriz de adjacências armazenar somente a conectividade dos vértices (arestas), apenas um bit será necessário para cada posição da matriz. Isso tornaria essa representação bastante compacta.



Grafo

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$



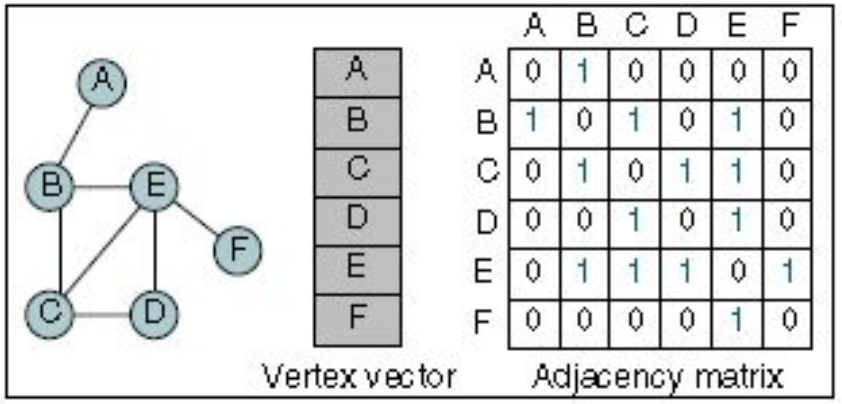
Digrafo

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

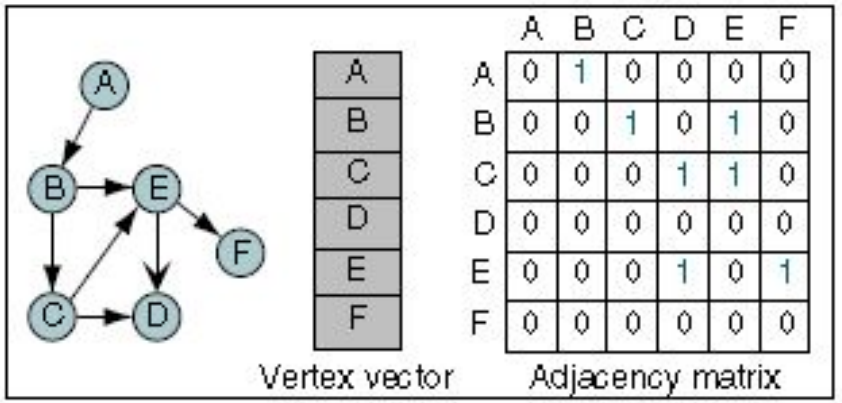
Grafos (Graphs)

Representação – Matriz de Adjacência

- Matriz de Adjacência



(a) Adjacency matrix for nondirected graph



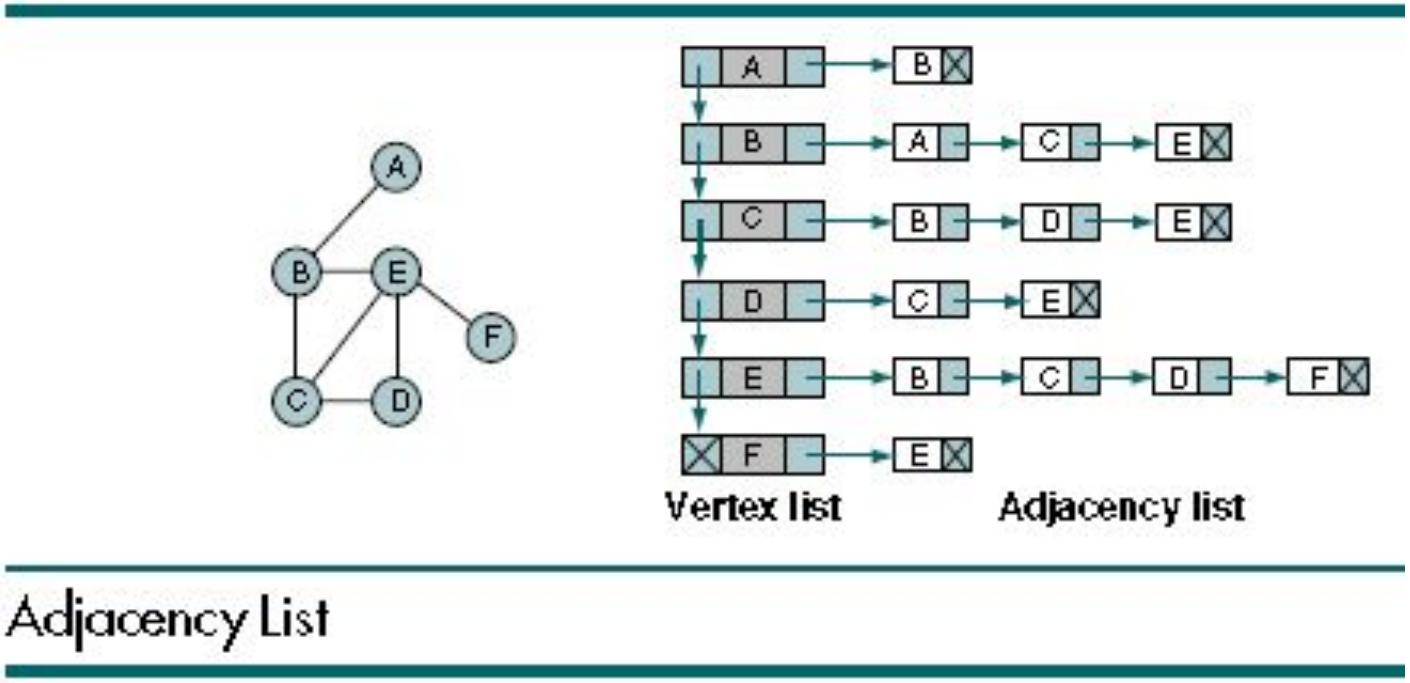
(b) Adjacency matrix for directed graph

Adjacency Matrix

Grafos (Graphs)

Representação – Lista de Adjacência

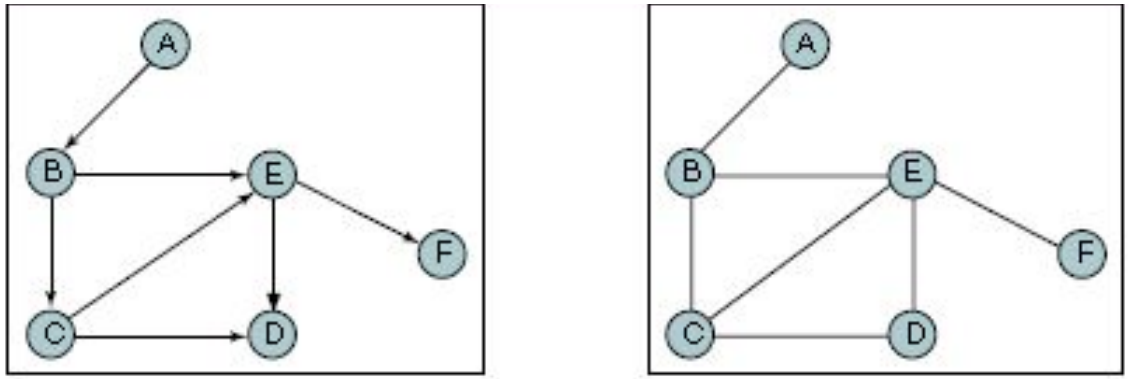
- Lista de Adjacência



Grafos (Graphs)

