Assignment 2 - code

2. Write a program that, given a directed graph and two vertices, finds a lowest length path between them, by using a backward breadth-first search from the ending vertex.

def lowest\_length\_path(self, start, end):

visited = []

queue = [end]

dist = {end: 0}

predecessors = {}

while queue:

x = queue.pop(0)

visited.append(x)

if x == start:

path = [x]

while path[-1] != end:

path.append(predecessors[path[-1]])

return list(path)

for i in self.\_\_list\_of\_predecessors[x]:

if i not in visited:

queue.append(i)

visited.append(i)

dist[i] = dist[x] + 1

predecessors[i] = x

return None

The function finds the lowest-length path from a given starting vertex to a given ending vertex in the graph, using breadth-first search. It takes two arguments which are the starting and ending vertices. It returns a list representing the path from start to end, or None if no such path exists.

The method initializes an empty visited list, a queue containing the end vertex, a dist dictionary mapping vertices to their distances from the end vertex (initialized to 0 for end), and a predecessors dictionary mapping vertices to their immediate predecessors in the path. It then enters a loop that continues as long as the queue is not empty.

In each iteration of the loop, the method dequeues the first vertex x from the queue, appends x to the visited list, and checks if x is equal to the start vertex. If it is, the method reconstructs and returns the path from start to end using the predecessors dictionary and the path list. Otherwise, the method iterates over the predecessors of x and, for each predecessor i that has not been visited yet, adds i to the queue, appends i to the visited list, sets its distance to dist[x] + 1, and sets its predecessor to x.

If the loop completes without finding a path from start to end, the method returns None.

• the minimum length path and the length from 1 to 100 in graph1k:

The lowest lenght path is: [1, 5, 487, 175, 699, 624, 100], lenght 6

• the minimum length path and the length from 1 to 100 in graph10k:

The lowest lenght path is: [1, 7317, 4118, 2404, 690, 1494, 739, 4722, 100], lenght 8

• the minimum length path and the length from 1 to 100 in graph100k:

The lowest lenght path is: [1, 17024, 27471, 14969, 3075, 4156, 32753, 14973, 100], lenght 8

• the minimum length path and the length from 100 to 1 in graph1k:

The lowest lenght path is: [100, 416, 354, 865, 109, 1], lenght 5

• the minimum length path and the length from 100 to 1 in graph10k:

The lowest lenght path is: [100, 5568, 2781, 1451, 4997, 528, 4260, 1], lenght 7

• the minimum length path and the length from 100 to 1 in graph100k:

The lowest lenght path is: [100, 44340, 54527, 6606, 53263, 95930, 98655, 58288, 1], lenght 8