

# Representação computacional de grafos

---

TEG0001 - Teoria dos grafos  
Prof. Dr. Ricardo José Pfitscher  
[ricardo.pfitscher@gmail.com](mailto:ricardo.pfitscher@gmail.com)



# Leitura complementar

- IME-USP:

[https://www.ime.usp.br/~pf/algoritmos\\_para\\_grafos/aulas/graphs.html](https://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/graphs.html)

- Conseguiram ler?

# Exercício - entregar no moodle

1. Acesse a ferramenta do google maps e encontre o endereço de sua residência ou um local conhecido por você
2. Faça uma captura de tela e marque os encontros entre ruas (cruzamentos) como sendo vértices de um grafo
3. Use a ferramenta de medição do google para medir a distância entre esses cruzamentos
4. Represente esse grafo na forma matemática e indique:
  - a. Qual o maior grau desse grafo?
  - b. Qual o tipo de grafo?

Três alunos sorteados para explicar sua resposta...

# Grafos - representação

- Como representar computacionalmente um grafo?
  - Precisamos armazenar os dados essenciais da definição de grafos
  - Adjacências entre vértices, vértices, arestas, etc...

# Grafos - representação

- Como representar computacionalmente um grafo?
  - Precisamos armazenar os dados essenciais da definição de grafos
  - Adjacências entre vértices, vértices, arestas, etc...
- Implementações mais utilizadas
  - Matriz de adjacências
  - Lista de adjacências

Também existe a matriz de incidências

# Grafos - matriz de adjacências

- Conceito de adjacência:
  - $u$  é adjacente a  $v$ , se  $u$  está conectado a  $v$
- Dado um grafo  $G$ , a matriz de adjacências  $r=(r_{ij})$  é uma matriz  $n \times n$  onde:

$$r_{ij} = \begin{cases} 1, & \text{se } i \text{ é adjacente a } j \\ 0, & \text{em caso contrário} \end{cases}$$

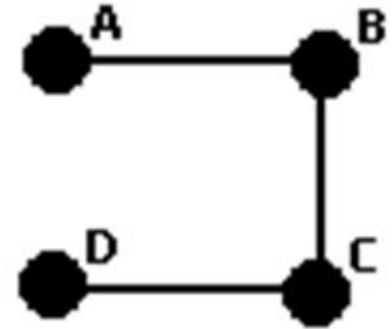
- Também pode ser utilizado o peso da aresta
- Em grafos direcionados utiliza-se valor positivo se a aresta sai do nó e valor negativo se a aresta incide sobre o nó

# Grafos - matriz de adjacências

- Como o seguinte grafo pode ser representado com uma matriz de adjacências?

| X | A | B | C | D |
|---|---|---|---|---|
| A |   |   |   |   |
| B |   |   |   |   |
| C |   |   |   |   |
| D |   |   |   |   |

$$r_{ij} = \begin{cases} 1, & \text{se } i \text{ é adjacente a } j \\ 0, & \text{em caso contrário} \end{cases}$$

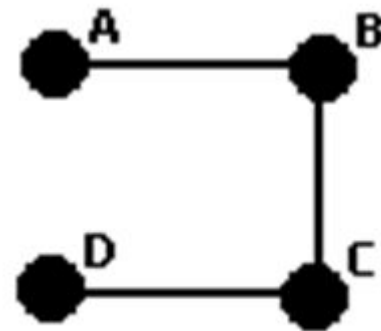


# Grafos - matriz de adjacências

- Como o seguinte grafo pode ser representado com uma matriz de adjacências?

| X | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 1 | 0 | 0 |
| B | 1 | 0 | 1 | 0 |
| C | 0 | 1 | 0 | 1 |
| D | 0 | 0 | 1 | 0 |

$$r_{ij} = \begin{cases} 1, & \text{se } i \text{ é adjacente a } j \\ 0, & \text{em caso contrário} \end{cases}$$



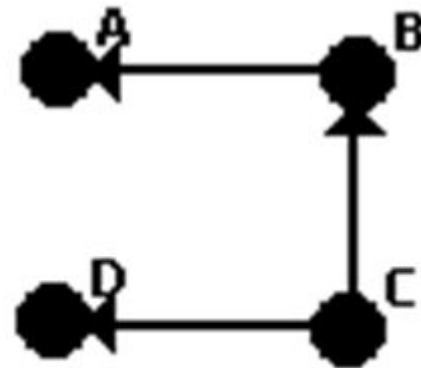


# Grafos - matriz de adjacências

- Como o seguinte dígrafo pode ser representado com uma matriz de adjacências?

| X | A | B | C | D |
|---|---|---|---|---|
| A |   |   |   |   |
| B |   |   |   |   |
| C |   |   |   |   |
| D |   |   |   |   |

$$r_{ij} = \begin{cases} 1, & \text{se } i \text{ é adjacente a } j \\ 0, & \text{em caso contrário} \end{cases}$$

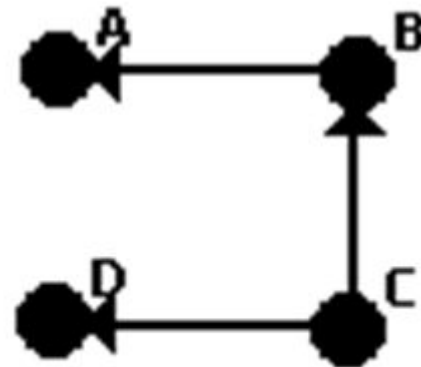


# Grafos - matriz de adjacências

- Como o seguinte dígrafo pode ser representado com uma matriz de adjacências?

| X | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 0 | 0 | 0 |
| B | 1 | 0 | 0 | 0 |
| C | 0 | 1 | 0 | 1 |
| D | 0 | 0 | 0 | 0 |

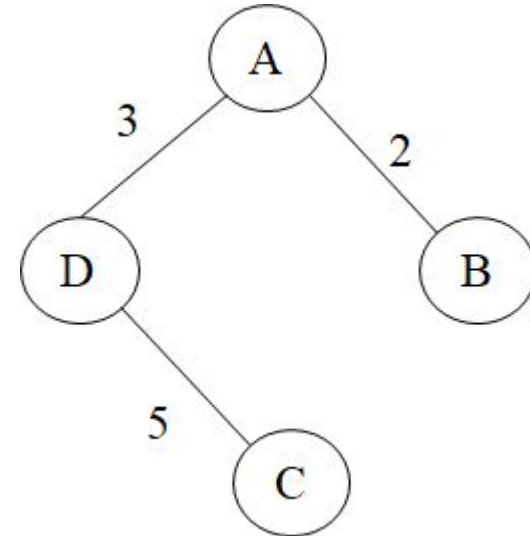
$$r_{ij} = \begin{cases} 1, & \text{se } i \text{ é adjacente a } j \\ 0, & \text{em caso contrário} \end{cases}$$



# Grafos - matriz de adjacências

- Como o seguinte grafo **ponderado** pode ser representado com uma matriz de adjacências?

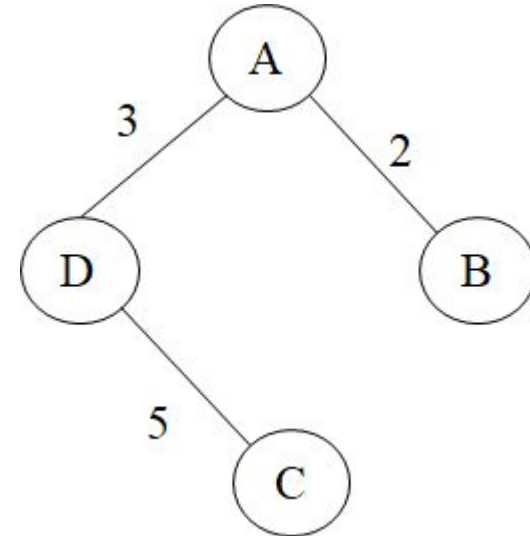
| X | A | B | C | D |
|---|---|---|---|---|
| A |   |   |   |   |
| B |   |   |   |   |
| C |   |   |   |   |
| D |   |   |   |   |



# Grafos - matriz de adjacências

- Como o seguinte grafo **ponderado** pode ser representado com uma matriz de adjacências?

