

# LAB 5: GRAPH

## 1 Exercises

1. The "*graph1.txt*" file contains information of an **Adjacency matrix** (Table 1). Read the file and output the information of the corresponding **Adjacency list**.
2. The "*graph2.txt*" file contains information of an **Adjacency list** (Table 1). Read the file and output the information of the corresponding **Adjacency matrix**.

| Adjacency matrix  | Adjacency list |
|-------------------|----------------|
| 9                 | 9              |
| 0 0 1 0 0 1 0 0 0 | 2 2 5          |
| 0 0 0 0 0 0 1 0 0 | 1 6            |
| 0 0 0 0 0 0 1 0 0 | 1 6            |
| 0 0 0 0 1 0 0 0 0 | 1 4            |
| 0 0 0 0 0 1 0 0 0 | 1 5            |
| 0 0 0 1 0 0 0 1 0 | 2 3 7          |
| 0 0 0 0 0 0 0 0 0 | 0              |
| 0 0 1 0 0 0 0 0 1 | 2 2 8          |
| 0 0 0 0 0 0 0 0 0 | 0              |

Table 1: Adjacency matrix and corresponding Adjacency list

3. Implement functions to provide the following information of a given graph:
  - Directed or Undirected Graph.
  - The number of edges and number of vertices.
  - Degree of each vertices for undirected graph. In-degree and Out-degree for directed graph.
  - List of isolated vertices / leaf vertices. .
  - Is the given graph special: **Complete graph**, **Circular graph**, **Bigraph**, **Complete bigraph**.
  - The number of **Connected components**. How many of them are trees?
  - The number of **Cut vertices** and **Bridge edges**.
4. Generate a **Base undirected graph** from a given directed graph.
5. Generate a **Complement graph** from a given undirected graph, outputting the corresponding adjacency matrix.
6. Generate a **Converse graph** from a given directed graph, outputting the corresponding adjacency matrix.
7. Determined Euler cycle from a given graph using Hierholzer's algorithm.

8. Find the spanning tree of a given graph using:

- DFS traversal
- BFS traversal

9. Find the minimum spanning tree of a given graph using:

- Prim algorithm.
- Kruskal algorithm.

10. Verify the connection between 2 vertices of a given graph.

11. Find the shortest path between 2 vertices of a given graph using:

- Dijkstra algorithm
- Bellman-Ford algorithm
- Floyd-Warshall algorithm