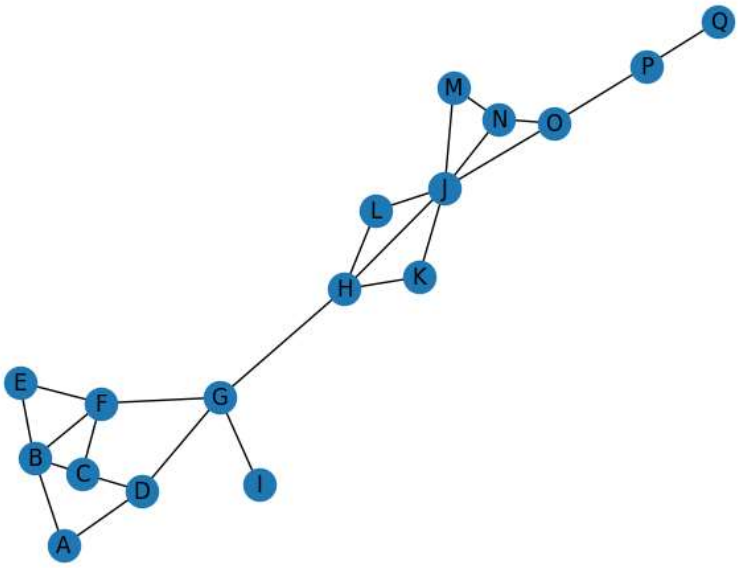
```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
```

Que1 . Generate the graph shown in figure 1 using Networkx and display the following network measures:

```
G1 = nx.Graph()

G1.add_nodes_from(["A", "B", "C", "D","E","F","G","H","I","J","K","L","M","N","O","P","Q"])


G1.add_edges_from([
    ("A","B" ), ("A","D"), ("B","C"),
 ("B","E"), ("B","F"), ("C","D"), ("C","F"), ("E","F"), ("F","G"),
  ("D","G"), ("G","I"),("G","H") ,("H","K") ,("H","L"), ("H", "J"),
   ("K","J"),("J","L"),("J","M"),("M","N"), ("N","J"),("O","J"),("O","N"), ("O","P") ,("P","Q")])


nx.draw(G1,with_labels=True)
```




```
degree_distribution = dict(nx.degree(G1))

print("Degree Distribution:", degree_distribution)
```

```
Degree Distribution: {'A': 2, 'B': 4, 'C': 3, 'D': 3, 'E': 2, 'F': 4, 'G': 4, 'H': 4, 'I': 1, 'J': 6, 'K': 2, 'L': 2, 'M': 2, 'N': 3
```

```
degree_centrality = nx.degree_centrality(G1)

print("Degree Centrality:", degree_centrality)
```

```
Degree Centrality: {'A': 0.125, 'B': 0.25, 'C': 0.1875, 'D': 0.1875, 'E': 0.125, 'F': 0.25, 'G': 0.25, 'H': 0.25, 'I': 0.0625, 'J':
```

```
closeness_centrality = nx.closeness_centrality(G1)
print("Closeness Centrality:", closeness_centrality)
```

```
Closeness Centrality: {'A': 0.2857142857142857, 'B': 0.2962962962962963, 'C': 0.2909090909090909, 'D': 0.34782608695652173, 'E': 0.2
```

