CSC261: Artificial Intelligence

Godawari College

Itahari, Sunsari

Tutorial 1

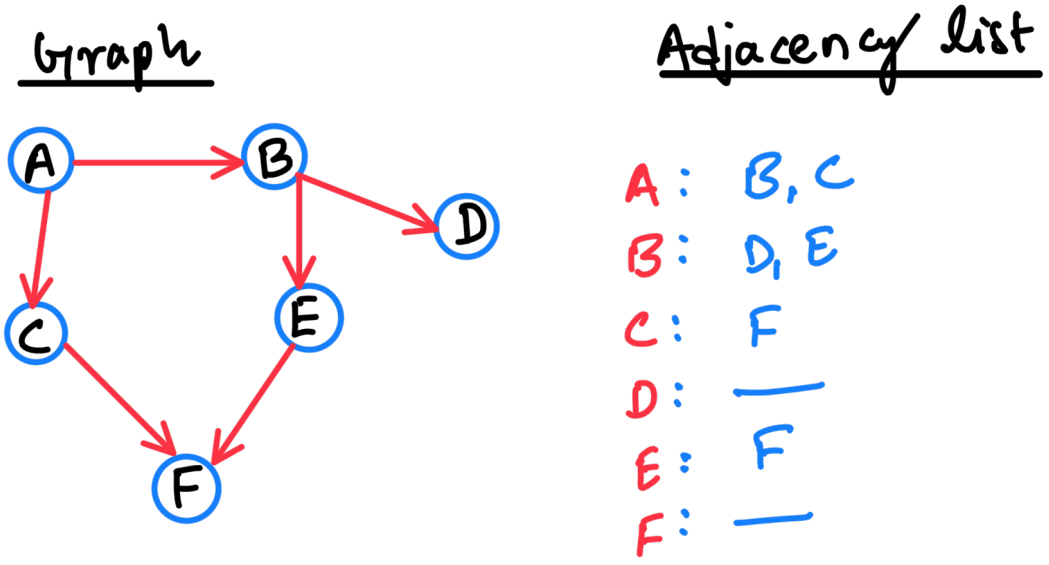
---

1. Graph:

   A Graph is a non-linear data structure consisting of nodes/vertex and edges and can be represented mainly in following ways.

   i. Adjacency List:

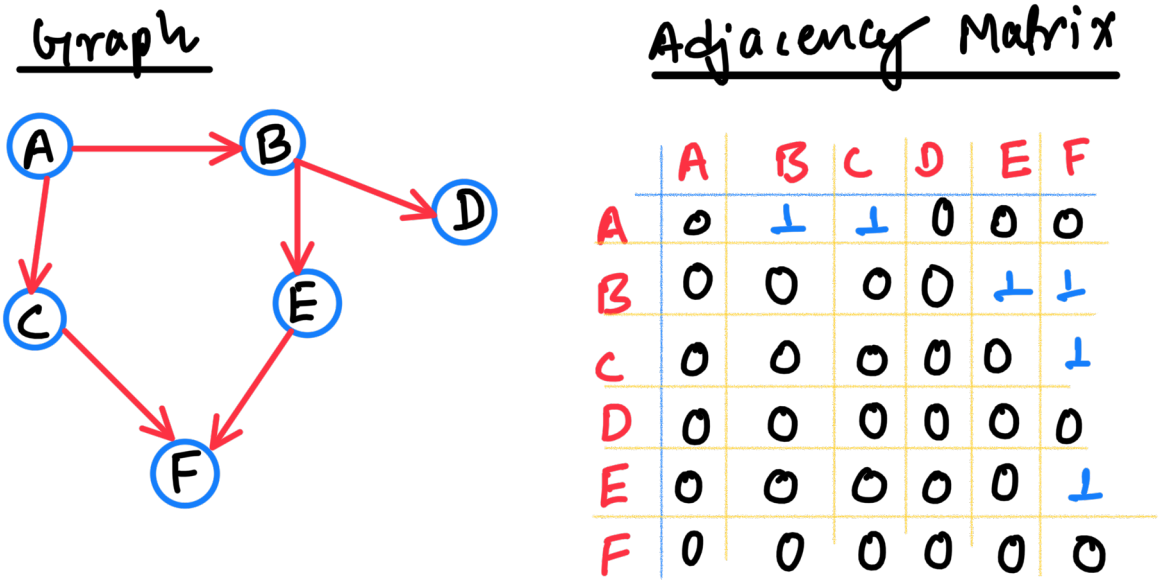
      An adjacency list is a collection of unordered lists used to represent a finite graph. Each list describes the set of neighbors of a vertex in the graph. In other words, for each vertex in the graph, we create a list of all the vertices that are connected to it by an edge. This representation is very space-efficient, especially for sparse graphs where the number of edges is much less than the number of vertices.



   ii. Adjacency Matrix:

      An adjacency matrix is a 2D matrix used to represent a graph. Each row and column of the matrix represents a vertex in the graph, and the entries in the matrix indicate whether there is an edge between the two vertices represented by that row and column. If there is

an edge between two vertices, the corresponding entry in the matrix is set to 1, otherwise, it is set to 0. This representation is useful for dense graphs where the number of edges is close to the number of vertices.



2. Given a graph represented by an adjacency list, implement a function (using any high level language; preferably Python) that performs a breadth-first search and returns the path from the starting node to the goal node, if there exist.
   i. Implementation:
   https://colab.research.google.com/drive/1TERjAT_6FxfW1zrqkO2aHrYGqdnBr-Rv?usp=sharing