# Assignment 3

Implement uniform cost search and find the path from S-G2 for a given graph.

Code:

# keywords

# S->0 , A->1 , B->2 , C->3 , D->4 ,

# E->5 , F->6 , G1->7 , G2->8 , G3->9

g = {

    0:[(1,5),(2,9),(4,6)],

    1:[(7,9),(2,3)],

    2:[(1,2),(3,1)],

    3:[(0,6),(6,7),(8,5)],

    4:[(3,2),(5,2)],

    5:[(9,7)],

    6:[(4,2),(9,8)],

    7:[],

    8:[],

    9:[]

}

g1 = {

    1:[(2,1),(4,2),(3,3)],

    2:[(1,1),(4,4),(5,1)],

    3:[(1,3),(4,2),(7,3)],

    4:[(1,2),(2,4),(3,2),(7,5),(5,1),(6,1)],

    5:[(2,1),(4,1),(6,3),(8,2)],

    6:[(4,1),(5,3),(7,2),(8,2),(9,2)],

    7:[(3,3),(4,5),(6,2)],

    8:[(5,2),(6,2)],

    9:[(6,2)]

}

class pqueue:

    def \_\_init\_\_(self):

        self.q = {}

    def insert(self,cost,dest):

        if(dest in self.q):

            self.q[dest]=min(self.q[dest],cost)

            if(self.q[dest]>cost):

                return True

            else:

                return False

        else:

            self.q[dest]=cost

            return True

    def get(self):

        self.q = {k:v for k,v in sorted(self.q.items(),key=lambda x:x[1])}

        x = list(self.q.items())[0]

        del self.q[x[0]]

        return x

    def empty(self):

        if(len(self.q)==0):

            return True

        return False

def ucs(g,s,t):

    pq = pqueue()

    visited = []

    parent = {}

    pq.insert(0,s)

    while(not pq.empty()):

        x = pq.get()

        if(x[0]==t):

            visited.append(x[0])

            pth = []

            cur = x[0]

            while(cur!=s):

                pth = [cur]+pth

                cur = parent[cur]

            pth = [s]+pth

            return x[1],pth

        elif(x[0] in visited):

            continue

        else:

            for nd in g[x[0]]:

                res = pq.insert(x[1]+nd[1],nd[0])

                if(res):

                    parent[nd[0]]=x[0]

            visited.append(x[0])

cost,path = ucs(g,0,8)

print("Sarthak Kapaliya")

print("20BCP072")

print("Source: 0", "Target: 8")

print("Min Cost: ",cost)

print("Path: "," -> ".join(list(map(str,path))))

Output:

