Data Analysis and Algorithm

Practical 7

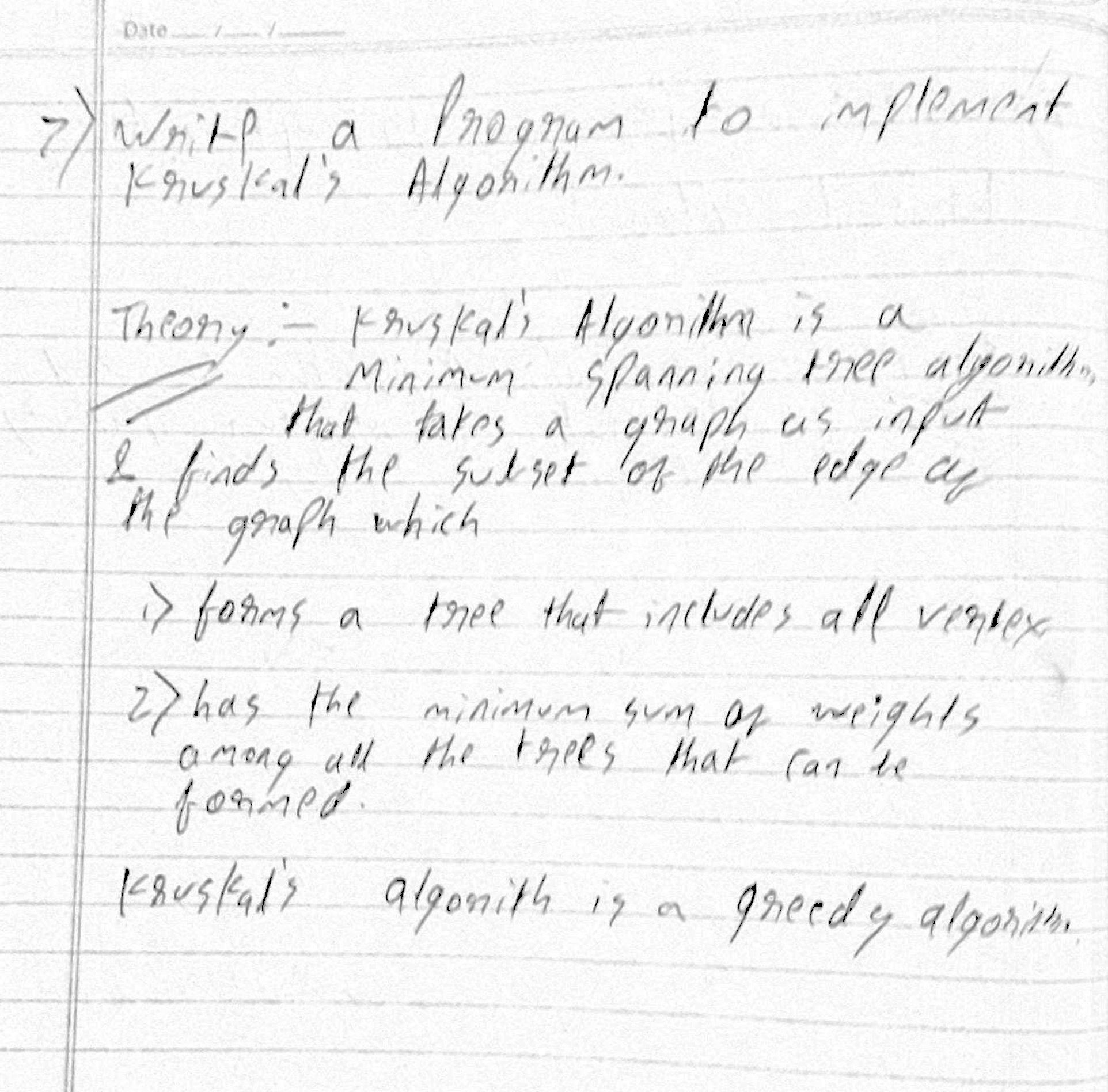
Write a program to implement Krushkals algorithm.