Vértices (Nós) adjacentes

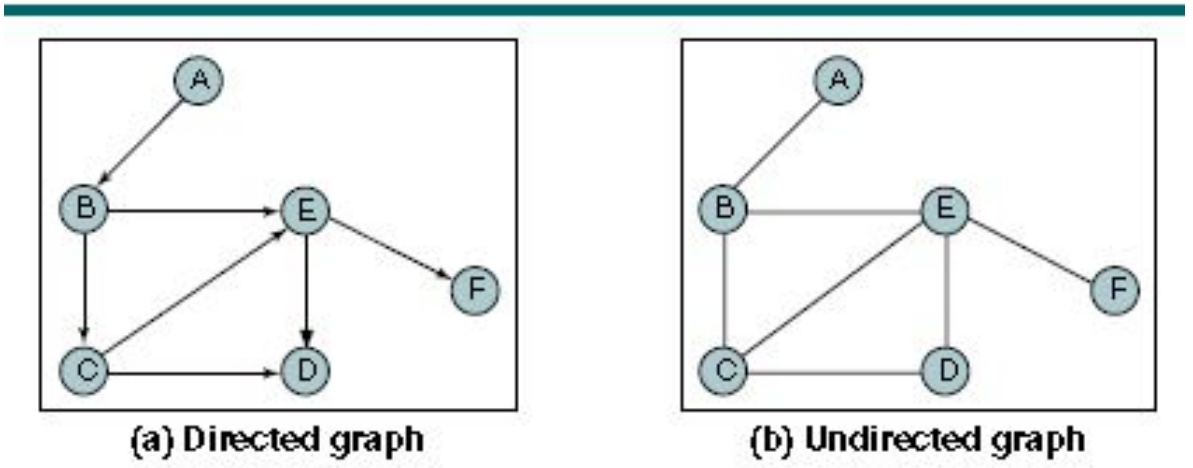
- Dois **vértices** (ou **nós**) em um grafo são ditos **vértices adjacentes** (ou **nós adjacentes**) se existe um caminho de comprimento um entre eles.



- Na figura (a), B é adjacente a A; D é adjacente a E; no entanto, E não é adjacente a D.
- Na figura (b), E e D são adjacentes; já D e F não são adjacentes.

Grafos (Graphs)
Caminhos

- Um **caminho** (*path*) é uma sequência de vértices em que cada vértice é adjacente ao próximo.

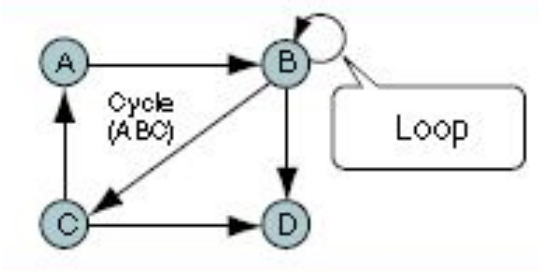


- Exemplos de caminhos: i) {A, B, C, E} ii) {A, B, E, F}
- Em ambos tipos de grafos, temos caminhos. No entanto, apenas em grafos não-direcionados podemos percorrer o caminho em qualquer direção.

Grafos (Graphs)

Ciclos e loops

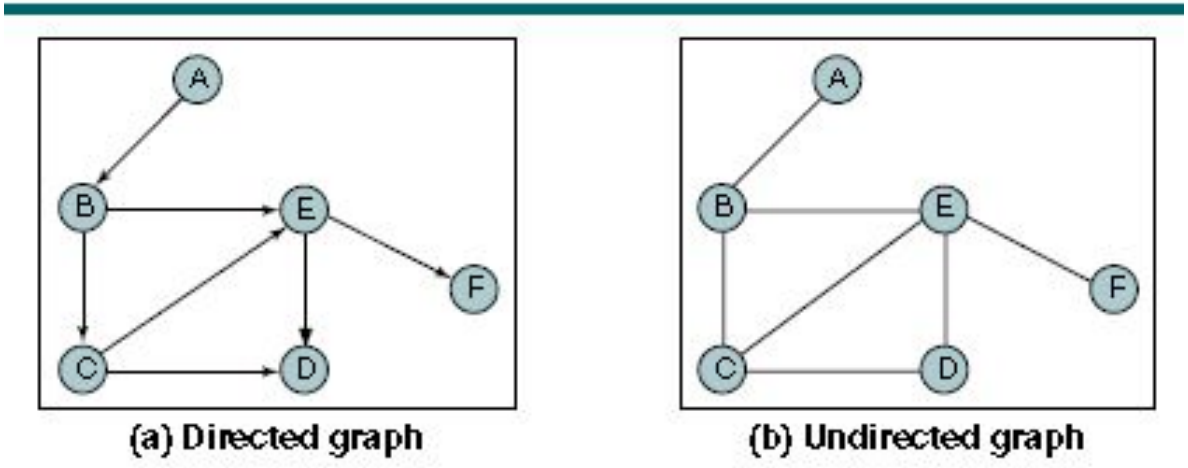
- Um **ciclo** é um caminho que começa e termina no mesmo vértice ou nó. Em princípio esse ciclo deve possuir pelo menos três vértices (nós). No entanto, podemos pensar em casos com dois e até apenas um vértices (nós).
- Um **loop** é um caso particular do ciclo no qual existe um único arco com inicio e fim no mesmo nó (uma única aresta com inicio e fim no mesmo vértice). Os dois extremos da linha são iguais.



- Exemplo de ciclo: i) {A, B, C, A}
- Exemplo de loop: i) {B, B}

Grafos (Graphs)

Ciclos e loops

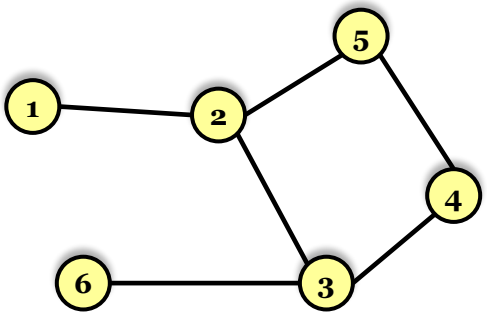


- Na figura (a), {B, C, E, B} não é um ciclo, porque ele não respeita a direção dos arcos.
- Na figura (b), {B, C, E, B} é um ciclo. Neste caso as arestas podem ser percorridas em qualquer direção.

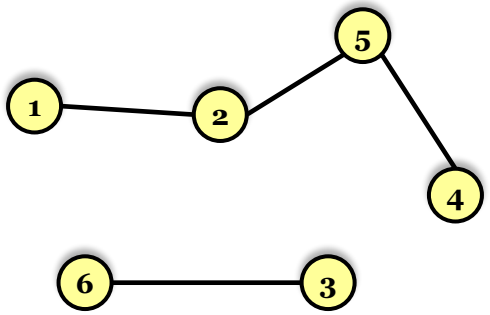
Tipos de Grafos

Grafo Conexo

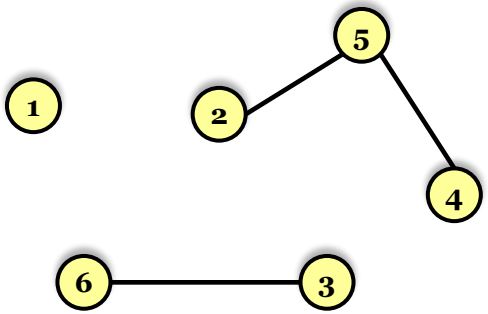
- Chama-se de grafo conexo todo grafo em que, para quaisquer dois vértices distintos, sempre existe um caminho que os une.
- Quando isso não acontece, temos um grafo desconexo. Um grafo desconexo contém no mínimo duas partes, cada uma delas chamada componente conexa.



