

```
import networkx as nx
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

1) Crie um grafo G considerando os vértices e arestas armazenados no arquivo Grafo3.csv. Verifique se o grafo é direcionado e apresente a resposta na tela (True ou False).

```
Grafo_um = pd.read_csv('drive/MyDrive/Grafo3_semana4 (1).csv', delimiter=',', header=None)
```

```
G1 = nx.from_pandas_adjacency(Grafo_um)
```

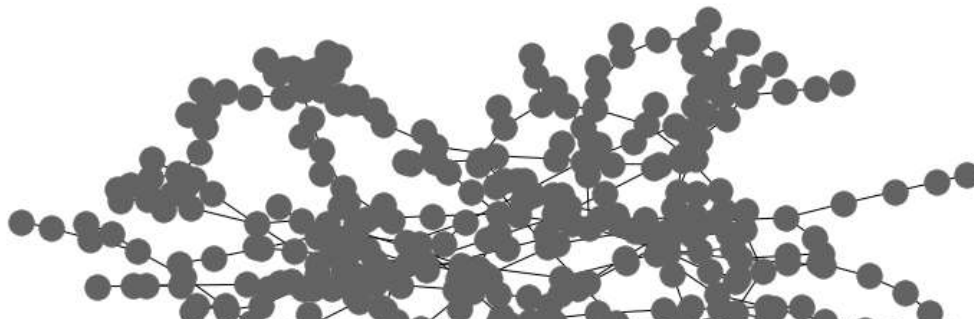
```
G1.is_directed()
```

```
False
```

2) Desenhe o grafo G na tela considerando as dimensões 12 x 7

Para desfazer a exclusão de células, use Ctrl+M Z ou a opção Desfazer no menu Editar ✕

```
plt.figure(figsize=(12, 7))
nx.draw(G1)
```



3) Extraia a quantidade de vértices e de arestas do grafo G.

```
print('Ordem do grafo G1 (qtde. de vértices):', G1.order())
print('Ordem do grafo G1 (qtde. de aresta):', G1.size())
```

```
Ordem do grafo G1 (qtde. de vértices): 512
Ordem do grafo G1 (qtde. de aresta): 512
```

4) Extraia o grau do vértice 33 do grafo G e responda qual a diferença do grau deste vértice em relação ao grau médio da rede.

Para desfazer a exclusão de células, use Ctrl+M Z ou a opção Desfazer no menu Editar ✕

```
Grau do vértice 33 do grafo G1: 3
```

```
graus_G1 = dict(G1.degree())
graus_G1 = list (graus_G1.values())
graus_medio_G1 = np.mean(graus_G1)

print('Grau médio da rede G1:', graus_medio_G1)
```

```
Grau médio da rede G1: 2.0
```

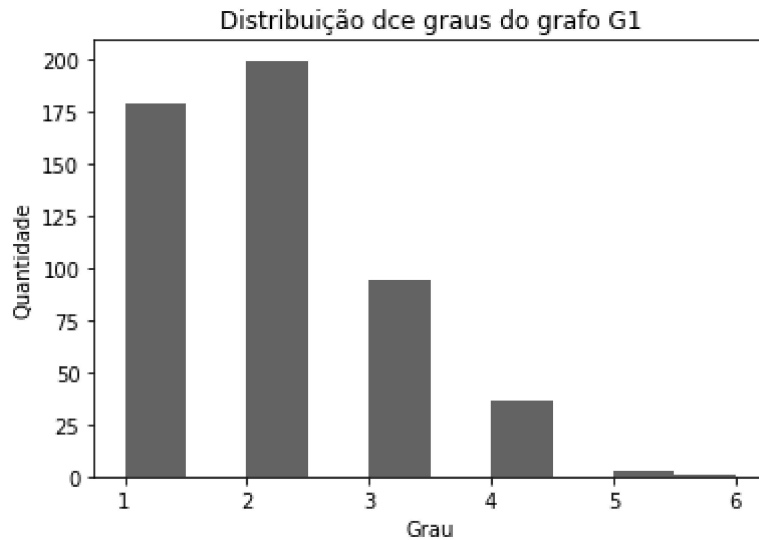
5) Extraia o conjunto de arestas que se conectam ao vértice 33 do grafo G e as apresente utilizando uma lista.

```
vizinhos_33 = list(G1.neighbors(33))
print(vizinhos_33)
```

[32, 34, 335]

```
plt.hist(graus_G1)
plt.title('Distribuição dce graus do grafo G1')
plt.xlabel('Grau')
plt.ylabel('Quantidade')
```

```
Text(0, 0.5, 'Quantidade')
```



```
print('Distância geodésica média de G1:', nx.average_shortest_path_length(G1))
```

Distância geodésica média de G1: 26.305887659001957

```
print('Tamanho do menor caminho entre vértices 5 e 67 do grafo G1:', nx.shortest_path_leng  
(G1, 33))
```

Para desfazer a exclusão de células, use Ctrl+M Z ou a opção Desfazer no menu Editar ✕

Caminho que se conectam ao vértice 33 do grafo G1: {33: [33], 32: [33, 32], 34: [33,

```
caminhos_minimos_G1 = []
for i in range(0, G1.order()):
    for j in range(0, G1.order()):
        if i != j:
            caminhos_minimos_G1.append(nx.shortest_path_length(G1, i, j))
```

```
plt.hist(caminhos_minimos_G1)
plt.title('Distribuição da distância geodésica do grafo G1')
plt.xlabel('Tamanho do caminho mínimo')
plt.ylabel('Quantidade')
```

```
Text(0, 0.5, 'Quantidade')
```



```
print('Excentricidade Do vértice 33 do grafo G1', nx.eccentricity(G1, 33))
```

Excentricidade Do vértice 33 do grafo G1 65

0 10 20 30 40 50 60 70 80

```
print('Raio da rede G1:', nx.radius(G1))
```

Raio da rede G1: 40

```
print('Diâmetro da rede G1', nx.diameter(G1))
```

Diâmetro da rede G1 80

```
print('Densidade da rede G1', nx.density(G1))
```

Densidade da rede G1 0.003913894324853229

Para desfazer a exclusão de células, use Ctrl+M Z ou a opção Desfazer no menu Editar ✕