Tutorial 10.

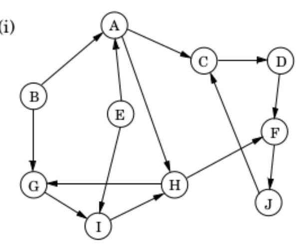
1. Compute the strongly connected components of graphs (i) and (ii) below.

2. Use the graph reversal algorithm to detect if (i) and (ii) are strongly connected.

3. Recall the definition of strong connectedness. Let us define v~w iff there is a path from v to w and a path from w to v. This is an equivalence relation on vertices. Let [v1]…[vk] be the equivalence classes of ~. We now define a new graph G'(V',E'), where V'={ [v1],...,[vk]}. We say ([vi],[vj]) is an edge in the new graph iff there is a path from vi to vj in the original graph.
   (A) Show that this new graph G' has no cycles.
   (B) Show that the edge ([vi],[vj]) does not depend on the representative vi of [vi].
   (C) Compute G' for the two example graphs.

4. Run the smallest arrival time algorithm for the example graphs for DFS starting at A and proceeding lexicographically. If you have not exhausted all vertices, start at the next unvisited vertex in lexicographic order. Record this time in a table.

(i)



(ii)