

```
In [5]: import networkx as nx
import matplotlib.pyplot as plt
from matplotlib.cm import ScalarMappable

# Lê o arquivo GML
G = nx.read_gml("karate.gml", label="id")

# Calcula o grau de cada nó
node_degrees = dict(G.degree())

# Define as cores dos nós com base no grau (por exemplo, nós com alto grau são a
node_colors = list(node_degrees.values())

# Define o tamanho dos nós com base no grau (por exemplo, nós com alto grau são
node_sizes = [v * 100 for v in node_degrees.values()]

# Plota o grafo com nós coloridos e tamanhos personalizados
plt.figure(figsize=(8, 6))
pos = nx.spring_layout(G) # Define o layout do gráfico

# Cria um objeto de mapa de cores para associar valores com cores
mapper = ScalarMappable(cmap=plt.cm.Blues)
mapper.set_array(node_colors)

# Desenha os nós usando as cores e tamanhos personalizados
nodes = nx.draw_networkx_nodes(G, pos, node_color=node_colors, node_size=node_si
edges = nx.draw_networkx_edges(G, pos, width=1.0, alpha=0.5)
nx.draw_networkx_labels(G, pos, font_size=10, font_weight='bold')

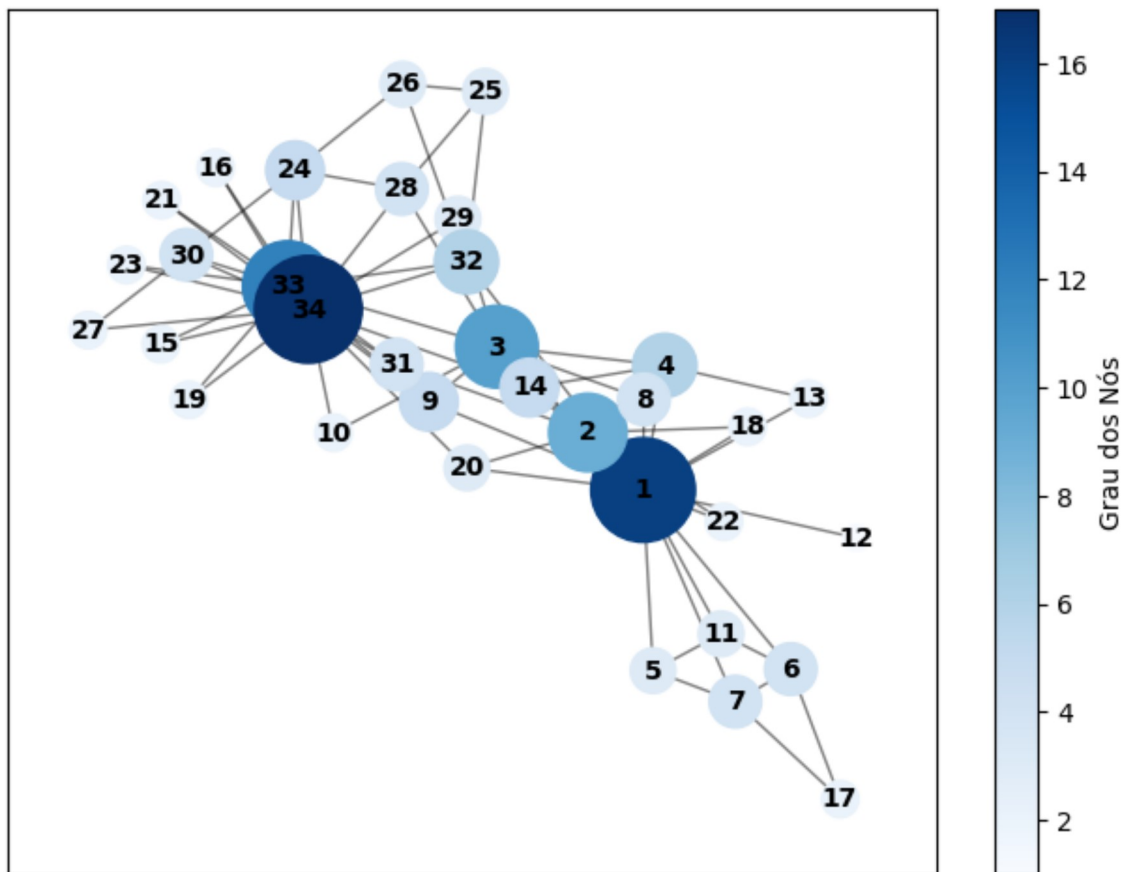
# Adiciona a barra de cores
plt.colorbar(mapper, label='Grau dos Nós')

# Exibe o gráfico
plt.title("Grafo Karate Club com Cores e Tamanhos Personalizados")
plt.show()
```