Grafo Conexo



Grafo Desconexo 1



Grafo Desconexo 2

Grafos (Graphs)

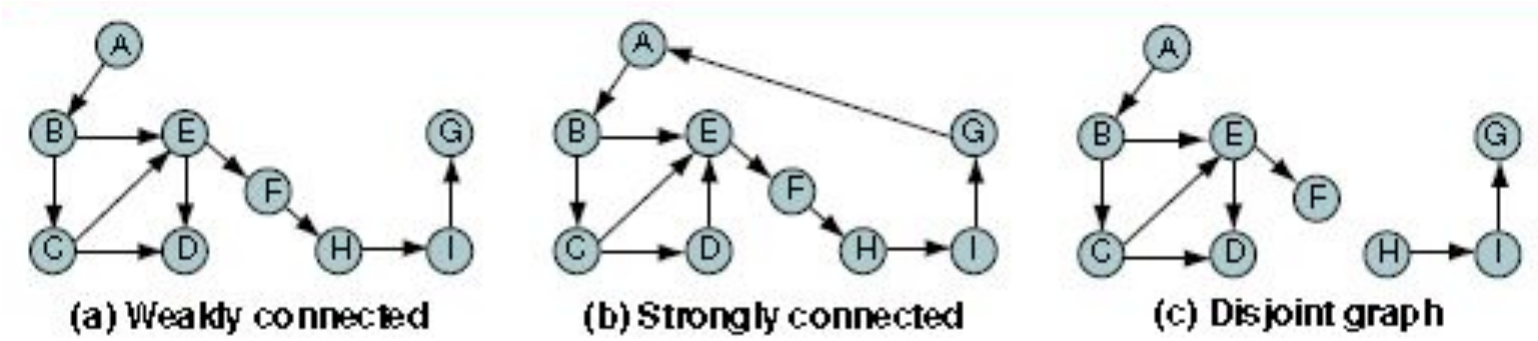
Grafos Conexos

- Dois vértices são ditos **conexos** (*connected*) se existe um caminho entre eles.
- Um grafo é dito **conexo** se, ignorando a direção, existe um caminho de qualquer vértice para qualquer outro vértice.
- Um grafo direcionado é **fortemente conexo** (*strongly connected*), se existe um caminho direcionado de cada vértice a cada outro vértice.
- Um grafo direcionado é **simplesmente conexo** (*weakly connected*), se não existe um caminho direcionado entre pelo menos um par de vértices.
- Um grafo é **desconexo** (*disjoint graph*) se não é conexo.

Grafos (Graphs)

Tipos de Conexidade

- Grafos Conexos e Não Conexos



Simplesmente Conexo

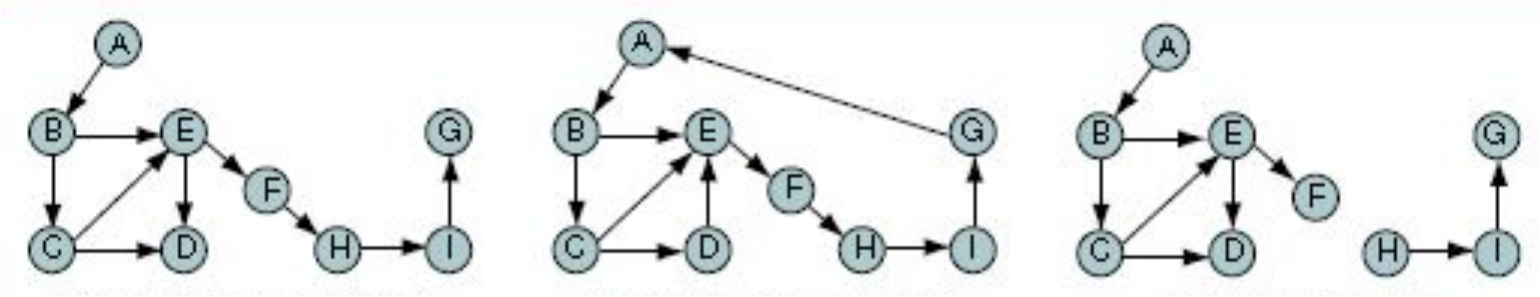
Fortemente Conexo

Desconexo

Grafos (Graphs)

Grau de um vértice (nó)

- O **grau** (*degree*) **de um vértice (**ou **nó)** é o número de linhas (arestas ou arcos) que incidem nele.
- Em grafos direcionados existem dois conceitos adicionais:
  - O **grau de saída** (*outdegree*) é o número de arcos que saem do nó.
  - O **grau de entrada** (in*degree*) é o número de arcos que entram no nó.
- O grau de um nó em grafos direcionados é igual a soma dos graus de saída e de entrada.



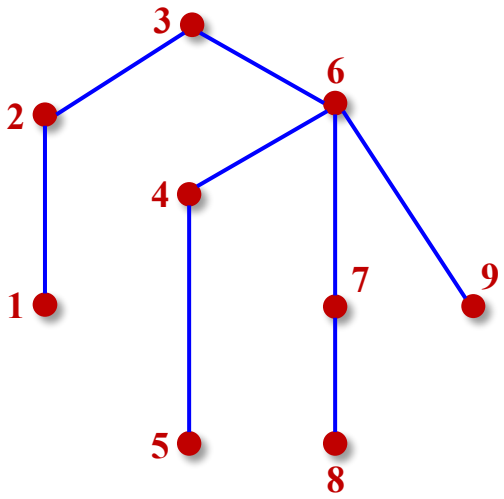
- Na figura (a), o nó B tem grau 3. O grau de entrada de B é 1, enquanto o grau de saída de B é 2.
- Na figura (b), o nó E tem grau 4. O grau de entrada de E é 3, enquanto o grau de saída de E é 1.

Grafos (Graphs)
Árvores são grafos

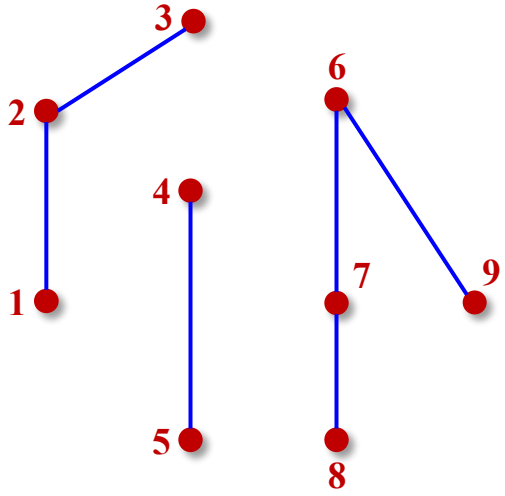
- Uma árvore é um tipo particular de grafo em que cada nó (ou vértice) possui somente um predecessor.
- Assim, toda árvore é um grafo, porém nem todo grafo é uma árvore.
- Alguns grafos possuem um ou mais árvores dentro deles os quais podem ser algoritmicamente determinados.