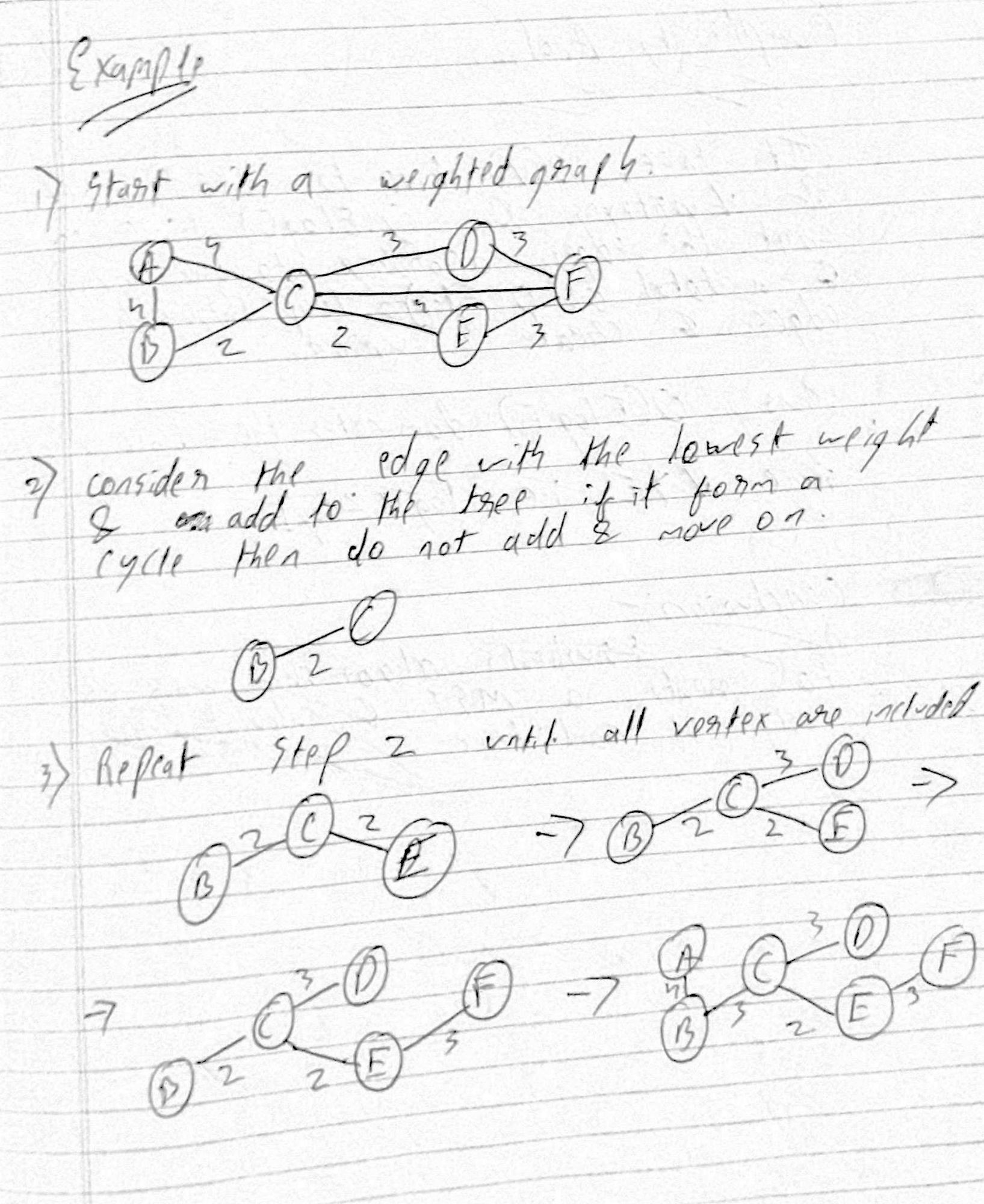
Date.: 19-10-21

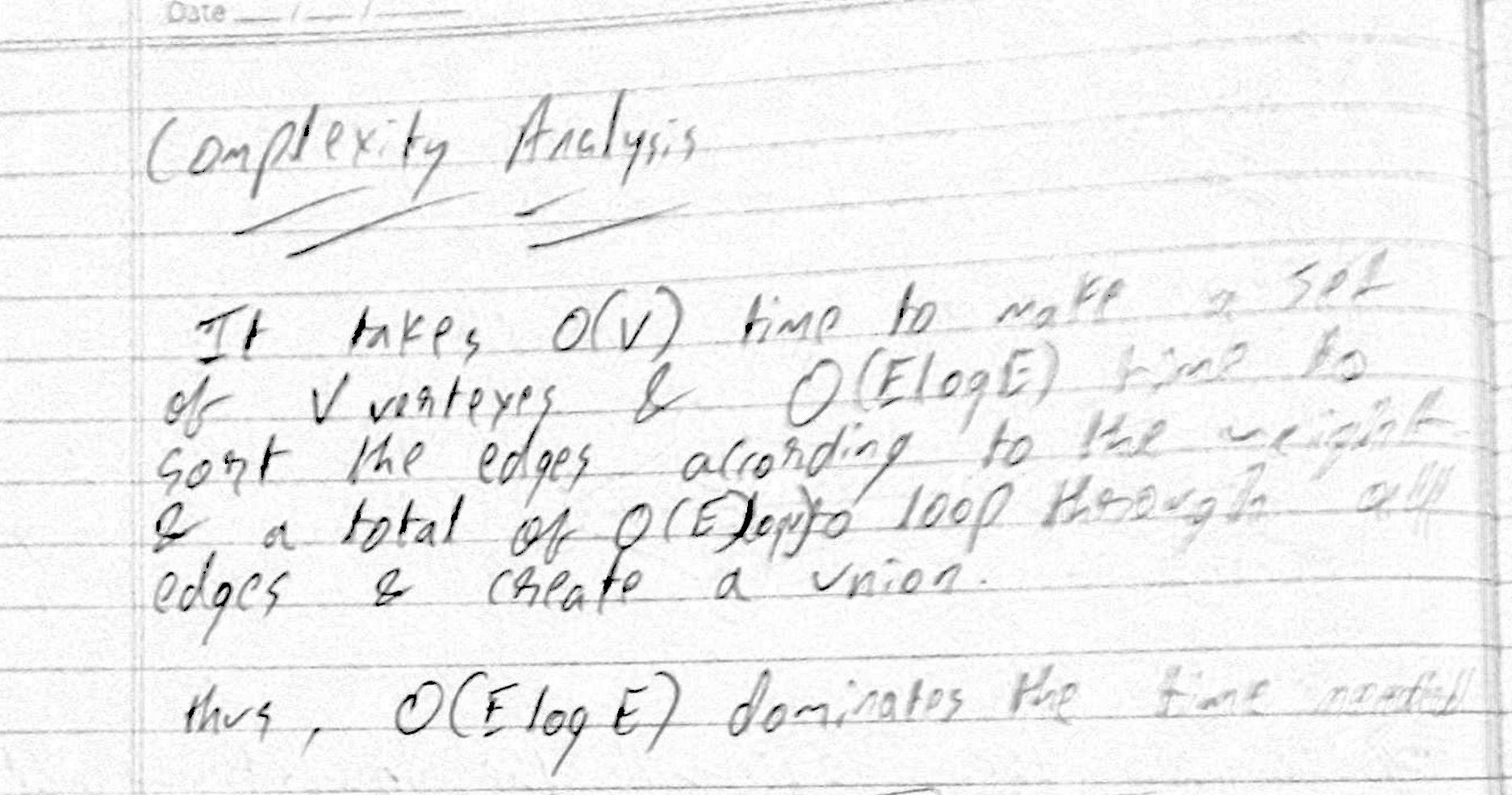
Name – Yash Vasudeo Prajapati

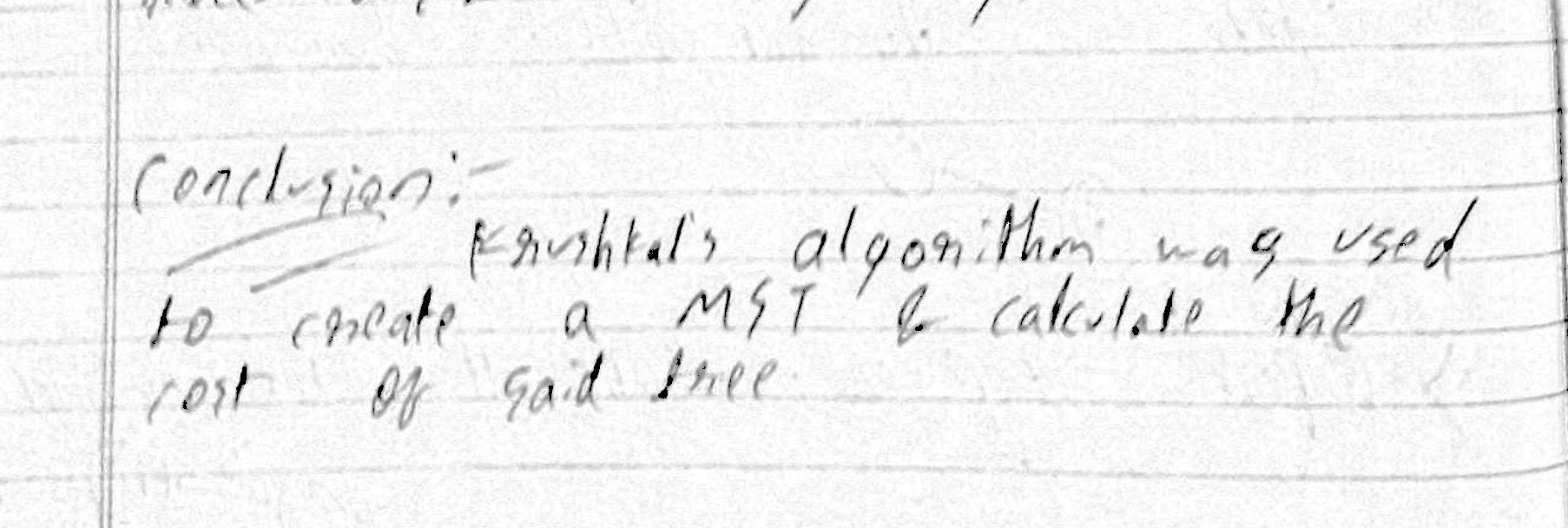
Rollno - 022

MSc. Computer Science









Program

import matplotlib.pyplot as plt

import networkx as nx

import random

class Graph:

def \_\_init\_\_(self,vertex):

self.v = vertex

self.graph = []

def add\_edge(self, source, destination, weight):

self.graph.append([source, destination, weight])

def algo(self):

parent, rank, result = [], [], []

i,e = 0,0

self.graph = sorted(self.graph, key=lambda item:item[2])