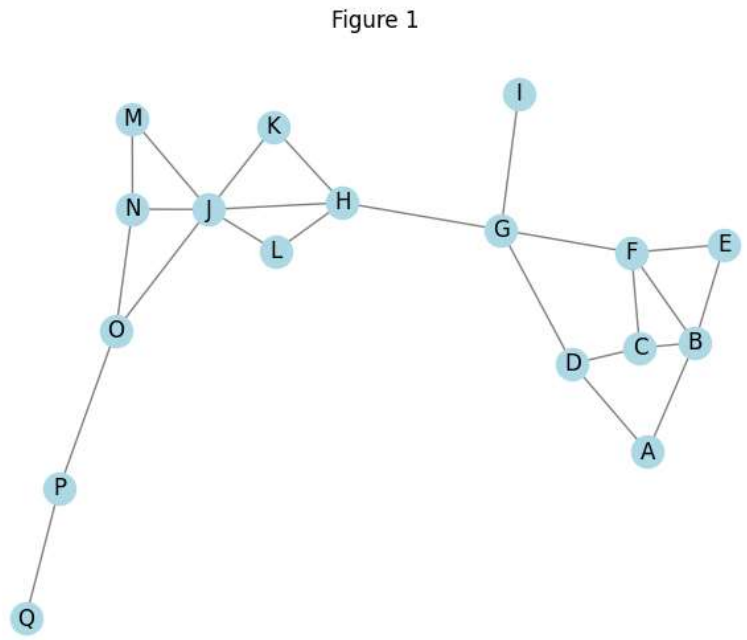
```
betweenness_centrality= nx.betweenness_centrality(G1)

print("Betweenness Centrality:", closeness_centrality)
```

```
Betweenness Centrality: {'A': 0.2857142857142857, 'B': 0.2962962962962963, 'C': 0.2909090909090909, 'D': 0.34782608695652173, 'E': 0
```

```
clustering_coefficient = nx.clustering(G1)

print("Clustering coefficient : ",clustering_coefficient )
```

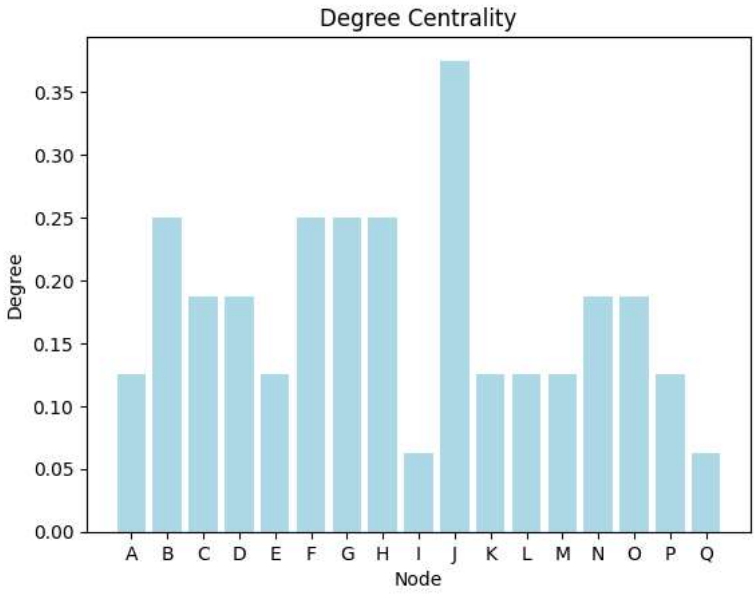
```
Clustering coefficient :  {'A': 0, 'B': 0.3333333333333333, 'C': 0.3333333333333333, 'D': 0, 'E': 1.0, 'F': 0.3333333333333333, 'G'
```

```
pos = nx.spring_layout(G1)
nx.draw(G1, pos, with_labels=True, node_color='lightblue', edge_color='gray')
plt.title("Figure 1")
plt.show()
```

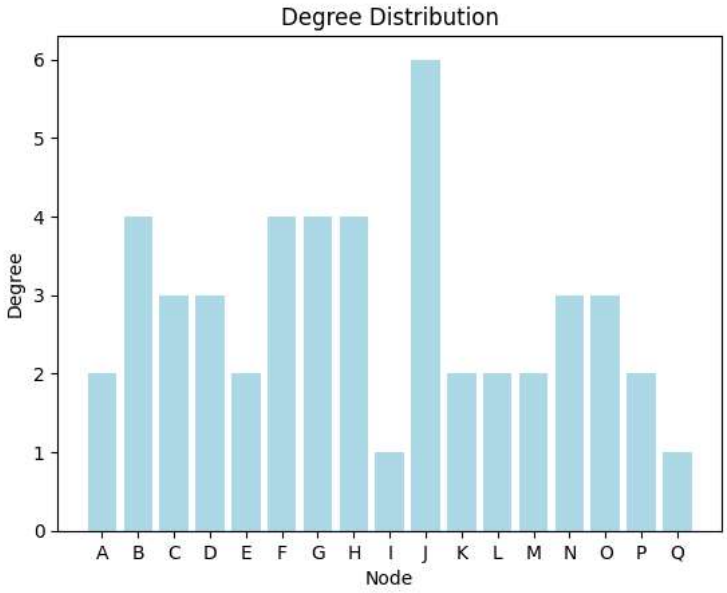



```
def plot_bar(centrality, title, ylabel):
    nodes = G1.nodes()
    values = [centrality[node] for node in nodes]

    plt.bar(nodes, values, color='lightblue')
    plt.title(title)
    plt.xlabel('Node')
    plt.ylabel(ylabel)
    plt.show()
```

