

grafos

July 21, 2022

Desenvolvimento Web (Front-End) > Conteúdo

Atualizado em: 23/04/2022

1 Conceitos Básicos

Prof. Eduardo Ono

1.1 Grafos

DEFINIÇÃO (*grafo*). Um *grafo* G pode ser definido como um par de conjuntos finitos (V, E) e denotado por $G = (V, E)$, onde V é um conjunto não vazio de *vértices* (ou *nós*) e E um conjunto de *arestas*, que pode ou não ser vazio, sendo cada aresta um conjunto de dois vértices a partir de V .

Um grafo G também pode ser denotado por $G = (V(G), E(G))$, onde $V = V(G)$ é o conjunto de vértices (ou nós) e $E = E(G)$ o conjunto de arestas (*edges* em inglês).

O conjunto de arestas pode ser representado por $E = \{e_1, e_2, \dots, e_m\}$. Cada aresta $e_i \in E(G)$ também pode ser representado como um par não ordenado (v_i, v_j) , $1 \leq i \leq n$ e $i \leq j \leq n$.

Dada uma aresta $(v_i, v_j) \in E(G)$, o vértice v_i é dito **extremidade inicial** e v_j é dito **extremidade final**.

1.1.1 Exemplo

O grafo $G = (V, E)$ tendo como conjunto de vértices $V = \{A, B, C, D, E\}$ e aretas $E = \{AB, BC, CD, DE, EA\}$ possui a seguinte representação gráfica:

```
[ ]: import networkx as nx
import matplotlib.pyplot as plt

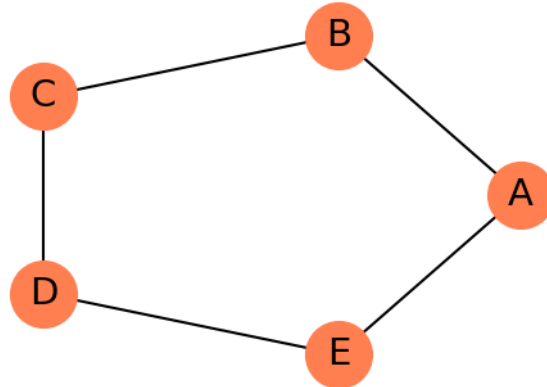
options = {
    'node_size': 600,
    'node_color': 'coral',
    'font_size': 14,
```

```

    'with_labels': True
}

E = [('A', 'B'), ('B', 'C'), ('C', 'D'), ('D', 'E'), ('E', 'A')]
G = nx.Graph()
G.add_edges_from(E)
plt.figure(figsize=[3,2], dpi=150)
nx.draw_circular(G, **options)

```



1.2 Terminologias

Duas arestas são **adjacentes** quando compartilham um mesmo vértice.

DEFINIÇÃO (subgrafo). Um **subgrafo** de um grafo $G = (V, E)$ é um subconjunto $G' = (V', E')$ tal que $V' \subseteq V$ e $E' \subseteq E$.

Por definição, o próprio grafo G é subgrafo dele próprio.

DEFINIÇÃO. O **grau de um vértice** $v \in V(G)$, denotado por d_v ou $\deg v$, é o número de arestas que incidem sobre v :

$$d(v) = \sum_{j \in V} w_{ij}$$

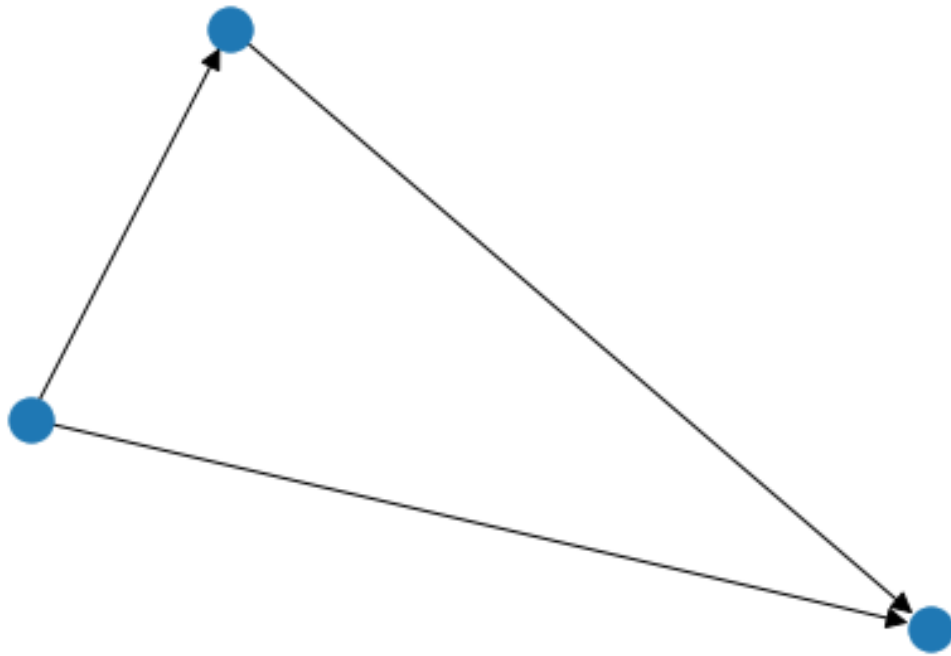
DEFINIÇÃO (digrafo). Um **grafo orientado**, **grafo dirigido** ou **digrafo** é um grafo que possui um conjunto finito de arestas orientadas ou direcionadas.