for node in range(self.v):

parent.append(node)

rank.append(0)

while e < self.v -1:

s, d, w = self.graph[i]

i += 1

x = self.find(parent, s)

y = self.find(parent, d)

if x != y:

e += 1

result.append([s, d, w])

self.union(parent, rank, x, y)

return result

def find(self, p, i):

if p[i] ==i:

return i

return self.find(p, p[i])

def union(self, p, r, x, y):

s = self.find(p, x)

d = self.find(p, y)

if r[s] < r[d]:

p[s] = d

elif r[s] > r[d]:

p[d] = s

else:

p[d] = s

r[s] += 1

def plot(G):

pos = nx.spring\_layout(G, seed=7)

nx.draw\_networkx\_nodes(G, pos, node\_size=700)

# edges

nx.draw\_networkx\_edges(G, pos, width=6)

nx.draw\_networkx\_edges(

G, pos, width=6, alpha=0.5, edge\_color="b", style="dashed"

)

# labels

nx.draw\_networkx\_labels(G, pos, font\_size=20, font\_family="sans-serif")

ax = plt.gca()

ax.margins(0.08)

plt.axis("off")

plt.tight\_layout()

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

graph = Graph(10)

for i in range(0 ,random.randint(19,50)):

graph.add\_edge(random.randint(0,9),random.randint(0,9), random.randint(0,9))

result = graph.algo();

