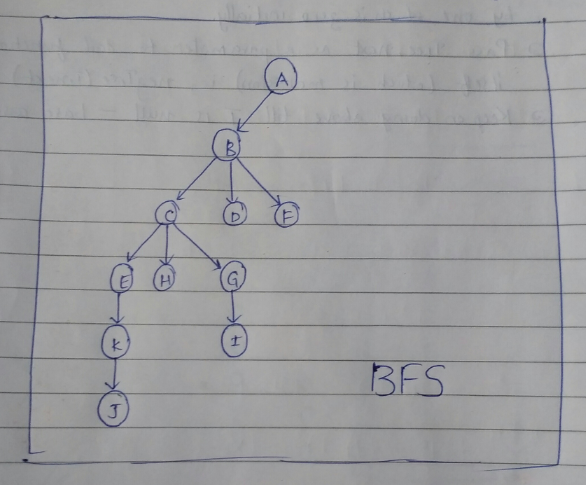
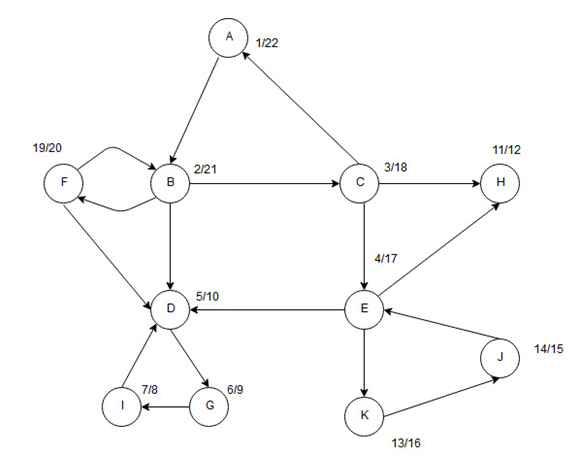
Bedemariam Degef