- Exemplo

```

[ ]: E = [('A', 'B'), ('A', 'C'), ('B', 'C')]
G = nx.DiGraph(E)
nx.draw(G, arrowsize=18)

```



TEOREMA (Euler) 3.1. (Teorema do Aperto de Mão) A soma dos graus dos vértices de um grafo G é igual ao dobro do número de arestas de G , ou seja:

$$\sum_{i=1}^n d(v_i) = 2 \cdot |E| = 2m$$

Demonstração.

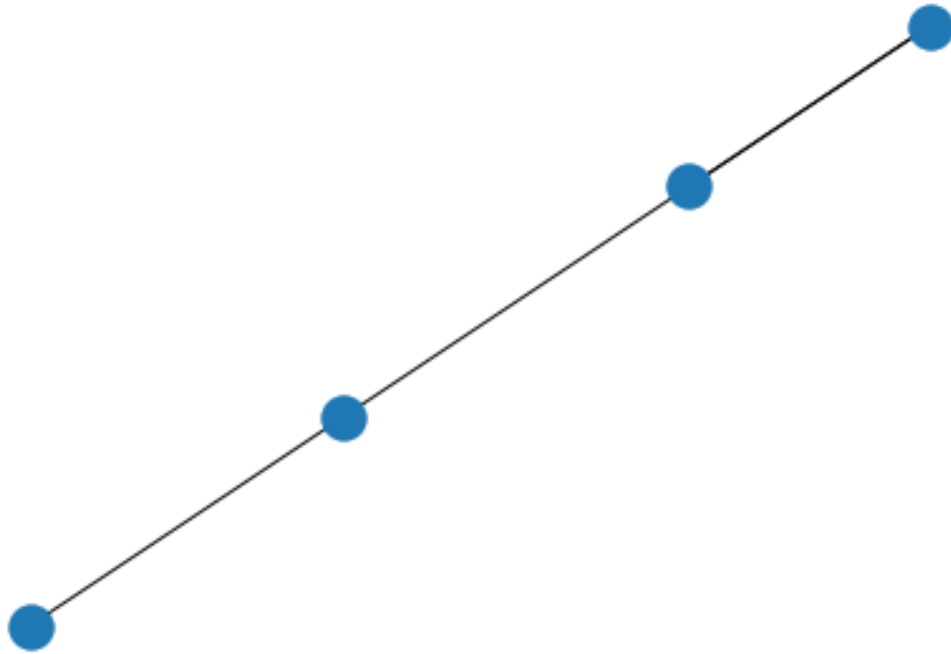
Fonte: https://www.youtube.com/watch?v=BTdB-l_B6bY

DEFINIÇÃO (multigrafo).