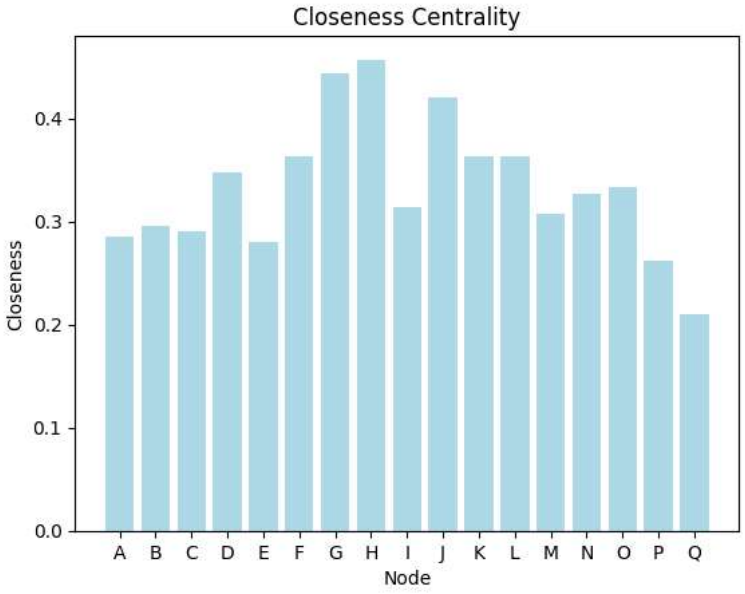
```
plot_bar(degree_centrality, 'Degree Centrality', 'Degree')
```



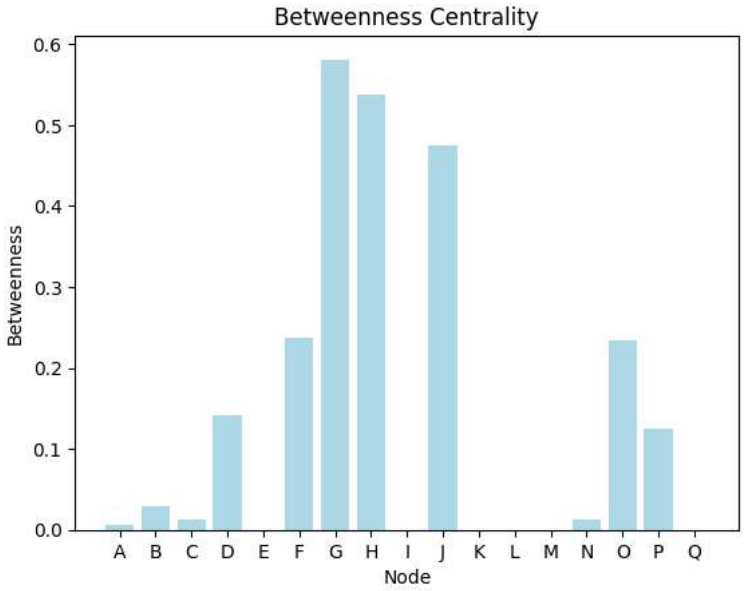

```
plot_bar(degree_distribution, 'Degree Distribution', 'Degree')
```




```
plot_bar(closeness_centrality, 'Closeness Centrality', 'Closeness')
```



```
plot_bar(betweenness_centrality, 'Betweenness Centrality', 'Betweenness')
```




```
plot_bar(clustering_coefficient, 'Clustering Coefficient', 'Clustering Coefficient')
```

