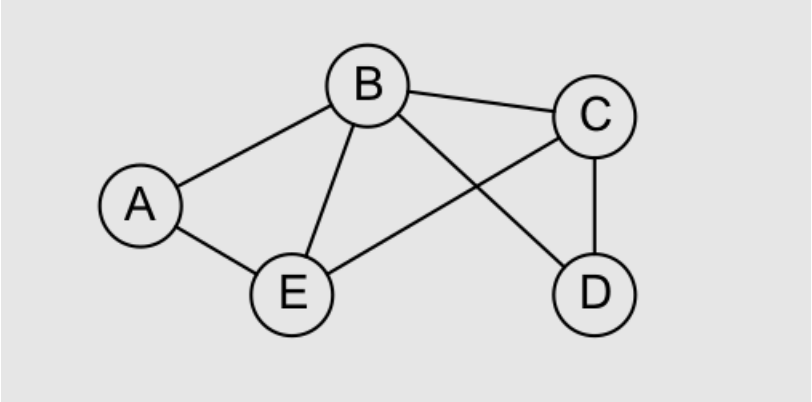
[CSL202] **2024-25-M**

TUTORIAL VI

Date: **Oct 18, 2024**.

1. Consider the graph G given below.



   (a) Write the adjacency matrix representation of G.

   (b) Write the adjacency-list representation of G.

   (c) Is G bipartite? justify your answer.

   (d) Is the graph connected?

   (e) Is the graph simple?

   (f) What is the distance between the vertices A and D.

2. In the above graph, perform a BFS from vertex E by giving preference for visiting lower-character vertices before higher-character vertices. Draw the BFS tree.

3. Find the time required for the following basic operations/tasks on a directed graph $G = (V, E)$, given (i) adjacency-list representation of G (ii) adjacency matrix representation of G.

   (a) Test if $uv \in E(G)$.

   (b) List v's out-neighbors.

   (c) Insert an edge uv.

(d) Delete an edge uv.

(e) List all the edges.

4. Show that a graph G is bipartite if and only if G has no odd length cycles.

5. The *transpose* of a directed graph $G = (V, E)$ is the graph $G^T = (V, E^T)$, where $E^T = \{(v, u) \mid (u, v) \in E(G)\}$. Thus, G^T is G with all its edges reversed. Describe efficient algorithms for computing G^T from G, for both the adjacency-list and adjacency-matrix representations of G. Analyze the running times of your algorithms.

6. The *square* of a directed (undirected) graph $G = (V, E)$ is the graph $G^2 = (V, E^2)$ such that $(u, v) \in E(G^2)$ if and only G contains a path with at most two edges between u and v.

   (a) For the graph G given in Q1, find G^2.

   (b) Describe efficient algorithms for computing G^2 from G for both the adjacency-list and adjacency-matrix representations of G. Analyze the running times of your algorithms.