Task #3

1.graph traversal algorithms

Graph traversal algorithms : they are used for navigation and exploration of graphs .

They aim to visit all nodes in graph in an organized and efficient manner .

The traversal method depends on the type of graph and the desired objective .

There are two types of graph traversal algorithms :

1.Breadth-first search (BFS)

USING QUEUE

In this method traversal is done layer by layer .

Starting from the first point and then moving to the adjacent point before going to the next layer .

A queue is used to keep the track of the visited nodes and the waiting nodes .

2.Depth-first search (DFS)

USING STACK

This method works by descending deeply into the graph before going to the next node .

A stack is used to keep the track of the visited nodes and nodes waiting to be visited .

2.Hash map usage in python

Hash map is used in python typically as a dictionary

Which is a data structure that allows you to map values with keys so you can access these values by these keys .

For example :

# creating a hash map (dictionary)

my\_dict={}

# add values

my\_dict[“apple”]=5

my\_dict[“banana”]=10

my\_dict[“orange”]=15

# access values by keys

Print(my-dict[“apple”]) # output: 5

Print(my-dict[“banan”]) # output: 10

Print(my-dict["orange”]) # output: 15

3.Graph implementation in python

Graph is one of data structures that we have to create a method to represent it .

There are two methods to implement graph in python .

1.adjacency list

In this method we use dictionary so the keys represent the nodes and the values represent the adjacent nodes . this method is efficient for graphs with fewer nodes .

# creating empty graph

graph={}

# add nodes

graph[‘A’]=[]

graph[‘B’]=[]

graph['C’]=[]

# adding edges by adding adjacent nodes

graph[‘A’].append(‘B’)

graph[‘B’].append(‘C’)

graph[‘C’].append(‘B’)

graph[‘B’].append(‘A’)

# get adjacent nodes

Print(graph[‘A’]) # output: [‘c’]

2.adjacency matrix

In this method we represent the graph by a 2-d matrix where rows and columns are the nodes and the values is the values of edges between them or the absence of it .