

# Teoria dos Grafos

## Aula 3

### Aula passada

- Exemplos
- Definições
- Algumas propriedades

### Aula de hoje

- Representando grafos
- Matriz e lista

# Grafo

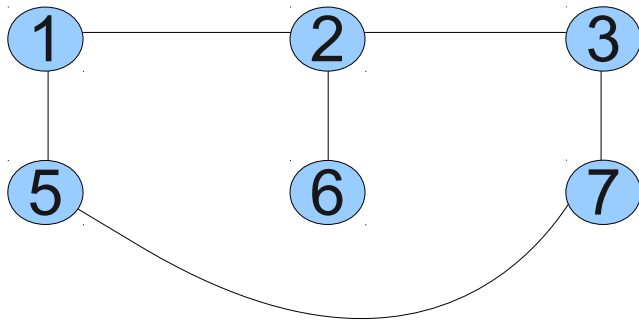
- Grafo  $G=(V, E)$

- $V$  = conjunto de vértices (inteiros)
- $E$  = conjunto de arestas (pares não-ordenados)

Representação  
matemática  
de grafos

- Exemplo

- $V = \{1, 2, 3, 5, 6, 7\},$
- $E = \{(1,2), (1,5), (2,3), (2,6), (3,7), (5,7)\}$



**Como representar  
no computador?**

# Representando Grafos

- Como representar grafos no computador?

## Estrutura de dados

- Duas estruturas fundamentais
  - matriz
  - lista
- Qual é a estrutura mais adequada (ou mais eficiente)?

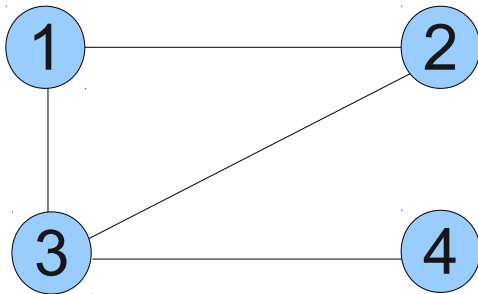
**Depende do algoritmo!**

# Representação via Matriz

- Como representar utilizando matrizes?
- **Idéia:** associar vértices à linhas e colunas da matriz
  - elemento da matriz indica se há aresta
- **Matriz de adjacência**
- Matriz  $n \times n$  ( $n$  é número de vértices)
  - $a_{ij} = 1$  , se existe aresta entre vértices  $i$  e  $j$
  - $a_{ij} = 0$  , caso contrário.

# Matriz de Adjacência

## ■ Exemplo



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

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

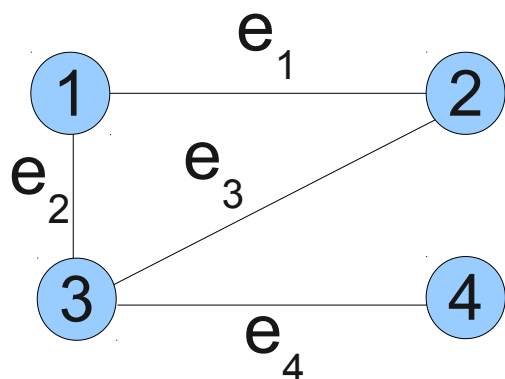
?

# Matriz de Incidência

- **Idéia:** associar vértices às linhas e arestas às colunas
  - elemento da matriz indica se aresta incide sobre o vértice
- **Matriz de incidência**
- Matriz  $n \times m$  ( $n$  vértices,  $m$  arestas)
  - $a_{ij} = 1$  , se vértice  $i$  incide sobre aresta  $j$
  - $a_{ij} = 0$  , caso contrário.

# Matriz de Incidência

## ■ Exemplo



	e <sub>1</sub>	e <sub>2</sub>	e <sub>3</sub>	e <sub>4</sub>
1	1	1	0	0
2	1	0	1	0
3	0	1	1	1
4	0	0	0	1

	e <sub>1</sub>	e <sub>2</sub>	e <sub>3</sub>	e <sub>4</sub>	e <sub>5</sub>	e <sub>6</sub>
1	1	1	0	0	0	1
2	1	0	1	1	1	0
3	0	1	0	1	0	0
4	0	0	0	0	1	0
5	0	0	1	0	0	1

→ ?

# Desvantagem

- Desvantagem da representação matricial?
- Considere grafos grandes e esparços
  - *grande*: muitos vértices
  - *esparço*: relativamente poucas arestas
- Matriz formada principalmente de zeros!

**Grande consumo de  
memória (desnecessário)!**

- Como resolver este problema?

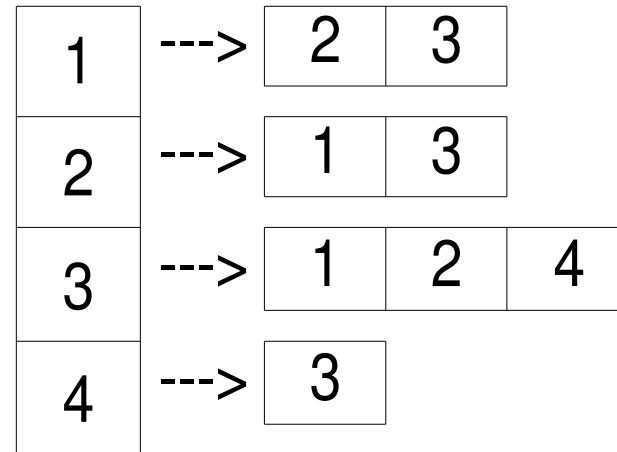
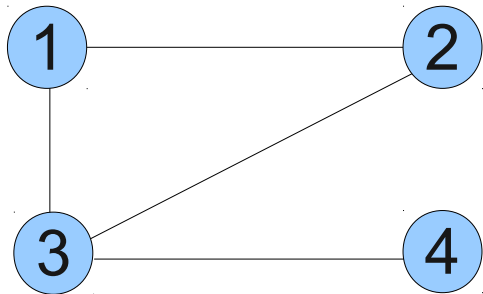


# Representação via Listas

- **Idéia:** associar a cada vértice uma lista de vértices adjacentes
- **Lista de adjacência**
- Vértices associados a um vetor, dimensão  $n$  (número de vértices no grafo)
- Cada vértice possui uma lista de vértices adjacentes

# Lista de Adjacência

## ■ Exemplo



# Desvantagem

- Desvantagem da representação com lista?
- Considere grafos onde vértices tem muitos vizinhos (mas bem menos do que  $n$ )
- Listas vão ser grandes (longas)
- Problema?

**Tempo de acesso! Ex. descobrir se dois vértices são vizinhos**

# Vantagens/Desvantagens

■ Tempo de execução	Matriz	Lista
■ Inserir aresta?	$O(1)$	$O(1)$
■ Remover aresta?	$O(1)$	$O(g_{\max})$
■ Testar adjacência ( $v_1$ e $v_2$ são vizinhos)?	$O(1)$	$O(g_{\max})$
■ Listar vizinhos de $v$ ?	$O(n)$	$O(g_{\max})$

**Melhor estrutura depende  
do algoritmo!**