# **GRAFOS**

Ao concluir esta aula, o aluno deverá ser capaz de:

* compreender o que são grafos;
* conhecer as operações básicas que se podem efetuar com os grafos;
* tomar contato com algumas aplicações de grafos.

## **DEFINIÇÃO**

Um grafo é uma estrutura de dados bastante genérica, que engloba todas as estruturas estudadas anteriormente. É constituído por um conjunto de nós (vértices) e um conjunto de arcos (arestas). Cada arco é especificado por um par de nós (relação binária entre os nodos).

## **EXEMPLO** DE UM GRAFO ASSIMÉTRICO OU DIRIGIDO (DIGRAFO)

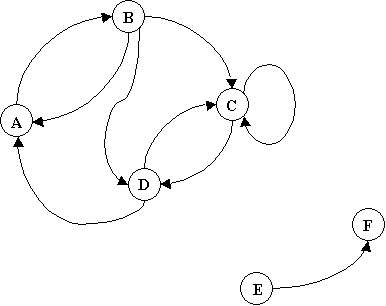
N ou V = {A, B, C, D, E, F}

A = {(A,B), (B,A), (B,C), (B,D), (C,C), (D,A), (D,C), (E,F)}

Este grafo pode ser representado pela figura abaixo:

MATRIZ DE ADJACÊNCIA

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| A |  | 1 |  |  |  |  |
| B | 1 |  | 1 | 1 |  |  |
| C |  |  | 1 | 1 |  |  |
| D | 1 |  | 1 |  |  |  |
| E |  |  |  |  |  | 1 |
| F |  |  |  |  |  |  |



## **USO PARA GRAFOS (representação)**

* mapa representando um conjunto de cidades ligadas por estradas. Pode-se desejar saber qual o menor caminho para ir de uma cidade a outra, entre outras coisas;
* circuito eletrônico (nós correspondem aos terminais dos dispositivos, e os arcos são os próprios dispositivos). Pode-se desejar saber se um terminal de um dispositivo está ligado a outro;
* representação de processos em Sistemas Paralelos ou Distribuídos (modelagem dinâmica de processos e sistemas);
* otimização de atividades em Projetos (nós representam as tarefas e os arcos indicam as relações de dependência entre elas). Pode-se desejar saber qual o prazo mínimo para execução de todo o projeto (identificação do caminho crítico).
* minimização de custos (identificação do custo mínimo).

## **TERMINOLOGIA**

Use como referência o grafo dado anteriormente.

* **Nó ou Nodo ou Vértice**: qualquer elemento do conjunto N;
* **Aresta ou Arco**: qualquer elemento do conjunto A;
* **Nós adjacentes (relação de adjacência ou de vizinho)**: são nós ligados por um arco: o nó A é adjacente ao nó D;
* **Arco incidente (relação de incidência)**: um arco é incidente em um nó quando leva a esse nó: o arco (C,D) é incidente em D;
* **Grau de um nodo**: é o número de arcos incidentes (arestas que chegam ao nodo e arestas que saem do nodo);
* **Caminho**: seqüência de um ou mais arcos: um caminho que leva o nó A ao nó C é o caminho dado por: < (A,B), (B,D), (D,C) >
* **Circuito (ciclo)**: é o caminho que leva ao mesmo nó do qual saiu: < (A,B), (B,D), (D,A) >
* **Laço**: é um circuito de um único arco: < (C,C) >
* **Subgrafo**: é o grafo formado por um subconjunto de nós de um grafo dado, contendo todos os arcos, cujas duas extremidades são nós desse subconjunto. Exemplo, o grafo abaixo é um subgrafo do grafo original;

**N’ =** {A, C, D, E}

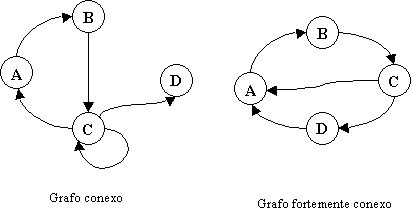
**A’ =** {(C,C), (D,A), (D,C), (C,D)}

* **Grafo parcial**: é o grafo formado por todos os nós de um grafo dado, contendo um subconjunto de seus arcos. Abaixo, apresenta-se um grafo parcial do grafo original.