Arvores
Árvores são grafos

- Uma árvore é um tipo especial de grafo **não direcionado** que possui as seguintes características:
  - É conexo;
  - Não possui ciclos.
- Toda árvore permite conectar entre si um conjunto de vértices utilizando o menor número de arestas possíveis;
- Uma coleção de árvores é chamada de **floresta**.



Arvore G_1



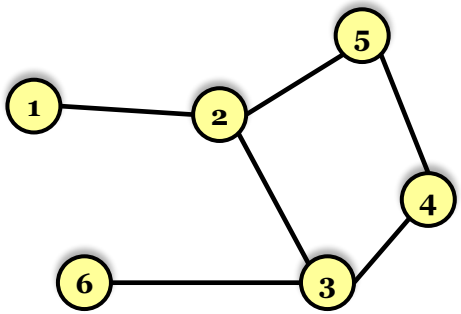
Floresta G_2

Tipos de Grafos
Definições

- Um grafo trivial é a forma mais simples de grafo que existe. Trata-se de um grafo que possui um único vértice e nenhuma aresta ou laço.

Grafo Trivial

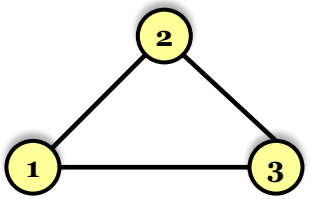
- Já um grafo simples é a forma mais comum de grafo que existe. Trata-se de um grafo não direcionado, sem laços e sem arestas paralelas.



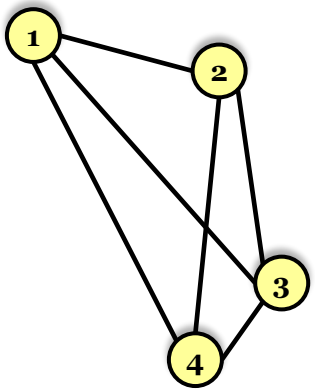
Grafo Simples

Tipos de Grafos
Grafo Completo

- Um grafo completo consiste em um grafo simples (ou seja, um grafo não direcionado, sem laços e sem arestas paralelas), onde cada vértice seu se conecta a todos os outros vértices do grafo.



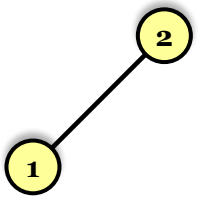
Grafo Completo



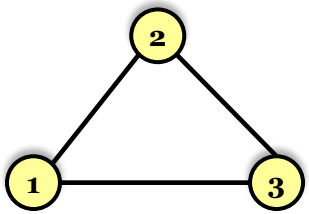
Grafo Completo

Tipos de Grafos
Grafo Regular

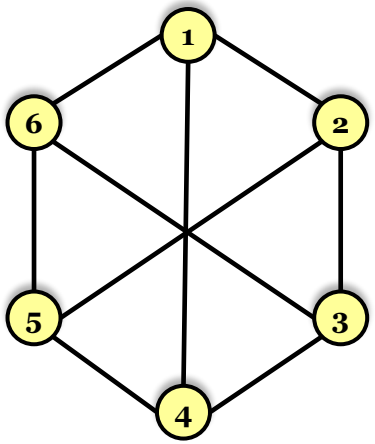
- Um grafo regular é um grafo em que todos os vértices possuem o mesmo grau (número de arestas ligadas a ele).



Grafo Regular
Grau 1



Grafo Regular
Grau 2

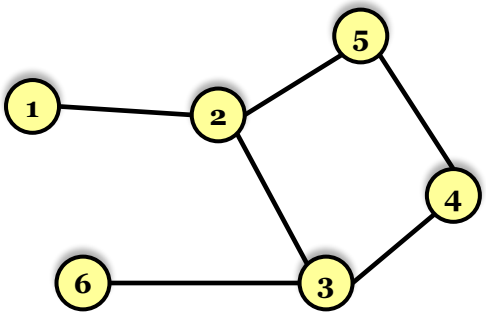


Grafo Regular
Grau 3

Tipos de Grafos

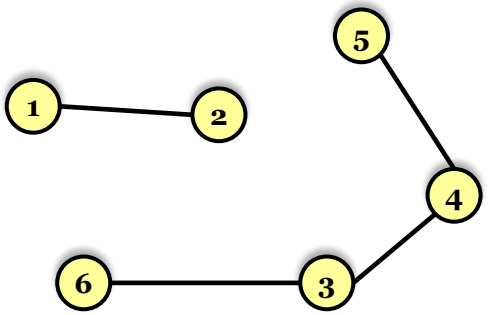
Subgrafo

- Dado um grafo $G(V, A)$, temos que o grafo $G_S\ (V_S, A_S)$, é um subgrafo de $G(V, A)$ se o conjunto de vértices V_S for um subconjunto de V, $V_S \subseteq V$, e se o conjunto de arestas A_S for um subconjunto de A, $A_S \subseteq A$.

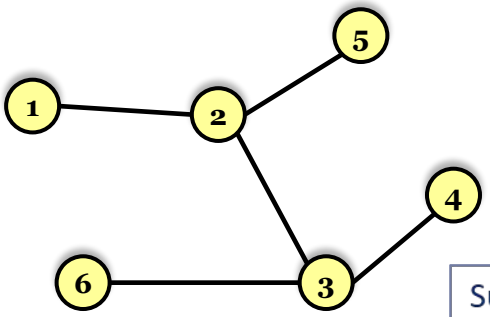


Grafo G

Subgrafo G_{S1}



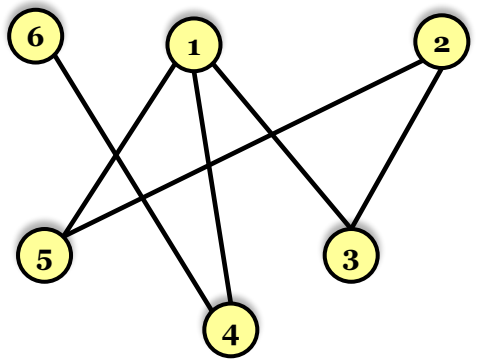
Subgrafo G_{S2}



Subgrafo G_{S3}

Tipos de Grafos
Grafo Bipartido

- Um grafo $G = (V, A)$ é chamado grafo bipartido se o seu conjunto de vértices puder ser dividido em dois subconjuntos V_1 e V_2 sem intersecção.
- De maneira que as arestas conectam apenas os vértices que estão em subconjuntos diferentes, ou seja, uma aresta sempre conecta um vértice de V_1 a V_2 ou vice-versa, porém ela nunca conecta vértices do mesmo subconjunto entre si.
- Podem se atribuir duas cores diferentes ao vértices, sem que existam arestas entre vértices de cores diferentes.
- Em um grafo bipartido, todo ciclo tem comprimento par.

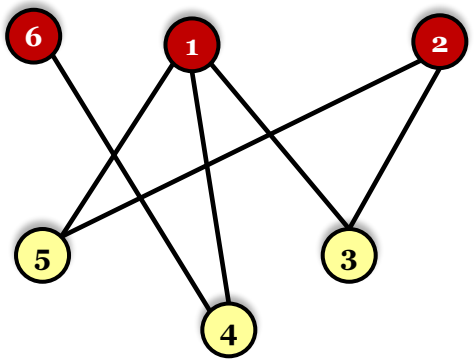


Grafo Bipartido

Tipos de Grafos

Definições

- Um grafo $G = (V, A)$ é chamado grafo bipartido se o seu conjunto de vértices puder ser dividido em dois subconjuntos V_1 e V_2 sem intersecção.
- De maneira que as arestas conectam apenas os vértices que estão em subconjuntos diferentes, ou seja, uma aresta sempre conecta um vértice de V_1 a V_2 ou vice-versa, porém ela nunca conecta vértices do mesmo subconjunto entre si.
- Podem se atribuir duas cores diferentes ao vértices, sem que existam arestas entre vértices de cores diferentes.
- Em um grafo bipartido, todo ciclo tem comprimento par.