Algorithms

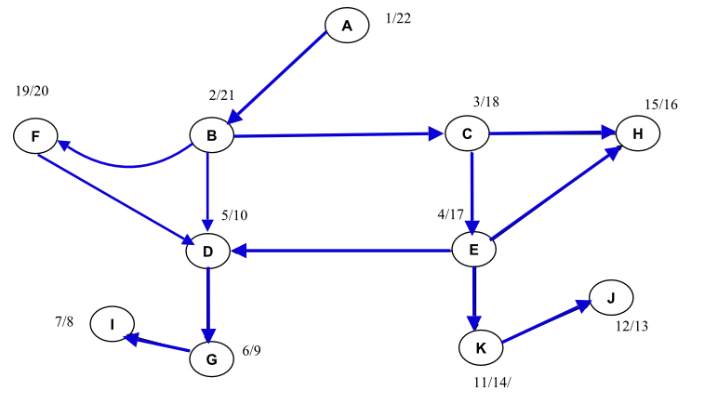
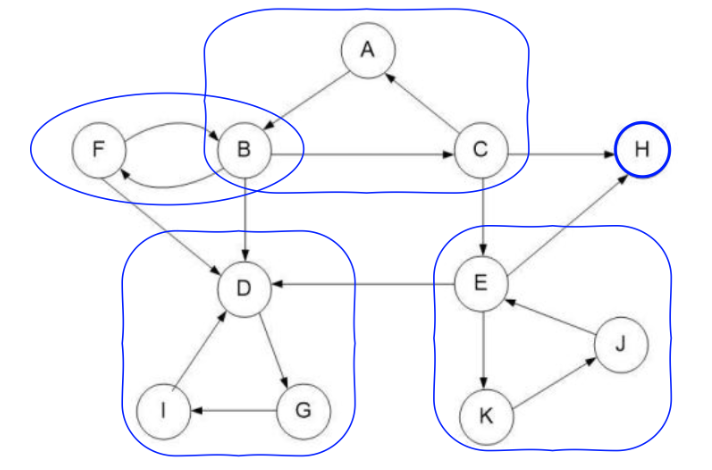
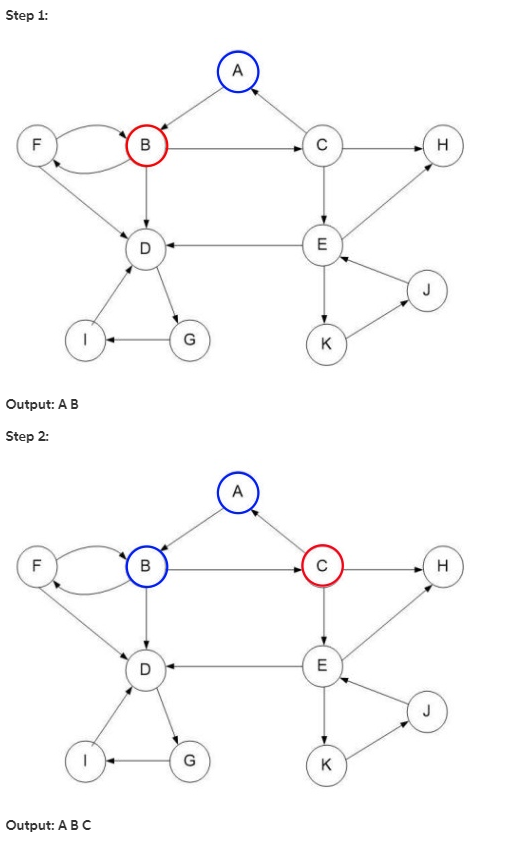
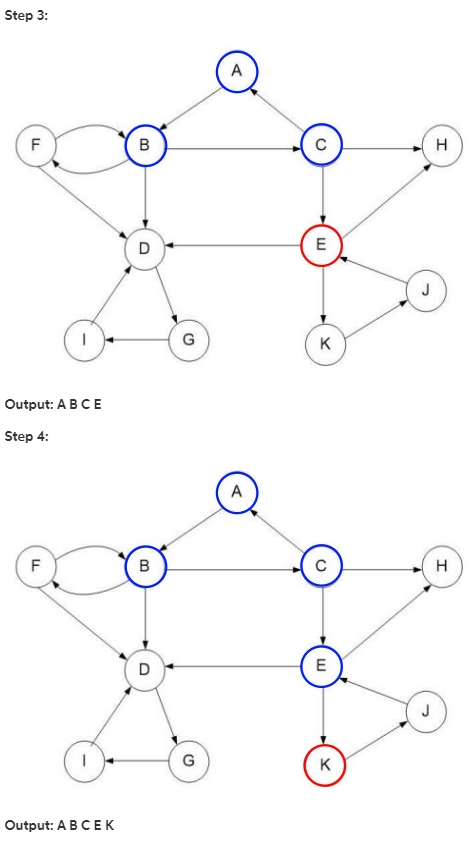
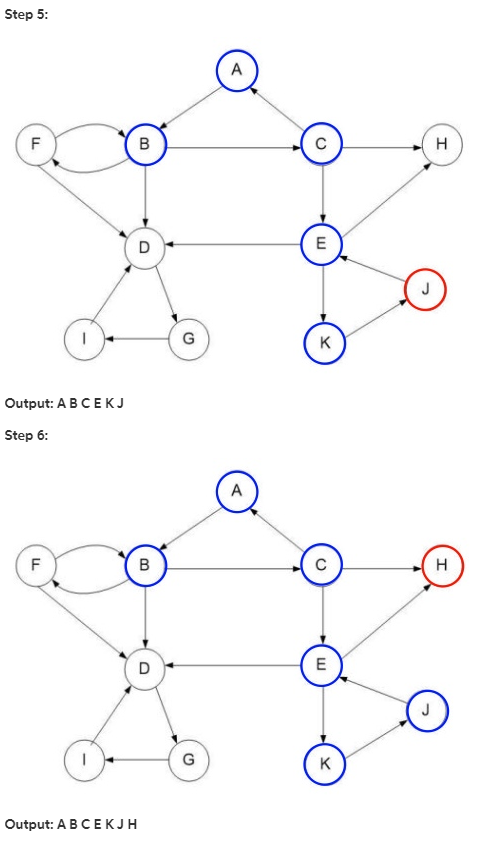
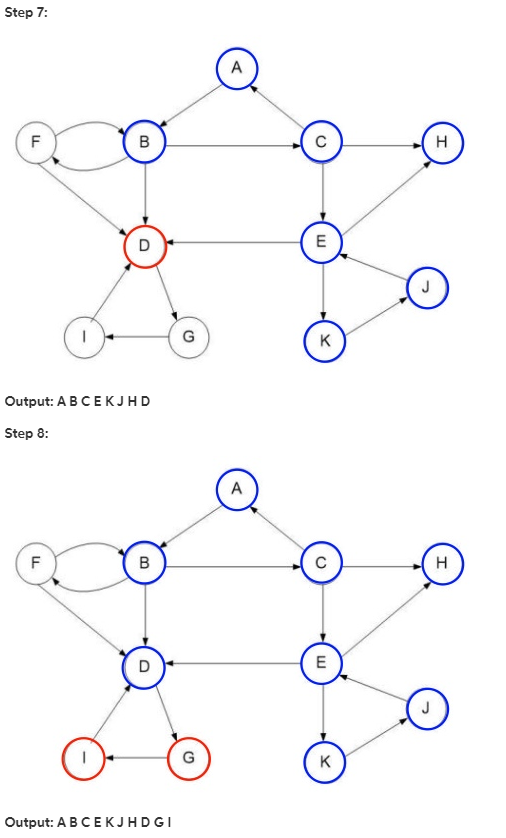
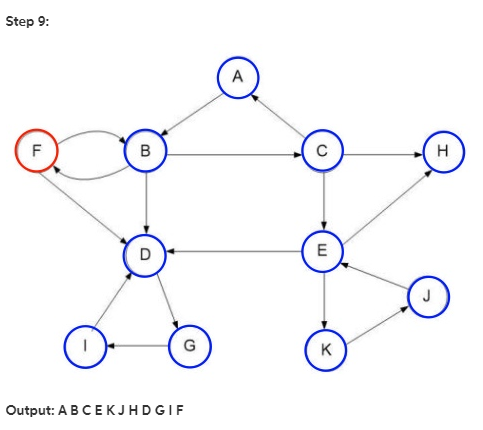
Homework 4