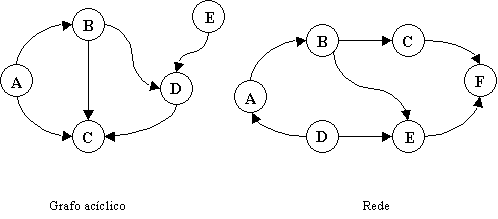
**N’ =** {A, B, C, D, E, F}

**A’ =** {(A,B), (B,C), (B,D), (E,F)}

* **Grafo conexo**: é um grafo no qual existe ao menos um nó a partir do qual existem caminhos para todos os outros nós (no exemplo abaixo, esse nó é o B);
* **Grafo fortemente conexo**: é um grafo no qual, para todos os nós, existem caminhos para todos ou outros nós;



* **Grafo acíclico**: é um grafo que não tem circuitos (ciclo);
* **Rede**: há muitos tipos de redes. Veremos a rede utilizada em sistemas PERT-COM (Program Evaluation and Review Technique – Critical Path Method): definida como um grafo acíclico, com nós especiais:
  + **Nó fonte:** a partir do qual se podem atingir todos os outros nós (nó A)
  + **Nó sorvedouro:** pode ser atingido a partir de qualquer outro nó (nó F)



* **Grafo não dirigido (SIMÉTRICO)**: é um grafo em que as relações entre os nós são simétricas (se existe um arco (A,B) então também existe um arco (B,A), e ambos são representados apenas por uma linha não orientada ligando A e B).Acesso aos nodos não possui uma direção indicada.
* **Grafo valorado:** grafo com valores associados aos nós ou aos arcos.

**Exercício:**

Dado um grafo (quadro), apresente os nós (vértices), arcos (arestas), grau incidências e adjacências de cada nodo, caminhos (ciclos), conexão, subgrafo.

## **CRITÉRIOS PARA PERCORRER GRAFOS**

Existem dois critérios mais utilizados para se percorrer grafos, a saber:

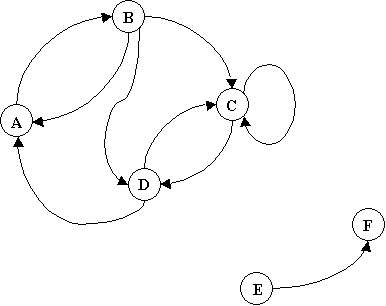
* Percurso em Amplitude (Largura ou Extensão);
* Percurso em Profundidade.

Independente do critério escolhido, para percorrer o grafo, parte-se de um nodo ***N*** escolhido arbitrariamente e “visita-se” este nodo ***N***. A seguir, considera-se cada um dos nodos ***NA*** adjacentes ao nodo ***N.***

1. no percurso em amplitude, para cada nodo ***NA***:
   * 1. visita-se o nodo ***NA*** e se marca este nodo como já visitado;
     2. coloca-se o nodo ***NA*** em uma fila;
     3. ao terminar de visitar os nodos ***NA***, toma-se o nodo que estiver na frente da fila e volta-se para o passo 1.
2. no percurso em profundidade, depois de encontrar o primeiro nodo ***NA*** de ***N***:
   * 1. visita-se o nodo ***NA*** e se marca este nodo como já visitado;
     2. coloca-se o nodo ***NA*** em uma pilha;
     3. faz-se ***N*** = ***NA***;
     4. se o novo ***N*** não possui nodos adjacentes que ainda não foram visitados, faz-se ***N*** = pilha(topo) e topo = topo – 1;
     5. volta-se ao passo 1.

###### **REPRESENTAÇÃO DE GRAFOS**

1. Matriz de Adjacências
2. Matriz de Incidências
3. Lista de Adjacências



MATRIZ DE ADJACÊNCIA

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| A |  | 1 |  |  |  |  |
| B | 1 |  | 1 | 1 |  |  |
| C |  |  | 1 | 1 |  |  |
| D | 1 |  | 1 |  |  |  |
| E |  |  |  |  |  | 1 |
| F |  |  |  |  |  |  |

matrizAdj

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |
| 0 |  | 1 |  |  | 1 |
| 1 | 1 |  | 1 |  |  |
| 2 |  |  |  | 1 |  |
| 3 | 1 |  |  |  |  |
| 4 |  |  |  | 1 |  |

DIGRAFO ou seja, assimétrico

Percurso/Caminho em Grafos

* AMPLITUDE ou LARGURA ou EXTENSÃO (algoritmo de força bruta)
  + Fila (Queue)

* + Vetor ou lista de visitados

0 1 2 3 4

1 1 1 1 1

* + Usar o vértice (printf, cont, if, …..)

4 3 0 1 2

==============================================

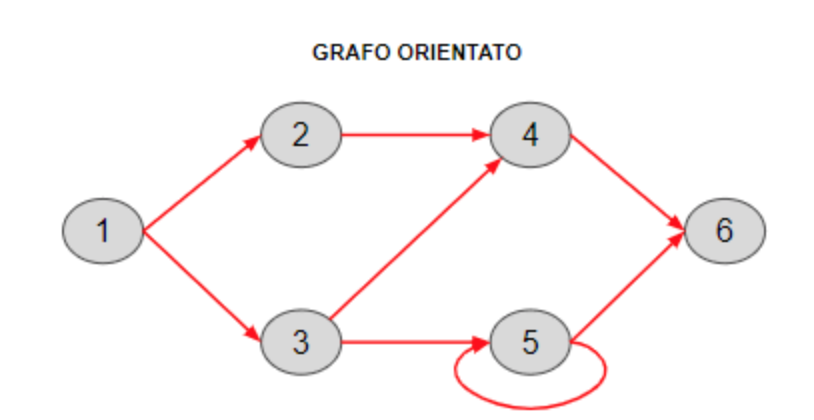
* PROFUNDIDADE (algoritmo de força bruta)
  + Pilha (Stack)

* + Vetor ou lista de visitados

0 1 2 3 4

0 0 0 0 0

* + Usar o vértice (printf, cont, if, …..)



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 |  | 1 | 1 |  |  |  |
| 2 |  |  |  | 1 |  |  |
| 3 |  |  |  | 1 | 1 |  |
| 4 |  |  |  |  |  | 1 |
| 5 |  |  |  |  | 1 | 1 |
| 6 |  |  |  |  |  |  |

* AMPLITUDE ou LARGURA ou EXTENSÃO (algoritmo de força bruta)
  + Fila (Queue)
  + Vetor ou lista de visitados

1 2 3 4 5 6

1 1 1 1 1 1

* + Usar o vértice (printf, cont, if, …..)

1 2 3 4 5 6

==============================================

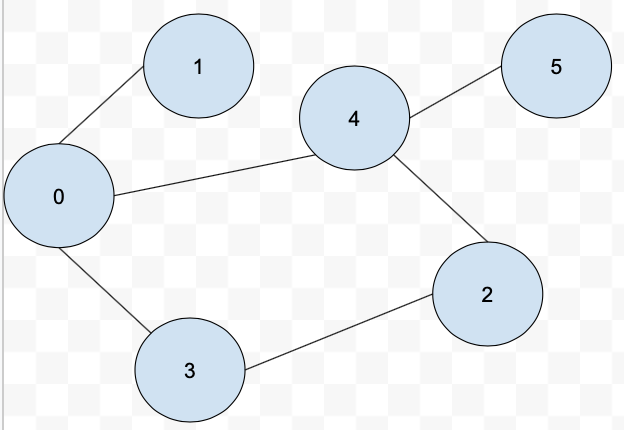
* PROFUNDIDADE (algoritmo de força bruta)
  + Pilha (Stack)

* + Vetor ou lista de visitados

1 2 3 4 5 6

0 0 0 0 0 0

* + Usar o vértice (printf, cont, if, …..)



3 0 2 1 4 5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| 0 |  | 1 |  | 1 | 1 |  |
| 1 | 1 |  |  |  |  |  |
| 2 |  |  |  | 1 | 1 |  |
| 3 | 1 |  | 1 |  |  |  |
| 4 | 1 |  | 1 |  |  | 1 |
| 5 |  |  |  |  | 1 |  |

matrizADJ[6][6]

qtdVertices = 6

qtdArestas = 12

* PERCURSO AMPLITUDE/LARGURA/EXTENSÃO
  + Fila
* Visitados (vetor ou uma lista)

0 1 2 3 4 5

0 0 0 0 0 0

* Visitar o vértice/nodo/no (print)

==============================================

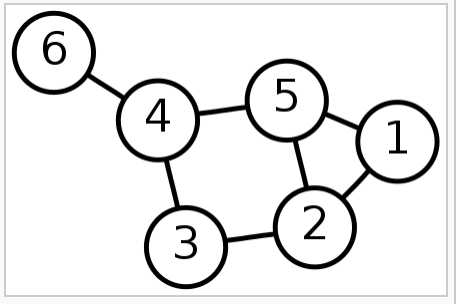
* PROFUNDIDADE (algoritmo de força bruta)
  + Pilha (Stack)

* + Vetor ou lista de visitados

0 1 2 3 4

0 0 0 0 0

* + Usar o vértice (printf, cont, if, …..)



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 |  | 1 |  |  | 1 |  |
| 2 | 1 |  | 1 |  | 1 |  |
| 3 |  | 1 |  | 1 |  |  |
| 4 |  |  | 1 |  | 1 | 1 |
| 5 | 1 | 1 |  | 1 |  |  |
| 6 |  |  |  | 1 |  |  |

matrizADJ

qtdVertices = 6

qtdArestas = 14

==============================================

* PROFUNDIDADE (algoritmo de força bruta)
  + Pilha (Stack)

* + Vetor ou lista de visitados

0 1 2 3 4

0 0 0 0 0

* + Usar o vértice (printf, cont, if, …..)