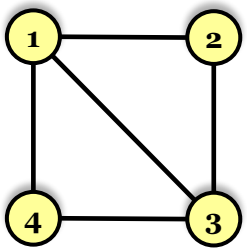
$V_1 = \{1, 2, 6\}$

$V_2 = \{3, 4, 5\}$

Grafo Bipartido

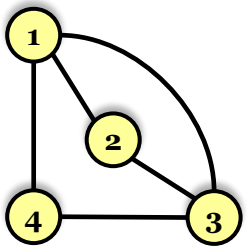
Tipos de Grafos
Grafos Isomorfos

- Dois grafos $G_1\ (V_1, A_1)$ e $G_2\ (V_2, A_2)$ são ditos isomorfos se existir uma função que faça o mapeamento de vértices e arestas de modo que os dois grafos se tornem coincidentes (idênticos em aparência).
- Em outras palavras, dois grafos são isomorfos se houver uma função f tal que, para cada dois vértices a e b adjacentes no grafo G_1 , $f(a)$ e $f(b)$ também sejam adjacentes no grafo G_2 .
- Condições necessárias para que dois grafos sejam isomorfos:
  - Possuírem o mesmo número de vértices;
  - Possuírem o mesmo número de arestas;
  - Possuírem o mesmo número de vértices com graus correspondentes.

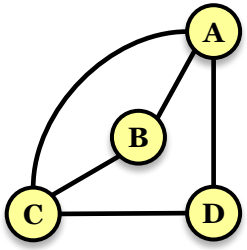


Grafo G_1

$f(1) = A$
$f(2) = B$
$f(3) = C$
$f(4) = D$



Grafo G_2



Grafo G_3

$f(1) = A$
$f(2) = B$
$f(3) = C$
$f(4) = D$

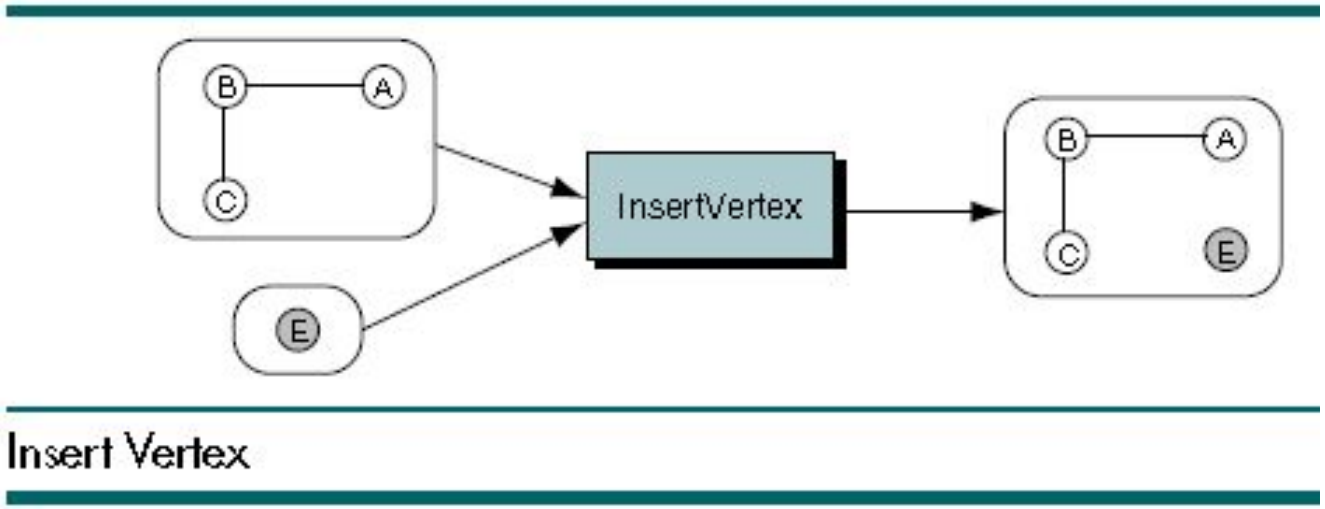
Grafos (Graphs)
Operações

- Definem-se seis operações primitivas de grafos que são necessárias para a manutenção de um grafo:
  - Inserção de um vértice (*insert vertex*);
  - Remoção de um vértice (*delete vertex*);
  - Adição de uma aresta (*add edge*);
  - Remoção de uma aresta (*delete edge*);
  - Buscar um vértice (*find vertex*);
  - Percorrer o grafo (*traverse graph*).

Grafos (Graphs)

Operações – Inserir um Vértice

- Insere um novo vértice ao grafo. Quando um vértice é inserido ele fica disjunto, não esta conectado aos outros vértices do grafo.

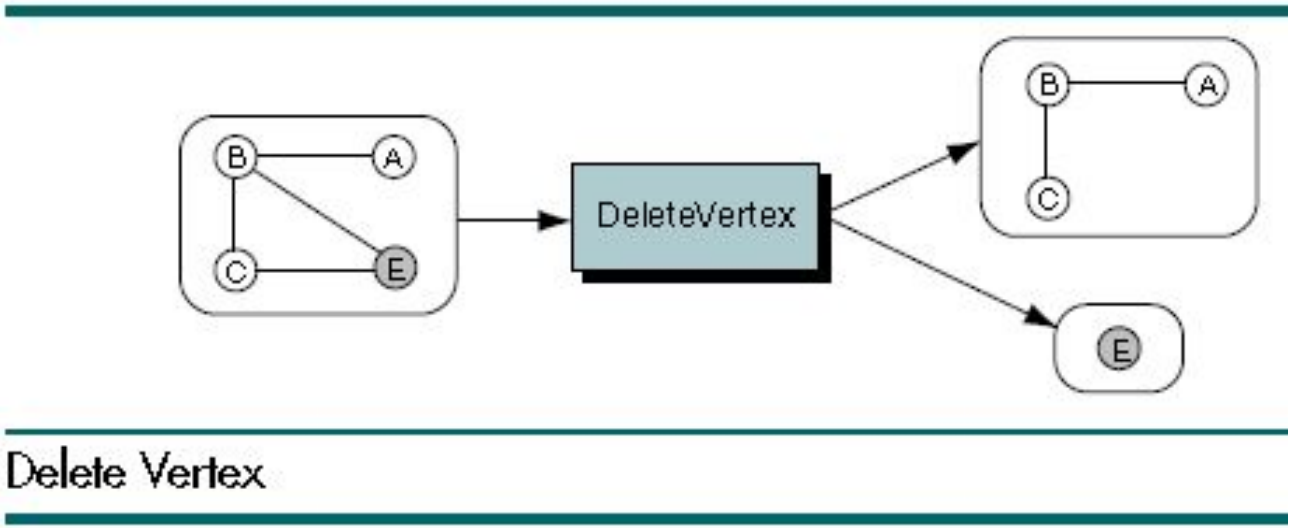


Insert Vertex

Grafos (Graphs)