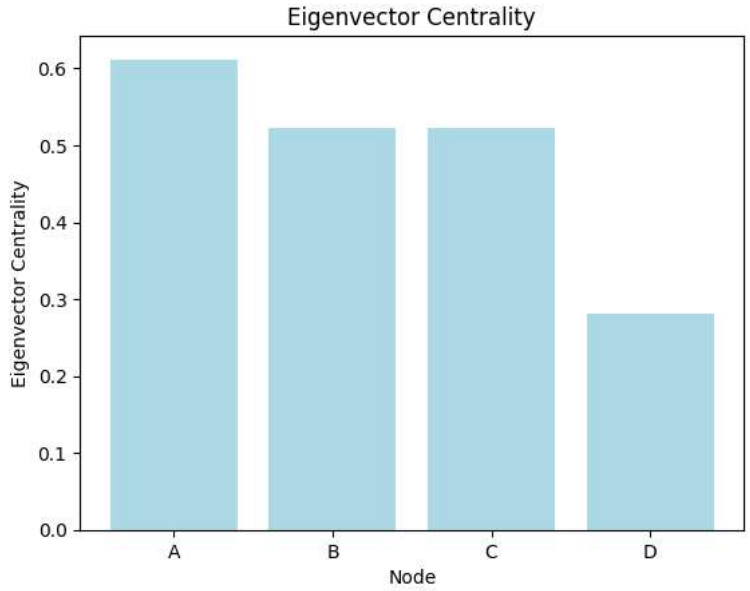
Que2. Generate the graph shown in figure 2 using Networkx and display its Eigenvector centrality

```
G2 = nx.Graph()

G2.add_nodes_from(["A", "B", "C", "D"])


G2.add_edges_from([
    ("A","B" ) , ("B","C"), ("C","A"), ("D","A")])


eigenvector_centrality2 = nx.eigenvector_centrality(G2)

print("Eigenvector centrality : ", eigenvector_centrality2)
```

```
Eigenvector centrality :  {'A': 0.6116286437343044, 'B': 0.5227204550943347, 'C': 0.5227204550943347, 'D': 0.28184579793865727}
```

```
def plot_bar(centrality, title, ylabel):
    nodes = G2.nodes()
    values = [centrality[node] for node in nodes]

    plt.bar(nodes, values, color='lightblue')
    plt.title(title)
    plt.xlabel('Node')
    plt.ylabel(ylabel)
    plt.show()
```

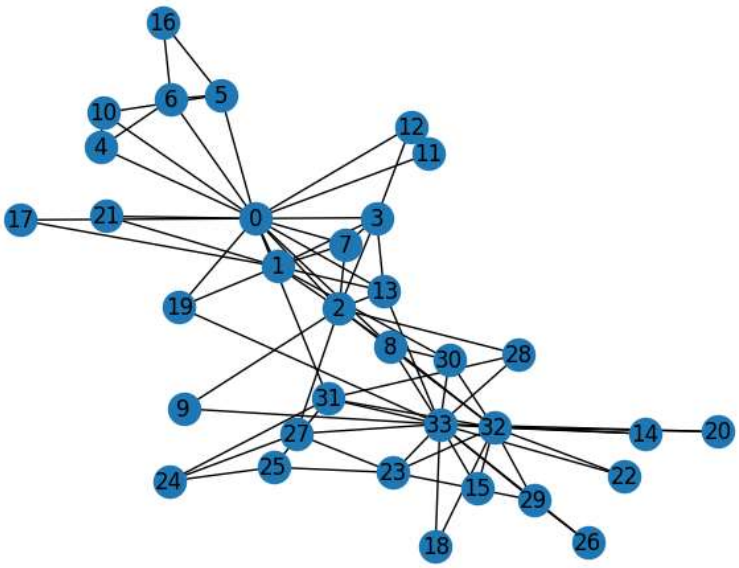
```
plot_bar(eigenvector_centrality2, 'Eigenvector Centrality', 'Eigenvector Centrality')
```