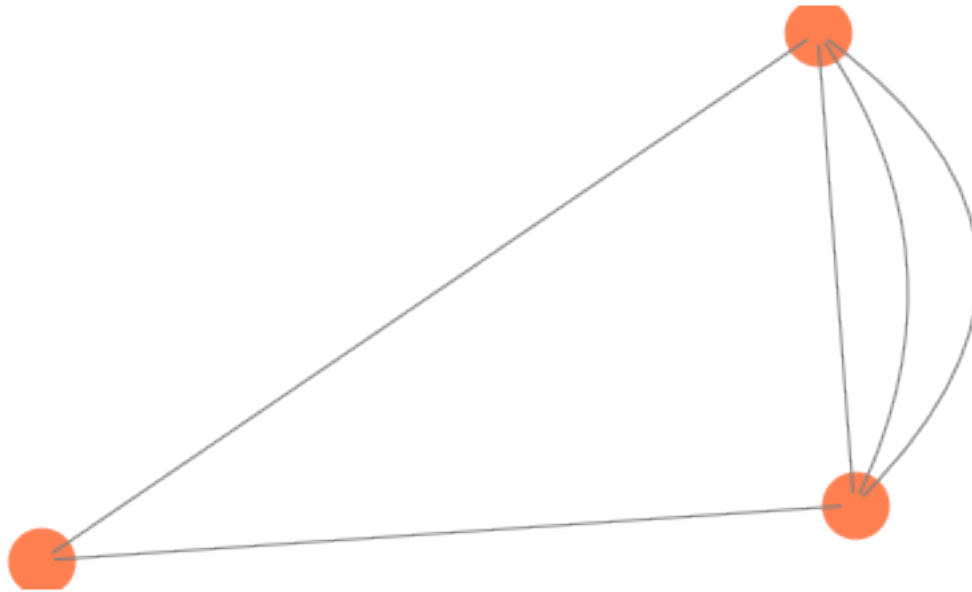
Um **multigrafo** é um grafo não orientado que possui no mínimo duas arestas paralelas.

Exemplo:

```
[ ]: G = nx.MultiGraph()
G.add_edges_from([('C','D'), ('A','B'), ('B','C'), ('C','D')])
nx.draw(G, connectionstyle="arc3, rad=0.1")
```



```
[ ]: G = nx.MultiGraph([(1,2),(1,2),(1,2),(3,1),(3,2)])
pos = nx.random_layout(G)
nx.draw_networkx_nodes(G, pos, node_color = 'coral', node_size = 600, alpha = 1)
ax = plt.gca()
for e in G.edges:
    ax.annotate("",
        xy = pos[e[0]], xycoords='data',
        xytext = pos[e[1]], textcoords='data',
        arrowprops = dict(arrowstyle="-", color="0.5",
            shrinkA=5, shrinkB=5, patchA=None, patchB=None,
            connectionstyle="arc3,rad=rrr".replace('rrr', str(0.3 * e[2])),
        ),
    )
plt.axis('off')
plt.show()
```

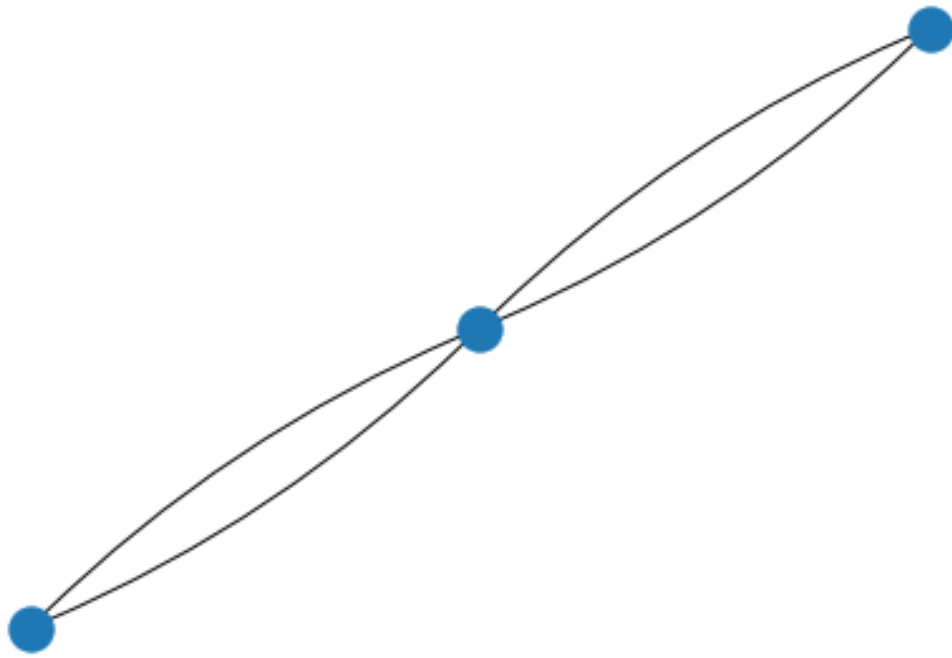


```
[ ]: import networkx as nx
import matplotlib.pyplot as plt

G = nx.MultiDiGraph()

G.add_edges_from([
    (1, 2),
    (2, 3),
    (3, 2),
    (2, 1),
])

# plt.figure(figsize=(8,8))
nx.draw(G, arrowstyle='-', connectionstyle='arc3, rad = 0.1',)
```



```
[ ]: a = [  
    'one',  
    'two',  
    'three',  
    'four',  
    'five'  
]  
  
print(len(a))
```

5