Operações – Eliminar um Vértice

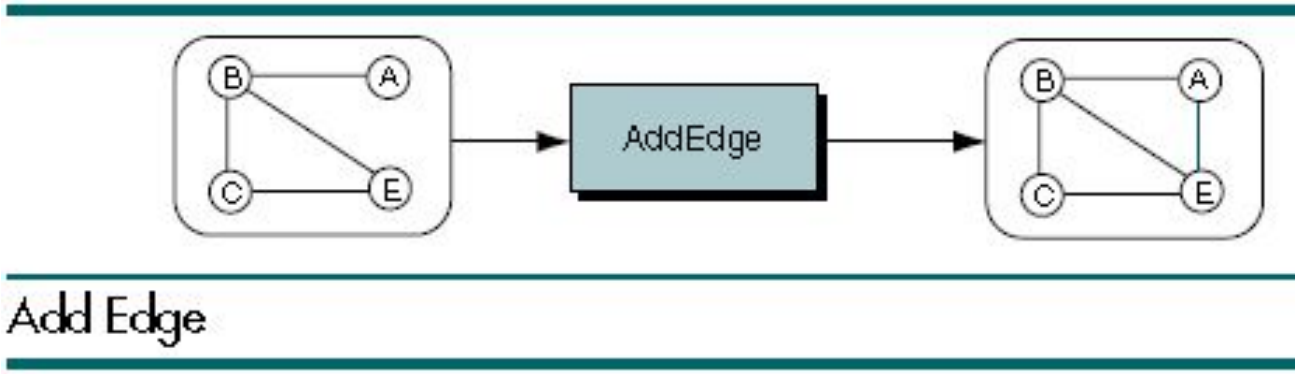
- Elimina um vértice do grafo. Quando um vértice é removido, todas as arestas que tem esse vértice com um de seus extremos, são também removidas.



Grafos (Graphs)

Operações – Adicionar uma Aresta

- Adicionar uma aresta conecta um vértice a outro vértice. Esta operação exige que dois vértices sejam especificados.
- Se o grafo for um digrafo, um dos vértices deve ser especificado como origem e o outro como destino.
- Se um vértice precisa de várias arestas, a operação de adição deve ser repetida para cada vértice adjacente.

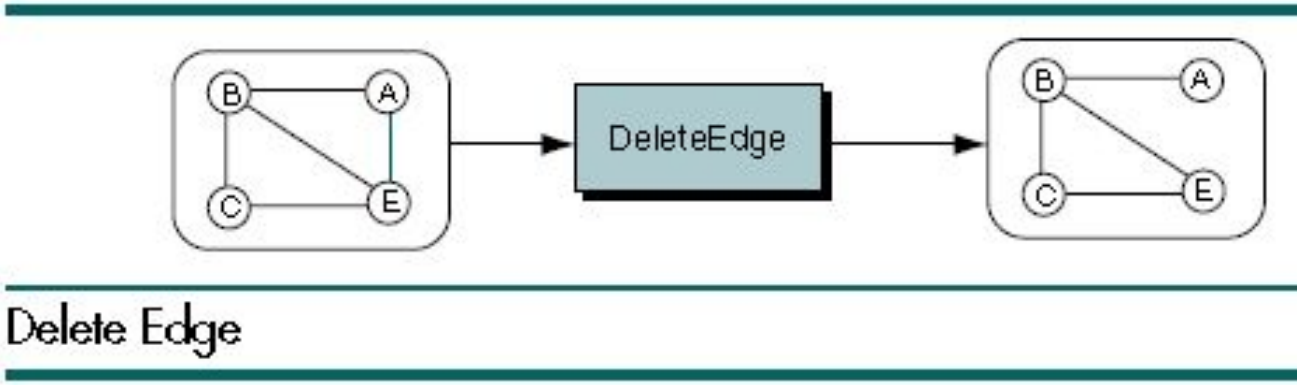


- A figura ilustra a inserção da aresta {A,E} ao grafo.

Grafos (Graphs)

Operações – Remover uma Aresta

- Esta operação remove uma aresta do grafo.

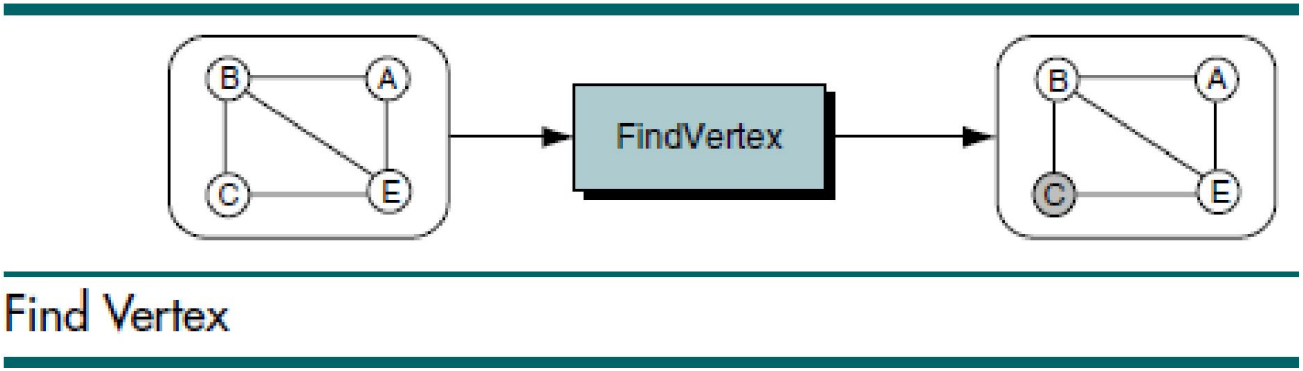


- A figura ilustra a remoção da aresta {A,E} do grafo.

Grafos (Graphs)

Operações – Buscar um vértice

- Esta operação percorre o grafo, buscando um vértice especifico. Se o vértice é encontrado seus dados são retornados. Caso contrário, retorna-se um indicador de erro.



Find Vertex

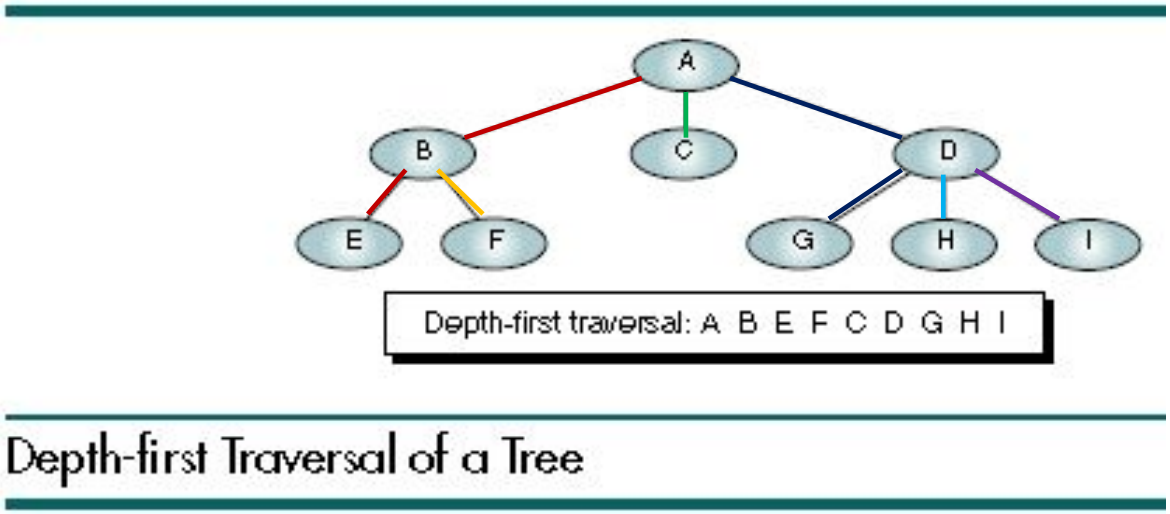
- A figura ilustra a busca pelo vértice C.

