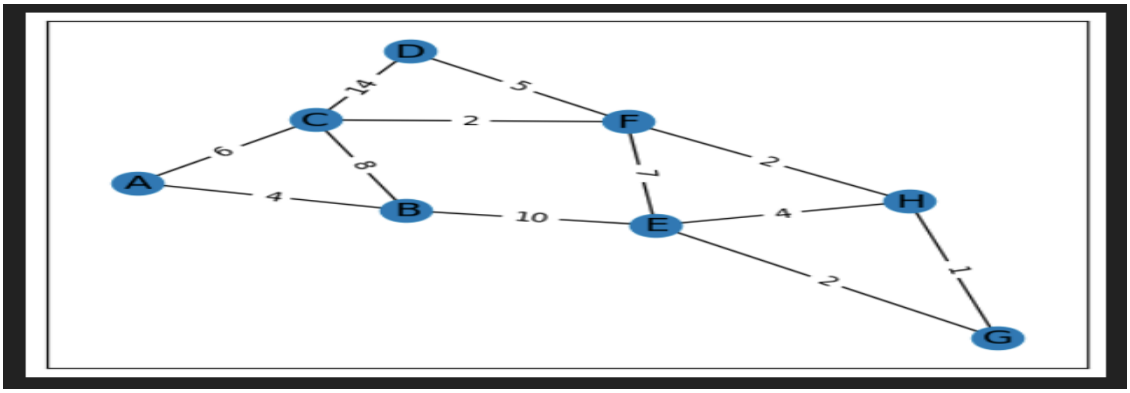
CSCI3070U: Analysis & Design of Algorithms
MST Activity

Syed Naqvi
100590852

November 26, 2023

Original Graph:



1)

Code:

```
def find_set(nodeSets, element):
    for s in nodeSets:
        if element in s:
            return s
    return None

def mst_Kruskal(G):

    # empty graph
    A = nx.Graph()

    # list of nodes sets
    nodeSets=[]
    for v in G.nodes():
        nodeSets.append(set(v))

    # edges sorted in ascending order by weight
    edges = []
    for v1, v2, w in G.edges(data=True):
        edges.append((v1, v2, w[':weight']))
    edges.sort(key=(lambda x: x[2]))

    print(nodeSets)
    for e in edges:
        set1 = find_set(nodeSets, e[0])
        set2 = find_set(nodeSets, e[1])
        if set1 != set2:
            A.add_weighted_edges_from([e])
            # Merge the two sets and update the nodeSets list
            merged_set = set1.union(set2)
            nodeSets.remove(set1)
            nodeSets.remove(set2)
            nodeSets.append(merged_set)
            print(nodeSets)
    return A
```

Output:

```
[{'A'}, {'B'}, {'C'}, {'D'}, {'E'}, {'F'}, {'H'}, {'G'}]
[{'A'}, {'B'}, {'C'}, {'D'}, {'E'}, {'F'}, {'H', 'G'}]
[{'A'}, {'B'}, {'D'}, {'E'}, {'H', 'G'}, {'F', 'C'}]
[{'A'}, {'B'}, {'D'}, {'F', 'C'}, {'E', 'G', 'H'}]
[{'A'}, {'B'}, {'D'}, {'E', 'F', 'G', 'H', 'C'}]
[{'D'}, {'E', 'F', 'G', 'H', 'C'}, {'B', 'A'}]
[{'B', 'A'}, {'E', 'D', 'F', 'G', 'H', 'C'}]
[{'E', 'G', 'H', 'D', 'B', 'F', 'C', 'A'}]
```