G = nx.Graph()

for u, v, w in graph.graph:

G.add\_edge(u, v, weight=w)

plot(G)

G = nx.Graph()

for u, v, w in result:

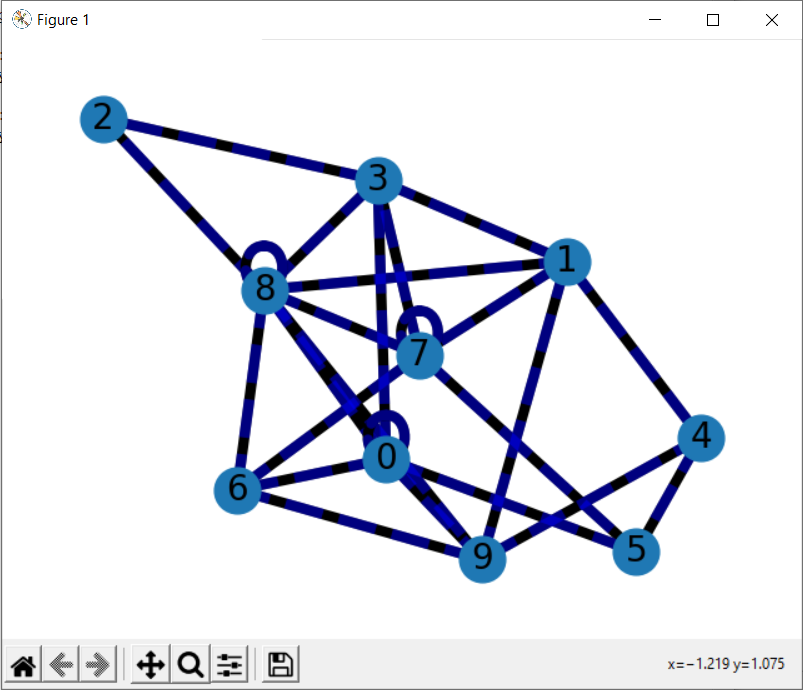
G.add\_edge(u, v, weight=w)

print("%d - %d: %d" % (u, v, w))

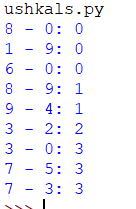
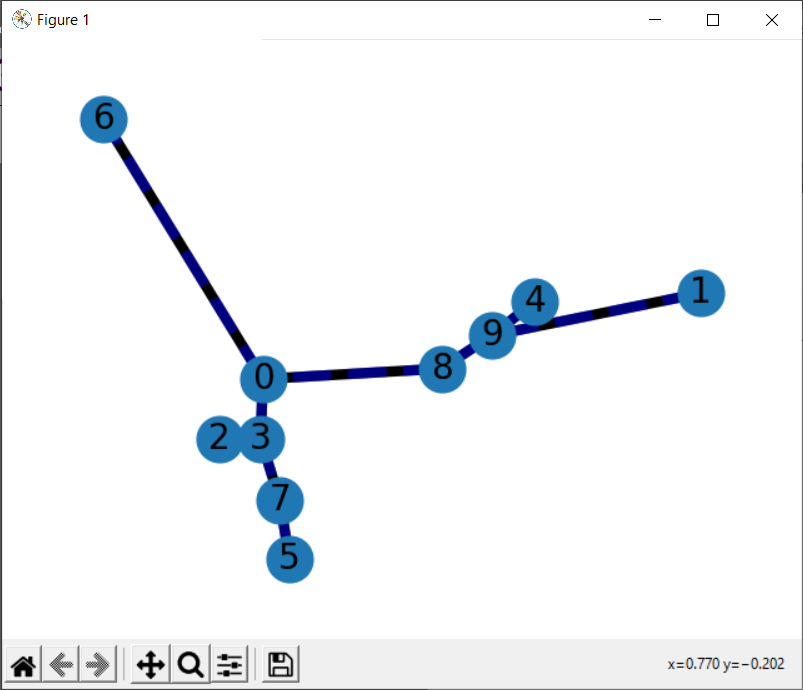
plot(G)

Output:-

Original



**MST:-**

****