Grafos (Graphs)
Percurso em Grafos

- Percorrer um Grafo
  - Profundidade Primeiro (Depth-first)
  - Largura Primeiro (Bread-first)

Grafos (Graphs)
Percurso em Profundidade

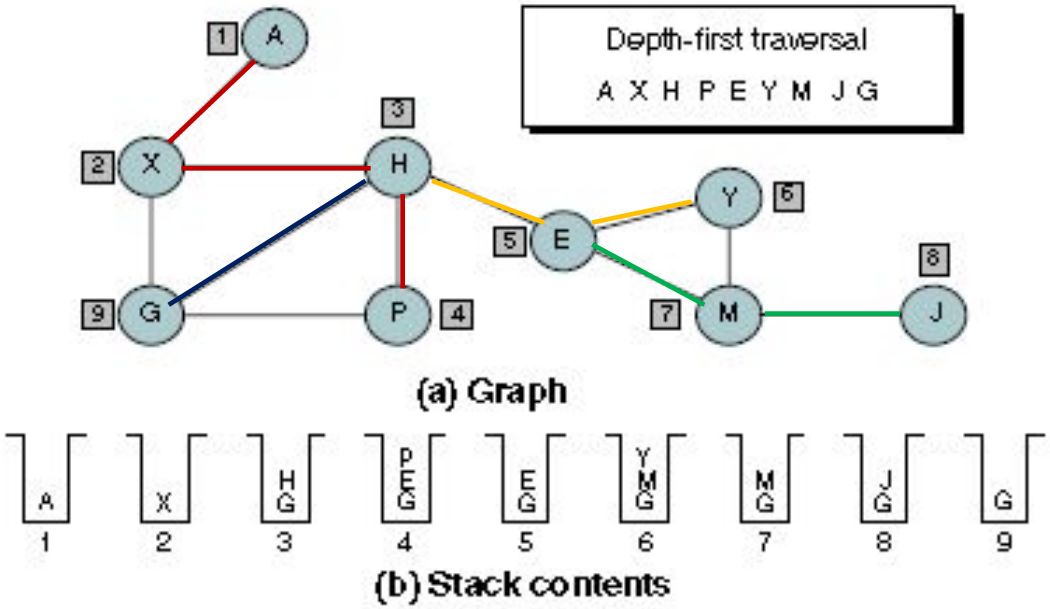
- No caso particular em que o grafo é uma árvore temos como ponto de partida a raiz.



Depth-first Traversal of a Tree

Grafos (Graphs)
Percurso em Profundidade

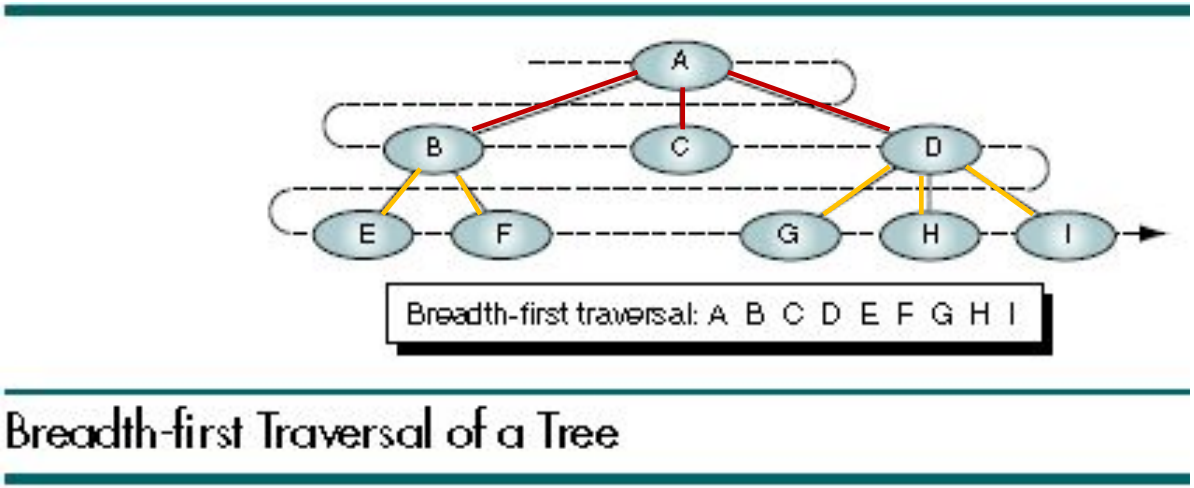
- Para percorrer um grafo em profundidade (Depth-first) fazemos uso de uma estrutura de pilha.
- Os vértices (ou nós) adjacentes são colocados na pilha assim que descobertos e processados de acordo com as regras dessa estrutura.



Depth-first Traversal of a Graph

Grafos (Graphs)
Percurso em Largura

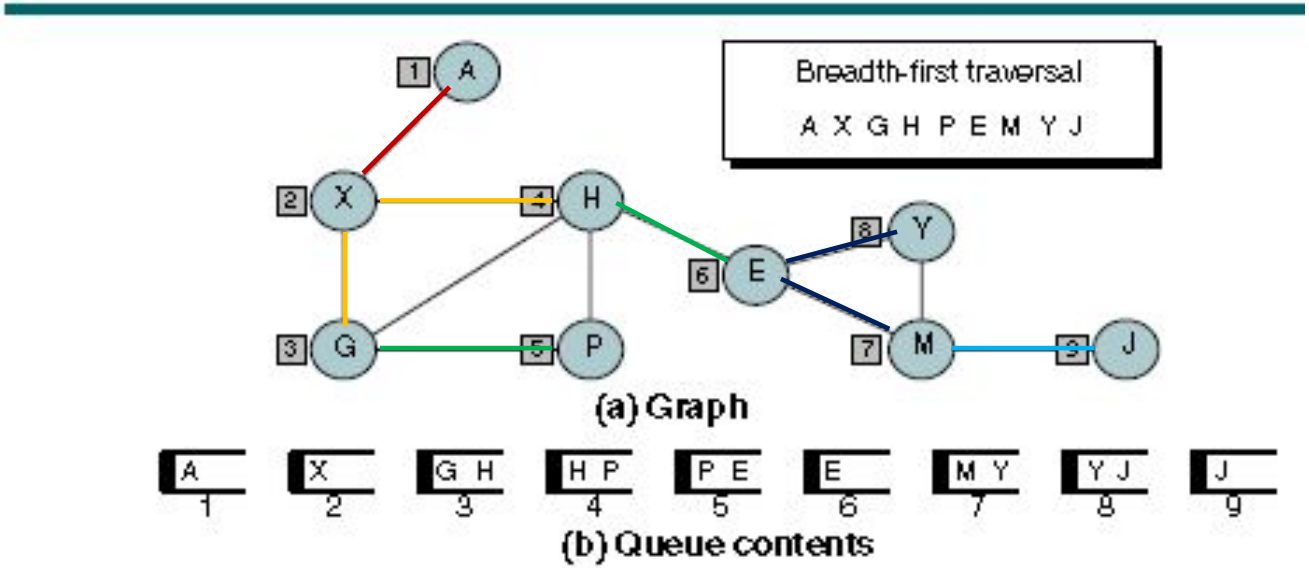
- No caso particular em que o grafo é uma árvore temos como ponto de partida a raiz.