* 1. Vertex A is starting vertex. We enqueue and visit. Queue is A.
  2. Pop vertex A. Enqueue and visit non-visited adjacent nodes of A. Queue is B.
  3. Pop vertex B. Enqueue and visit non-visited adjacent nodes of B. Queue is C, D, F.
  4. Pop vertex C. Enqueue and visit non-visited adjacent nodes of C. Queue is D, F, E, H.
  5. Pop vertex D. Enqueue and visit non-visited adjacent nodes of D. Queue is F, E, H, G.
  6. Pop vertex F. No non-visited adjacent nodes of F. Queue is E, H, G.
  7. Pop vertex E. Enqueue and visit non-visited adjacent nodes of E. Queue is H, G, K.
  8. Pop vertex H. No non-visited adjacent nodes of H. Queue is G, K.
  9. Pop vertex G. Enqueue and visit non-visited adjacent nodes of G. Queue is K, I.
  10. Pop vertex K. Enqueue and visit non-visited adjacent nodes of K. Queue is I, J.
  11. Pop vertex I. No non-visited adjacent nodes of I. Queue is J.
  12. Pop vertex J. No non-visited adjacent nodes of J. End of traversal  
        
      Breadth-First Search Tree:  
      
  13. **Tree Edge**: During the traversal of edge (x, y) if y is visited for the first time, then the edge is called tree edge.
  14. **Back Edge**: During the traversal of edge (x, y) if y is already visited and y is x’s ancestor, then the edge is called back edge
  15. **Forward Edge**: During the traversal of edge (x, y) if y is already visited and y is x’s descendant, then the edge is called forward edge.
  16. **Cross Edge**: During the traversal of edge (x, y) if y is already visited but y is neither x’s descendant nor ancestor, then the edge is called cross edge.

|  |  |  |
| --- | --- | --- |
| **Type of Edge** | **Criteria for Start Time** | **Criteria for End Time** |
| Tree Edge | Start(x) < Start(y) | End(x) > End(y) |
| Back Edge | Start(x) > Start(y) | End(x) < End(y) |
| Forward Edge | Start(x) < Start(y) | End(x) > End(y) |
| Cross Edge | Start(x) > Start(y) | End(x) > End(y) |

DFS of graph with each vertex start/end time:  
 

Tree Edges   
 Back Edges   
 Forward Edges   
 Cross Edges

* 1. The Depth First Search of the Graph with the Back Edges removed and labeled with start and finish times (the DAG) is shown below:  
     Enumerating the vertices as they finish:  
     **I, G, D, J, K, H, E, C, F, B, A  
       
     If we reverse the above enumeration, we get our Depth-First Topological order:  
     A, B, F, C, E, H, K, J, D, G, I**  
       
     A breadth-first topological order for the graph is, as we solved in question 1):  
     **A, B, C, D, F, E, H, G, I, K, J**
  2. The strongly connected components of the graph are as follows:  
     
  3. The final Depth First Search of the transpose graph is as follows:  
       
       
       
       
       
     The DFS is **A, B, C, E, K, J, H, D, G, I, F**