

PUCPR - Graph Mining - ATP 1:

Aluno: Fabricio Magalhães

Curso: Inteligência Artificial

Com base na matriz de adjacências armazenada no arquivo Grafo3.csv, realize as tarefas descritas a seguir utilizando a linguagem de programação Python e as bibliotecas adequadas. Para cada tarefa, indique o código de programação utilizado e o resultado obtido.

```
In [43]: import networkx as nx
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [44]: file_with_graph_data_as_adj_matrix = './Grafo3.csv'
```

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).

```
In [45]: graph_data = pd.read_csv(file_with_graph_data_as_adj_matrix, header=None)
graph_data = graph_data.values

graph = nx.Graph(graph_data)

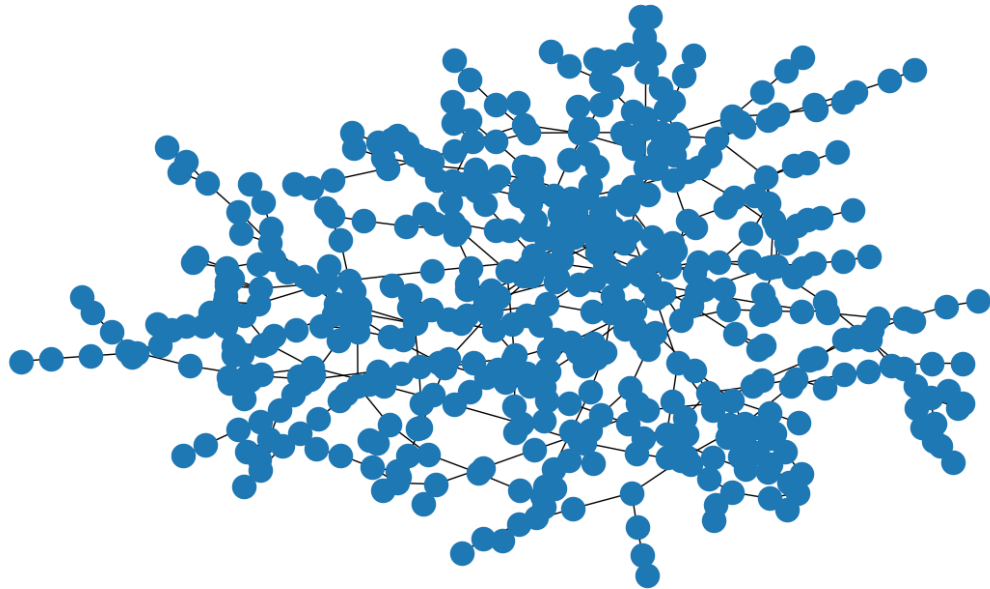
print('É direcionado? ' + str(graph.is_directed()))
```

É direcionado? False

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

```
In [46]: plt.figure(figsize=(12, 7))

nx.draw(graph)
```



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

```
In [47]: nodes_quantity = graph.number_of_nodes()
edges_quantity = graph.number_of_edges()

print('Quantidade de nós: ' + str(nodes_quantity))
print('Quantidade de arestas: ' + str(edges_quantity))
```

Quantidade de nós: 512
Quantidade de arestas: 512

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.

```
In [60]: node_33 = graph[33]
degree_of_node_33 = graph.degree(33)
print('Grau do nó 33: ' + str(degree_of_node_33))

avg_degree = np.mean(list(dict(graph.degree()).values()))
print('Grau médio da rede: ' + str(avg_degree))
```

Grau do nó 33: 3
Grau médio da rede: 2.0

Resposta: O vértice 33 está acima da média da rede.

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

```
In [54]: node_33_edges = graph.edges(33)
print('Arestas do nó 33: ' + str(node_33_edges))
```

Arestas do nó 33: [(33, 32), (33, 34), (33, 335)]

Apresente a distribuição de graus do grafo G em uma figura com dimensões 10 x 5

```
In [57]: plt.figure(figsize=(10, 5))

degree_sequence = sorted([d for n, d in graph.degree()], reverse=True)
degreeCount = np.unique(degree_sequence, return_counts=True)
plt.bar(degreeCount[0], degreeCount[1], color='b')
plt.title('Degree Histogram')
plt.ylabel('Count')
plt.xlabel('Degree')
plt.show()
```