Que3. Consider the famous social graph published in 1977 called Zachary's Karate Club graph. It is an in-built Graph in Networkx. Demonstrate all the centrality measures for this Graph in NetworkX.

```
G3= nx.karate_club_graph()
```

```
nx.draw(G3,with_labels=True)
```



```
degree_centrality = nx.degree_centrality(G3)
closeness_centrality = nx.closeness_centrality(G3)
betweenness_centrality = nx.betweenness_centrality(G3)
eigenvector_centrality = nx.eigenvector_centrality(G3)
katz_centrality = nx.katz_centrality(G3)
pagerank_centrality = nx.pagerank(G3)
clustering_coefficient = nx.clustering(G3)
```

```
print("Degree Centrality:", degree_centrality)
```

```
Degree Centrality: {0: 0.48484848484848486, 1: 0.2727272727272727, 2: 0.30303030303030304, 3: 0.18181818181818182, 4: 0.090909090909
```

```
print("Closeness Centrality:", closeness_centrality)
```

```
Closeness Centrality: {0: 0.5689655172413793, 1: 0.4852941176470588, 2: 0.559322033898305, 3: 0.4647887323943662, 4: 0.3793103448275
```

```
print("Betweenness Centrality:", betweenness_centrality)
```

```
Betweenness Centrality: {0: 0.43763528138528146, 1: 0.053936688311688304, 2: 0.14365680615680618, 3: 0.011909271284271283, 4: 0.0006
```

```
print("Eigenvector Centrality:", eigenvector_centrality)
```

```
Eigenvector Centrality: {0: 0.35548349418519426, 1: 0.2659538704545024, 2: 0.3171893899684447, 3: 0.21117407832057056, 4: 0.0759664
```

```
print("Katz Centrality:", katz_centrality)
```

```
Katz Centrality: {0: 0.3213245969592325, 1: 0.2354842531944946, 2: 0.2657658848154288, 3: 0.1949132024917254, 4: 0.1219044056494841
```

```
print("PageRank Centrality:", pagerank_centrality)
```

```
PageRank Centrality: {0: 0.08850807396280012, 1: 0.057414840497110056, 2: 0.06276686454603017, 3: 0.037212081536313777, 4: 0.0205039
```

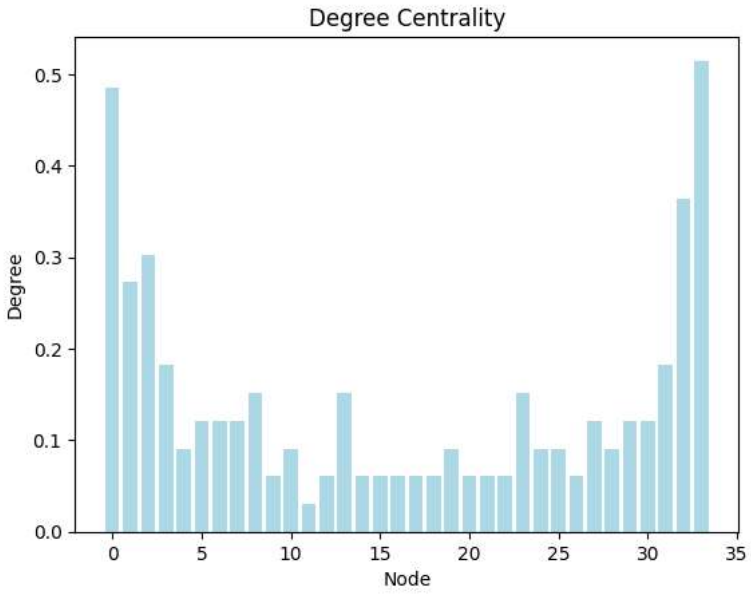
```
print("Clustering Coefficient:", clustering_coefficient)
```

```
Clustering Coefficient: {0: 0.15, 1: 0.3333333333333333, 2: 0.24444444444444444, 3: 0.6666666666666666, 4: 0.6666666666666666, 5: 0
```