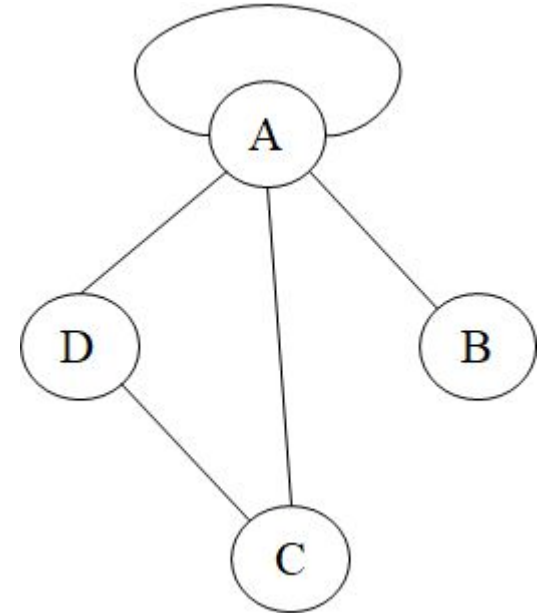
| X | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 2 | 0 | 3 |
| B | 2 | 0 | 0 | 0 |
| C | 0 | 0 | 0 | 5 |
| D | 3 | 0 | 5 | 0 |



# Grafos - matriz de adjacências

- Como o seguinte grafo **com laço** pode ser representado com uma matriz de adjacências?

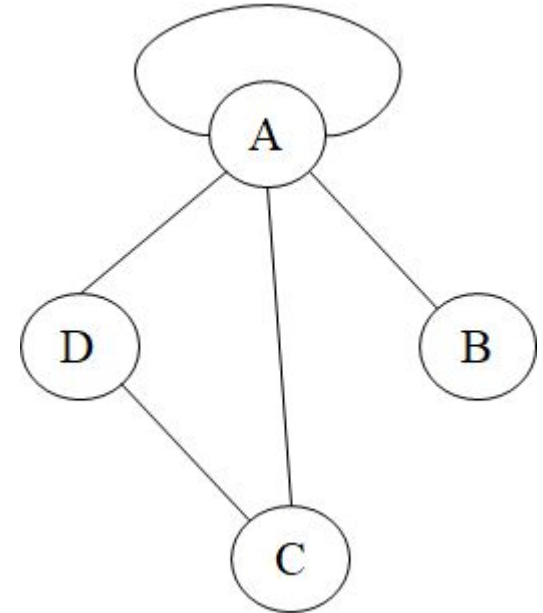
| X | A | B | C | D |
|---|---|---|---|---|
| A |   |   |   |   |
| B |   |   |   |   |
| C |   |   |   |   |
| D |   |   |   |   |



# Grafos - matriz de adjacências

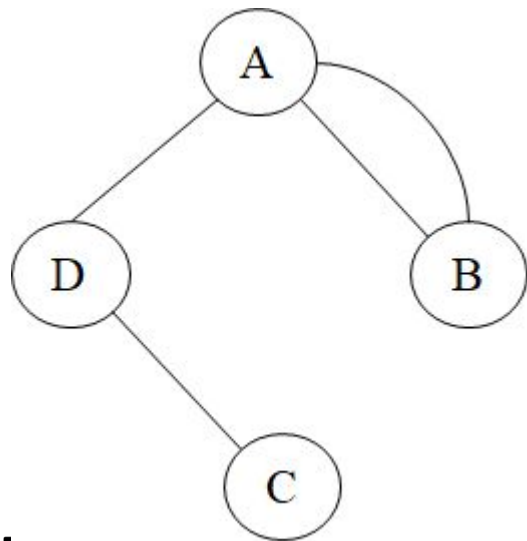
- Como o seguinte grafo **com laço** pode ser representado com uma matriz de adjacências?

| X | A | B | C | D |
|---|---|---|---|---|
| A | 1 | 1 | 1 | 1 |
| B | 1 | 0 | 0 | 0 |
| C | 1 | 0 | 0 | 1 |
| D | 1 | 0 | 1 | 0 |



