**Unit 3 Algorithmics**

**Submit Task – Week 5**

Debugging

Find the errors in the pseudocode below.

Depth first search:

DFS(node):

mark node as visited

print node

for each neighbour in neighbours of node:

if neighbour is not visited:

mark neighbour as visited

DFS(neighbour)

Breadth first search:

BFS(graph, start):

create an empty queue Q

enqueue start into Q

mark start as visited

while Q !empty:

node = dequeue Q

print node

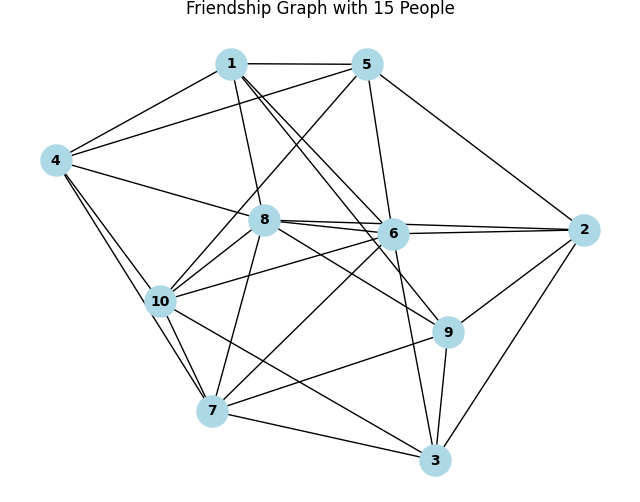
for each neighbour in neighbours of node:

if neighbour is not visited:

mark neighbour as visited

enqueue neighbour into Q

Rumour Spreading



This graph models 10 people as nodes, with their friendship connections as edges.

1. Assuming that nodes are chosen in numerical order as a tie-breaker, write down the order in which BFS and DFS would each choose nodes, starting with #1.

BFS: 1,4,5,6,8,10,5,6,2,3,7,9

DFS: 1,4,10,7,3,8,9,2,5,6

1. If a person spreads a rumour to all of their friends on a given day, which search pattern is more appropriate?

BFS (This will ensure that all close friends are systematically search before moving onto layer 2 and 3 instead of just going down one path indefinitely)

1. Write your own BFS code, using the template provided. (The DFS code is included, so you can get an idea of how it might be changed).

<https://trinket.io/python3/d0a8e9b22b>