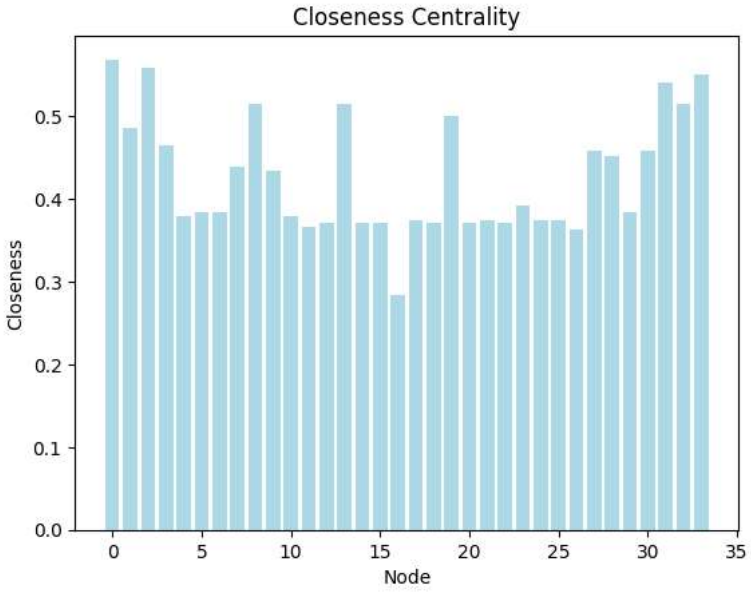
```
pos = nx.spring_layout(G3)
```

```
def plot_bar(centrality, title, ylabel):
    nodes = G3.nodes()
    values = [centrality[node] for node in nodes]

    plt.bar(nodes, values, color='lightblue')
    plt.title(title)
    plt.xlabel('Node')
    plt.ylabel(ylabel)
    plt.show()
```

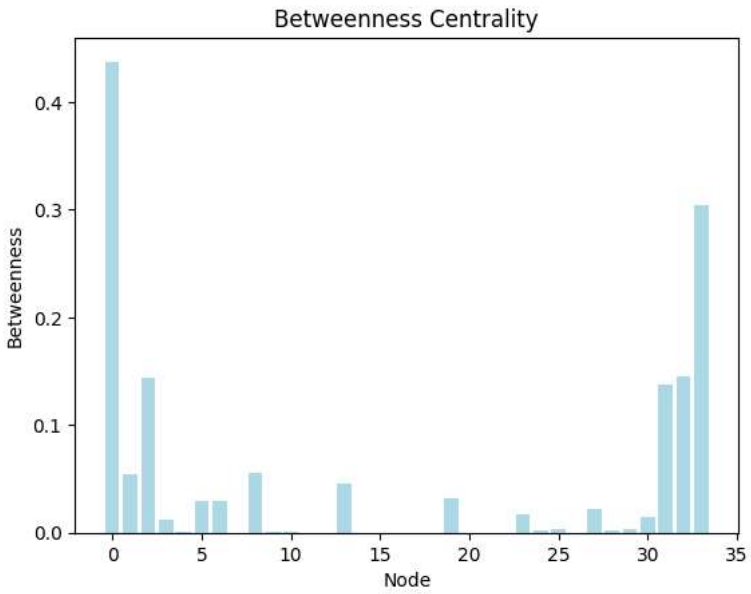
```
plot_bar(degree_centrality, 'Degree Centrality', 'Degree')
```