C:\Users\Nawiwa\AppData\Local\Temp\ipykernel_12892\2589583943.py:31: MatplotlibDeprecationWarning: Unable to determine Axes to steal space for Colorbar. Using.gca(), but will raise in the future. Either provide the *cax* argument to use as the Axes for the Colorbar, provide the *ax* argument to steal space from it, or add *mappable* to an Axes.

```
plt.colorbar(mapper, label='Grau dos Nós')
```

Grafo Karate Club com Cores e Tamanhos Personalizados



```
In [6]: # Calcula N (número de nós)
N = len(G.nodes())

# Calcula L (número de links)
L = len(G.edges())

# Calcular o <k> (Grau médio)
avarage_degree = sum(dict(G.degree()).values()) / N

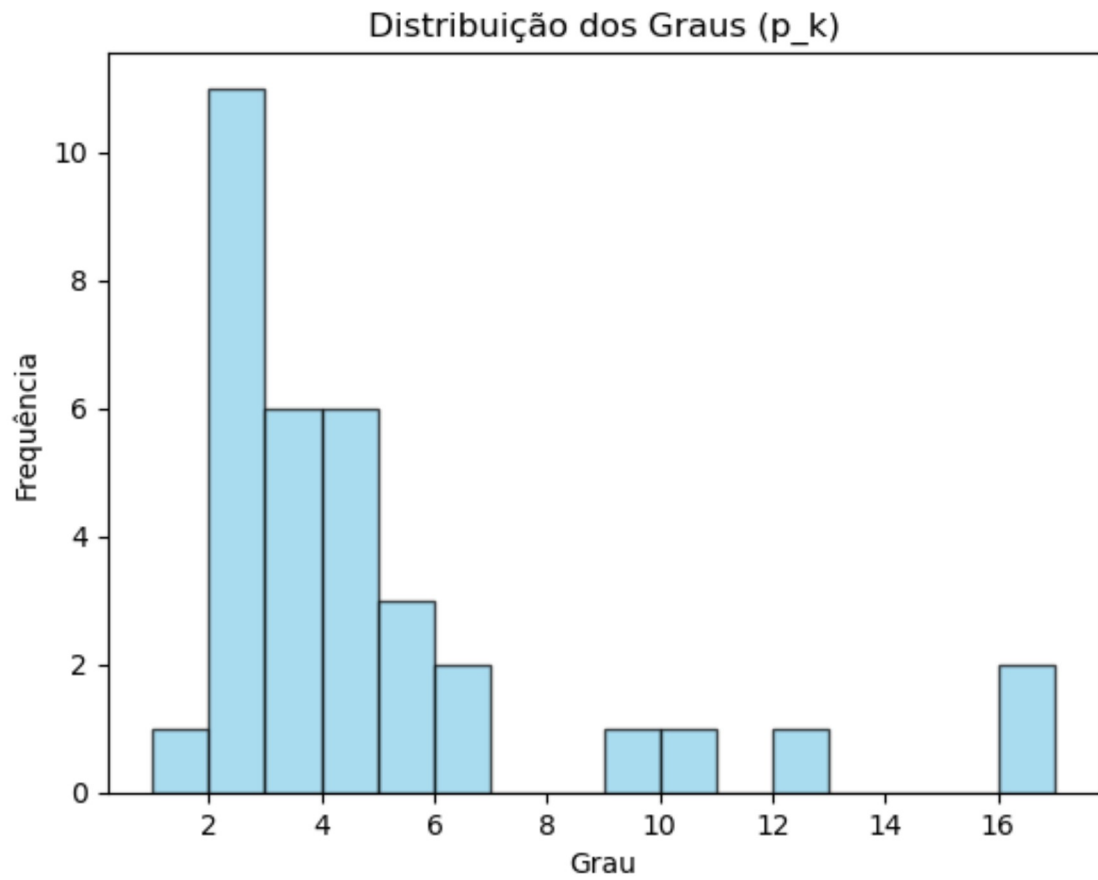
# Calcula L_max (número máximo possível de arestas)
L_max = N * (N - 1) / 2

# Calcula o diâmetro do grafo (D)
diameter = nx.diameter(G)

# Imprime os resultados
print("Numero de Nós (N):", N)
print("Numero de Links (L):", L)
print("Grau Médio <k>:", avarage_degree)
print("Numero maximo possivel de arestas (L_max):", L_max)
print("Diametro do Grafo (D):", diameter)

# Para plotar a distribuição dos graus (p_k)
degrees = dict(G.degree()).values()
plt.hist(degrees, bins=range(min(degrees), max(degrees) + 1), alpha=0.7, color='r')
plt.title("Grafo Karate Club com Cores e Tamanhos Personalizados")
plt.xlabel("Grau")
plt.ylabel("Frequência")
plt.title("Distribuição dos Graus (p_k)")
plt.show()
```

Numero de Nós (N): 34
Numero de Links (L): 78
Grau Médio $\langle k \rangle$: 4.588235294117647
Numero maximo possivel de arestas (L_max): 561.0
Diametro do Grafo (D): 5



In []: