# **Lab 7\_2 Graphs continued**

# **Week beginning Monday 20/11/2023**

Implement breadth-first traversal. Algorithm as given in slides:

bfs(start\_vertex)

mark all vertices as not\_visited   
  create an empty queue  
  queue.enqueue(start\_vertex) //add to queue  
  while not queue.empty do  
     vertex = queue.dequeue() // remove from queue  
     visit(vertex)  
 mark vertex as visited

for all vertices v adjacent to **vertex** such that v is not\_visited

queue.enqueue(v) //add to queue

mark vertex v as waiting

Questions on bfs() method:

(a). What is the header for the method?

(b). Include bfs() as method in Graph interface. In which class will you implement the method?

(c). The algorithm requires a queue. Java has a Queue interface. Use LinkedList as the implementation of the Queue interface. What methods of LinkedList will you use?

(d). In this algorithm, each vertex is marked as one of not\_visited, visited or waiting. Initially all vertices are marked as not\_visited. How will you implement this?