

Student Name : Aaron Baggot
Student Number: C22716399

DFS (using recursion) and BFS (using a queue) working on both adjacency lists graph and adjacency matrix graph

Graph Class

Represents a graph using an adjacency lists representation.
Contains inner class Node for linked list nodes representing edges.
Constructor reads graph data from a file and initializes adjacency lists.
Provides methods for displaying the graph, performing DFS and BFS traversals.
Includes methods for inserting elements, removing elements, and maintaining the heap structure.

GraphLists Class (Main Class)

- Contains the main method to execute the program.
- Creates an instance of the Graph class, reads graph data from a file, and displays the graph.
- Executes DFS and BFS traversals starting from a specified vertex.

The screen output includes the adjacency lists representation of the graph followed by the results of DFS and BFS traversals starting from a specified vertex.

```
adj[A] -> |G| 6| -> |F| 2| -> |B| 1| ->
adj[B] -> |E| 4| -> |D| 2| -> |C| 1| ->
adj[C] -> |E| 4| ->
adj[D] -> |F| 1| -> |E| 2| ->
adj[E] -> |L| 4| -> |G| 1| -> |F| 2| ->
adj[F] -> |L| 2| ->
adj[G] -> |L| 5| -> |J| 1| -> |H| 3| ->
adj[H] -> |I| 2| ->
adj[I] -> |K| 1| ->
adj[J] -> |M| 2| -> |L| 3| -> |K| 1| ->
adj[K] ->
adj[L] -> |M| 1| ->
adj[M] ->

Depth First Search:
A G L M J K H I F B E D C

Breadth First Search:
A G F B L J H E D C M K I
```

This output shows the adjacency lists representation of a graph and the traversal order for both DFS and BFS starting from vertex A.

GraphMatrix Class

- Reads graph data from a file.
- Provides methods for displaying the graph, performing DFS, and BFS traversals.
- Depth First Search (DFS):
 - Implemented recursively for adjacency matrix.
 - Marks visited vertices and prints traversal order.
- Breadth First Search (BFS):
 - Implemented using a queue for both representations.
 - Marks visited vertices and prints traversal order.
- Creates an instance of GraphMatrix by passing the graph file name to the constructor.
- Display the graph using the display() method.
- Execute DFS or BFS traversal by calling the respective method (DF() for DFS and BF() for BFS) and passing the starting vertex as an argument.
- The screen output includes the adjacency matrix or lists representation of the graph followed by the traversal order for DFS and BFS.

```
adj[1] = 0 0 0 0 0 0 0 0 0 0 0 0 0 0
adj[2] = 1 0 0 0 0 0 0 0 0 0 0 0 0 0
adj[3] = 0 1 0 0 0 0 0 0 0 0 0 0 0 0
adj[4] = 0 2 0 0 0 0 0 0 0 0 0 0 0 0
adj[5] = 0 4 4 2 0 0 0 0 0 0 0 0 0 0
adj[6] = 2 0 0 1 2 0 0 0 0 0 0 0 0 0
adj[7] = 6 0 0 0 1 0 0 0 0 0 0 0 0 0
adj[8] = 0 0 0 0 0 0 3 0 0 0 0 0 0 0
adj[9] = 0 0 0 0 0 0 0 2 0 0 0 0 0 0
adj[10] = 0 0 0 0 0 0 1 0 0 0 0 0 0 0
adj[11] = 0 0 0 0 0 0 0 0 1 1 0 0 0 0
adj[12] = 0 0 0 0 4 2 5 0 0 3 0 0 0 0
adj[13] = 0 0 0 0 0 0 0 0 0 2 0 1 0 0
```

Depth First Graph Traversal
Starting with Vertex D

```
DF just visited vertex D along edge @--D
DF just visited vertex E along edge D--E
DF just visited vertex F along edge E--F
DF just visited vertex L along edge F--L
DF just visited vertex M along edge L--M
DF just visited vertex G along edge E--G
DF just visited vertex H along edge G--H
DF just visited vertex I along edge H--I
DF just visited vertex K along edge I--K
DF just visited vertex J along edge G--J
```

Breadth First Graph Traversal
Starting with Vertex D

```
BF just visited vertex D along edge @--D
BF just visited vertex E along edge D--E
BF just visited vertex F along edge D--F
BF just visited vertex G along edge E--G
BF just visited vertex L along edge E--L
BF just visited vertex H along edge G--H
BF just visited vertex J along edge G--J
BF just visited vertex M along edge L--M
BF just visited vertex I along edge H--I
BF just visited vertex K along edge J--K
```

This implementation allows for efficient traversal of graphs represented either as adjacency lists or adjacency matrices, facilitating various graph analysis tasks.