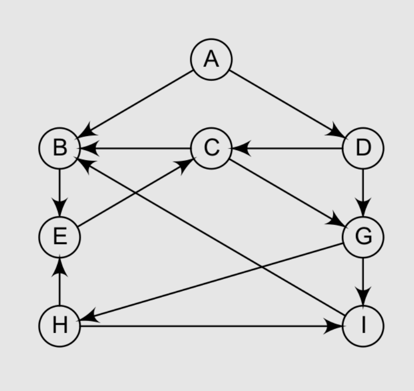
[CSL202] **2024-25-M**

TUTORIAL VII

Date: **Oct 25, 2024**.

1. Perform depth-first search on the following directed graph G: start from the node A. Give a preference for visiting lower-numbered vertices before higher-numbered vertices. (a) Write down the discover and finish times, (b) Draw the DFS forest (c) Classify the edges (d) Is the graph strongly connected? If No, find the strongly connected components (SCCs) (e) Find the topological ordering (sort) of G (if exists).



2. TRUE/FALSE:

   (a) Every directed graph G has exactly one topological ordering.

   (b) Every directed acyclic graph (DAG) G has exactly one topological ordering.

   (c) If a directed graph G has a topological ordering then G is a DAG.

   (d) Every directed acyclic graph has a vertex with no incoming edges.

   (e) The number of strongly connected components of a graph always decreases by one, if a new edge is added to G.

3. TRUE/FALSE:

   (a) If a directed graph G contains a path from u to v and if $u.d < v.d$ in a depth-first search of G, then v is a descendant of u in the depth-first forest produced.

(b) A vertex u of a directed graph can end up in a depth-first tree containing only u, even though u has both incoming and outgoing edges in G.

(c) If a directed graph G contains a path from u to v, then any depth-first search must result in $v.d \leq u.f$.

4. Give an algorithm to detect whether a given directed graph contains a cycle. If the graph contains a cycle, then your algorithm should output one. (It should not output all cycles in the graph, just one of them.) The running time of your algorithm should be $O(m + n)$ for a graph with n nodes and m edges.