LAB4: Algorytmy grafowe – najkrótsza ścieżka w grafie

Zadanie 1

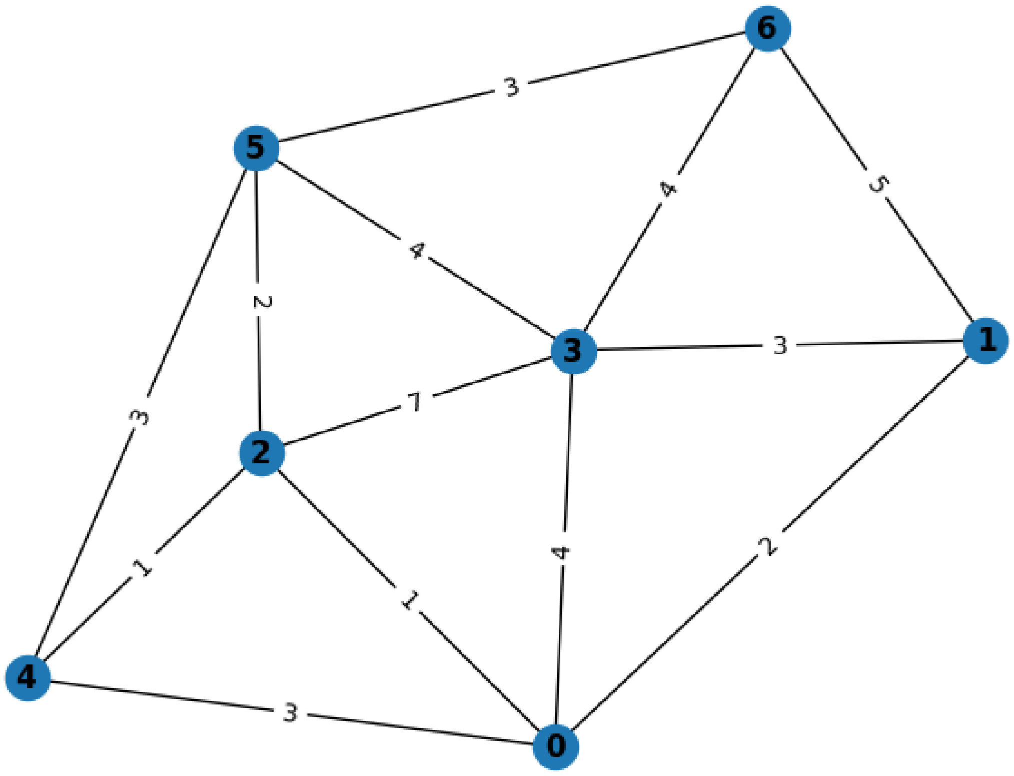
In [19]:

```
from typing import List, Dict, Tuple, Set
import matplotlib.pyplot as plt
import networkx as nx
edge = Tuple[int, int]
graph = Dict[int, List]
inf = float('inf')

def Bellman_Ford(G: graph, a: List[List[int]], s: int) -> Tuple[int, List[int]]:
    d = {v: inf for v in G}
    p = {v: -1 for v in G}

    d[s] = 0

    for i in range(len(G)-1):
        for u in G:
            for v in G[u]:
                if d[v] > d[u] + a[u][v]:
                    d[v] = d[u] + a[u][v]
                    p[v] = u
    return sum(d.values()), p
```

In [22]:

```
graph = {
    0: [1, 2, 3, 4],
    1: [0, 3, 6],
    2: [0, 3, 4, 5],
    3: [0, 1, 2, 5, 6],
    4: [0, 2, 5],
    5: [2, 3, 4, 6],
    6: [1, 3, 5]
}

a = [[inf, 2, 1, 4, 3, inf, inf],
    [2, inf, inf, 3, inf, inf, 5],
    [1, inf, inf, 7, 1, 2, inf],
    [4, 3, 7, inf, inf, 4, 4],
    [3, inf, 1, inf, inf, 3, inf],
    [inf, inf, 2, 4, 3, inf, 3],
    [inf, 5, inf, 4, inf, 3, inf]]

def info_graph(graph, weights, title=''):
    G = nx.Graph(graph)
    for u, v in G.edges:
        G.edges[u, v]['weight'] = weights[u][v]
    print(nx.single_source_bellman_ford_path(G, source=1))
    pos = nx.spring_layout(G)
    nx.draw(G, pos, with_labels=True, font_weight='bold')
    labels = nx.get_edge_attributes(G, 'weight')
    nx.draw_networkx_edge_labels(G, pos, edge_labels=labels)
    plt.show()
info_graph(graph, a)


d, path = Bellman_Ford(graph, a, 0)

print(path)
```

```
{1: [1], 0: [1, 0], 3: [1, 3], 6: [1, 6], 2: [1, 0, 2], 4: [1, 0, 2, 4], 5: [1, 0,
2, 5]}
```



```
{0: -1, 1: 0, 2: 0, 3: 0, 4: 2, 5: 2, 6: 5}
```

In [ ]:

```
G = nx.DiGraph(graph2)
pos = nx.spring_layout(G)
nx.draw(G, pos, with_labels=True)
nx.draw_networkx_edge_labels(G, pos, edge_labels={(u, v): str(a2[u][v]) for u, nbrs
plt.show()
```