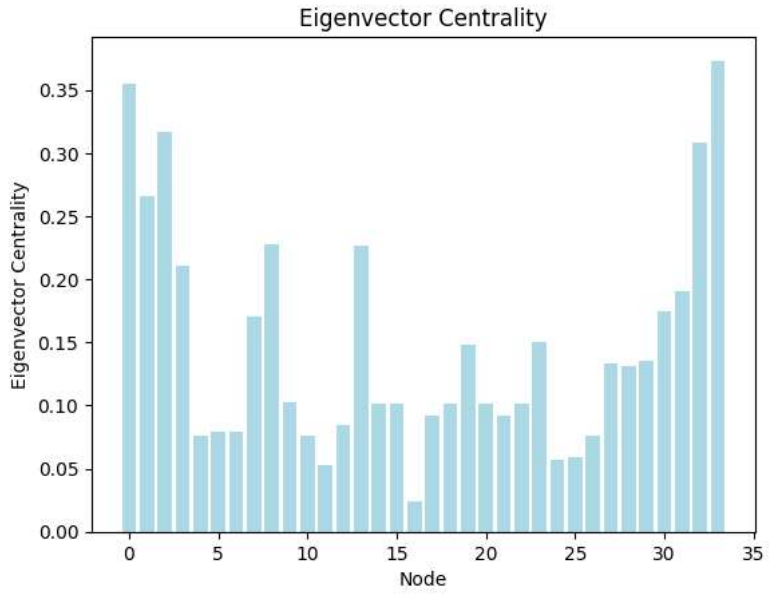

```
plot_bar(closeness_centrality, 'Closeness Centrality', 'Closeness')
```



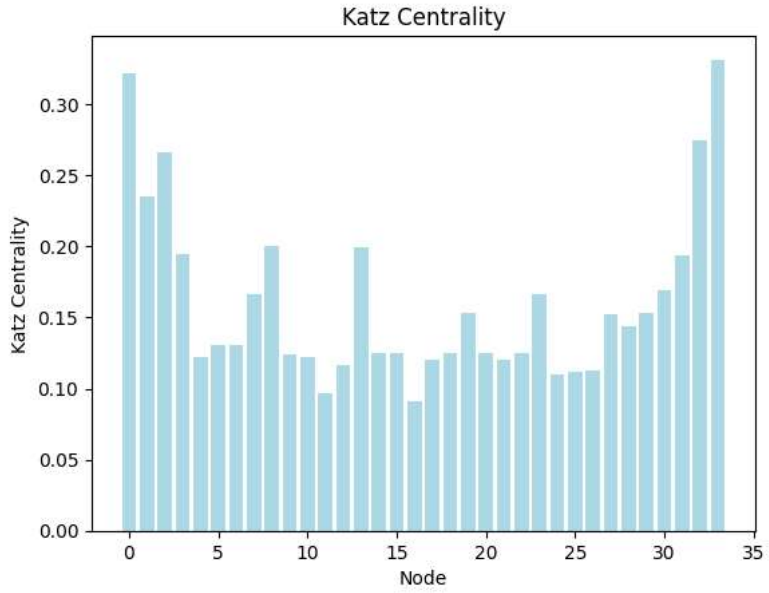

```
plot_bar(betweenness_centrality, 'Betweenness Centrality', 'Betweenness')
```



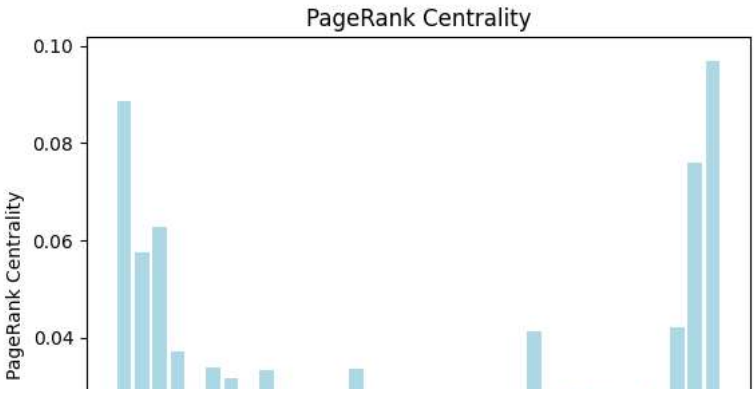
```
plot_bar(eigenvector_centrality, 'Eigenvector Centrality', 'Eigenvector Centrality')
```




```
plot_bar(katz_centrality, 'Katz Centrality', 'Katz Centrality')
```




```
plot_bar(pagerank_centrality, 'PageRank Centrality', 'PageRank Centrality')
```


PageRank Centrality
PageRank Centrality
0.10
0.08
0.06
0.04