# Grafos - matriz de adjacências

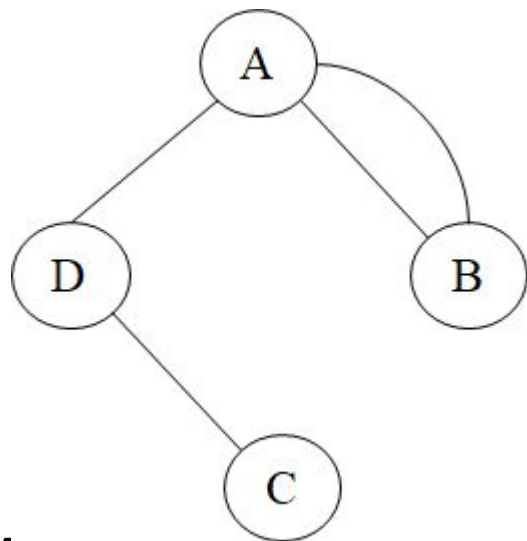
- Como o seguinte grafo **com arestas paralelas** pode ser representado com uma matriz de adjacências?



| X | A | B | C | D |
|---|---|---|---|---|
| A |   |   |   |   |
| B |   |   |   |   |
| C |   |   |   |   |
| D |   |   |   |   |

# Grafos - matriz de adjacências

- Como o seguinte grafo **com arestas paralelas** pode ser representado com uma matriz de adjacências?



| X | A | B | C | D |
|---|---|---|---|---|
| A | 0 | ? | 0 | 1 |
| B | ? | 0 | 0 | 0 |
| C | 0 | 0 | 0 | 1 |
| D | 1 | 0 | 1 | 0 |



# Grafos - matriz de adjacências

- Vantagens
  - Fácil visualização para vértices adjacentes
  - Útil em algoritmos que precisamos de rapidez para saber se existe conexão entre dois vértices
    - Acesso pertence à  $O(1)$
  - Cálculo do grau do nó
    - Grafos não direcionados: Soma dos números de uma linha
    - Grafos direcionados:
      - Soma da linha: grau de saída
      - Soma da coluna: grau de entrada
- Desvantagem:
  - Complexidade de espaço,  $O(n^2)$



# Grafo - Matriz - implementação - Python

- Vamos estabelecer uma Classe para armazenar a matriz de adjacências e uma lista de vértices (rótulos)

```
class Graph():  
  
    def __init__(self, nvertices):  
        self.N = nvertices  
        self.graph = [[0 for column in range(nvertices)]  
                       for row in range(nvertices)]  
        self.V = ['0' for column in range(nvertices)]  
  
print('Qual o número de vértices?')  
n = int(input())  
g = Graph(n)  
print(g.graph)
```

# Grafo - Matriz - implementação - Python

- Modifique a classe Grafo e adicione um método para nomear os vértices do grafo

# Grafo - Matriz - implementação - Python

- Modifique a classe Grafo e adicione um método para nomear os vértices do grafo

```
#...
def nameVertex(self):
    for i in range(self.N):
        print("Qual o rotúlo do vértice %i?"%(i))
        self.V[i]=input()
#...
```

# Grafo - Matriz - implementação - Python

- Modifique a classe Grafo e adicione um método para ler o peso das adjacências entre os vértices

# Grafo - Matriz - implementação - Python

- Modifique a classe Grafo e adicione um método para ler o peso das adjacências entre os vértices

```
#...
def setEdge(self,u,v,w):
    self.graph[u][v]=w

def loadEdges(self):
    for i in range(self.N):
        for j in range(self.N):
            if i!=j:
                print("Qual o peso entre %c e %c?"%(
                    self.V[i],self.V[j]))
                self.setEdge(i,j,input())
```

# Exercícios

1. Implemente computacionalmente o grafo do exercício de mapeamento de bairro que você fez na aula passada.
2. Implemente métodos para:
  - a. verificar se dois vértices são adjacentes e a distância entre eles
  - b. retornar o grau de um dado vértice
  - c. verificar se um grafo  $a$  é subgrafo do grafo  $b$
  - d. retornar o número de arestas do grafo
  - e. verificar se o grafo é completo