Breadth-first Traversal of a Tree

Grafos (Graphs)
Percurso em Largura

- Para percorrer um grafo em largura (Depth-first) fazemos uso de uma estrutura de fila.
- Os vértices (ou nós) adjacentes são colocados na fila assim que descobertos e processados de acordo com as regras dessa estrutura.

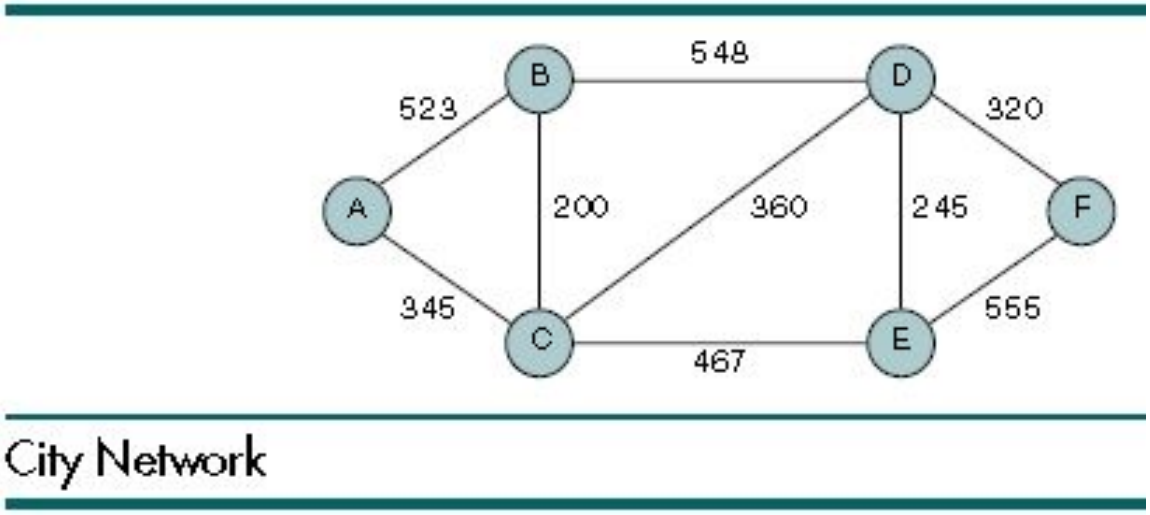


Breadth-first Traversal of a Graph

Redes (Networks)

Definição

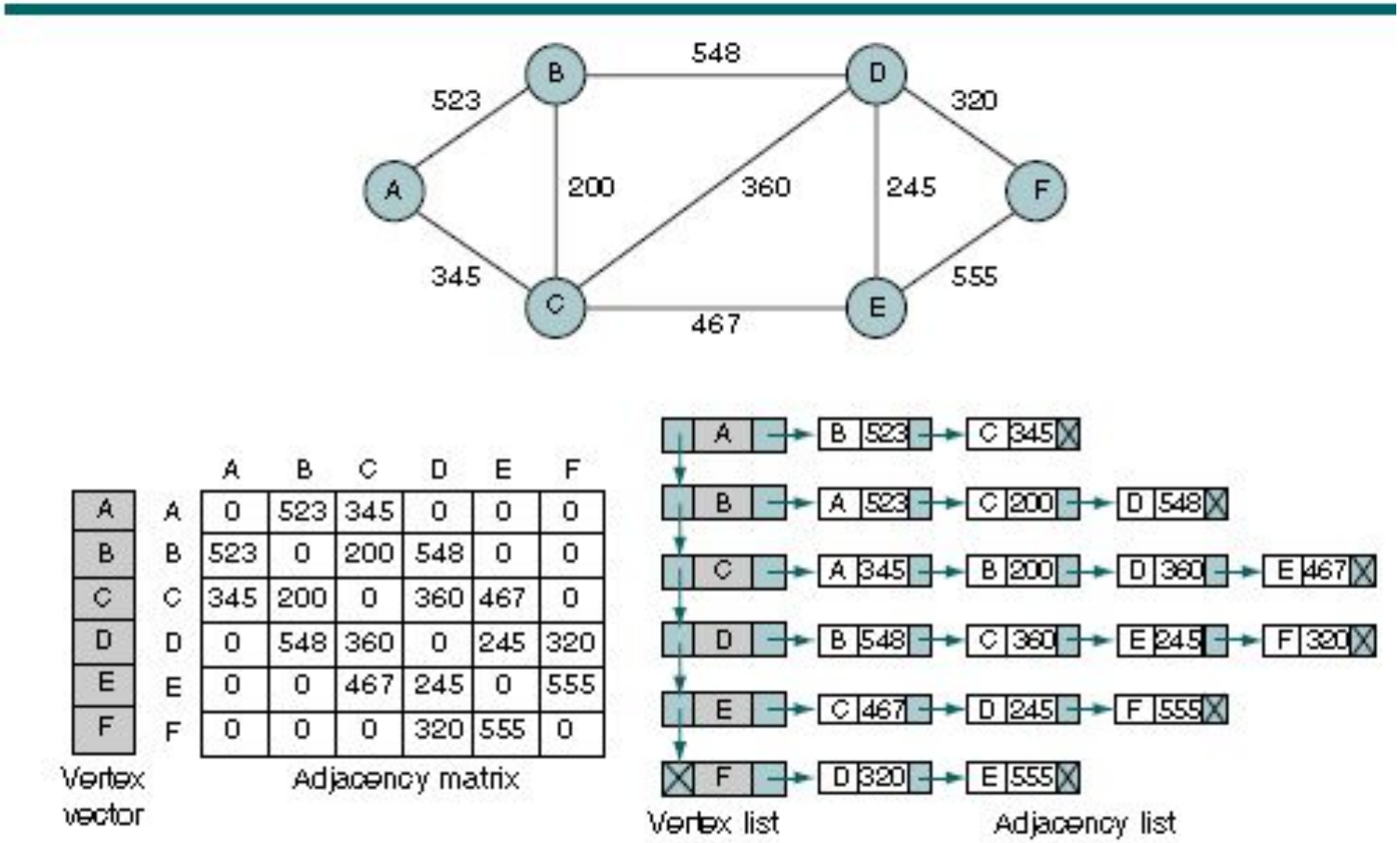
- Uma rede ou grafo ponderado é um grafo em que existe um ou mais atributos associados a suas conexões. Por exemplo, um atributo representando distância ou capacidade de uma conexão.



City Network

Redes (Networks)
Representação da Rede

- Representações possíveis de uma rede ou grafo ponderado.



Referências

- Gilberg, R.F. e Forouzan, B. A. Data Structures_A Pseudocode Approach with C. Capítulo 11. Graphs. Segunda Edição. Editora Cengage, Thomson Learning, 2005.