

Ashcon Abae

Algorithms

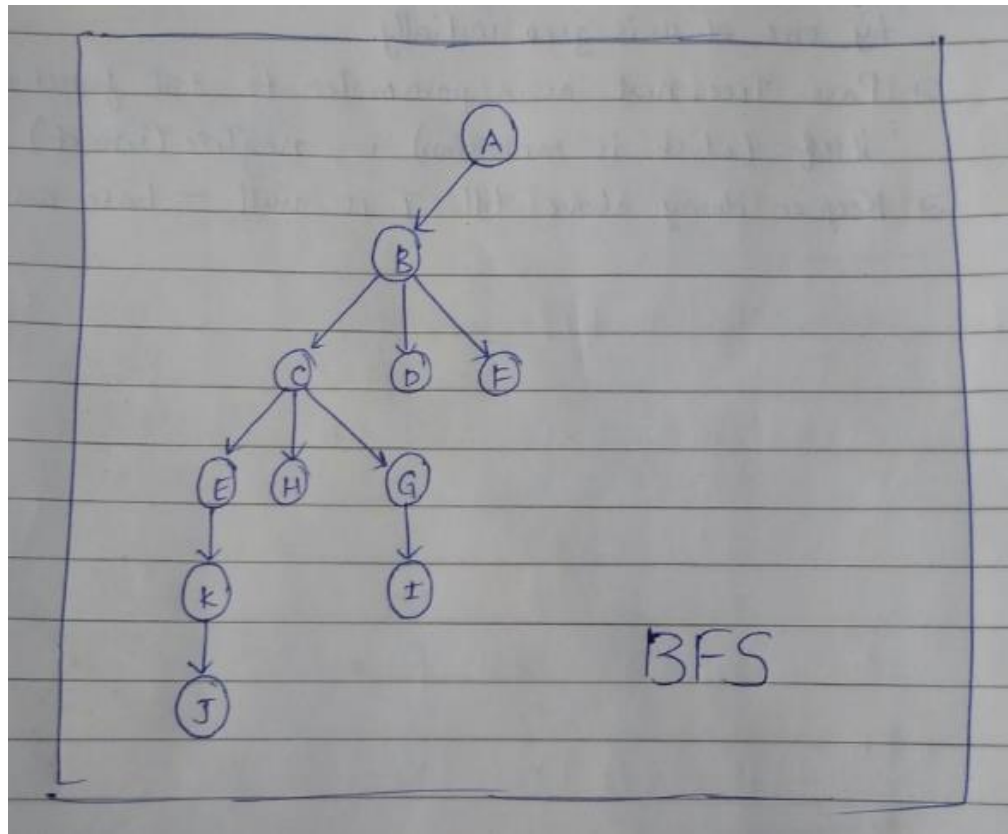
24 November 2019

Homework 4

1.

- a. Vertex A is starting vertex. We enqueue and visit. Queue is A.
- b. Pop vertex A. Enqueue and visit non-visited adjacent nodes of A. Queue is B.
- c. Pop vertex B. Enqueue and visit non-visited adjacent nodes of B. Queue is C, D, F.
- d. Pop vertex C. Enqueue and visit non-visited adjacent nodes of C. Queue is D, F, E, H.
- e. Pop vertex D. Enqueue and visit non-visited adjacent nodes of D. Queue is F, E, H, G.
- f. Pop vertex F. No non-visited adjacent nodes of F. Queue is E, H, G.
- g. Pop vertex E. Enqueue and visit non-visited adjacent nodes of E. Queue is H, G, K.
- h. Pop vertex H. No non-visited adjacent nodes of H. Queue is G, K.
- i. Pop vertex G. Enqueue and visit non-visited adjacent nodes of G. Queue is K, I.
- j. Pop vertex K. Enqueue and visit non-visited adjacent nodes of K. Queue is I, J.
- k. Pop vertex I. No non-visited adjacent nodes of I. Queue is J.
- l. Pop vertex J. No non-visited adjacent nodes of J. End of traversal

Breadth-First Search Tree:

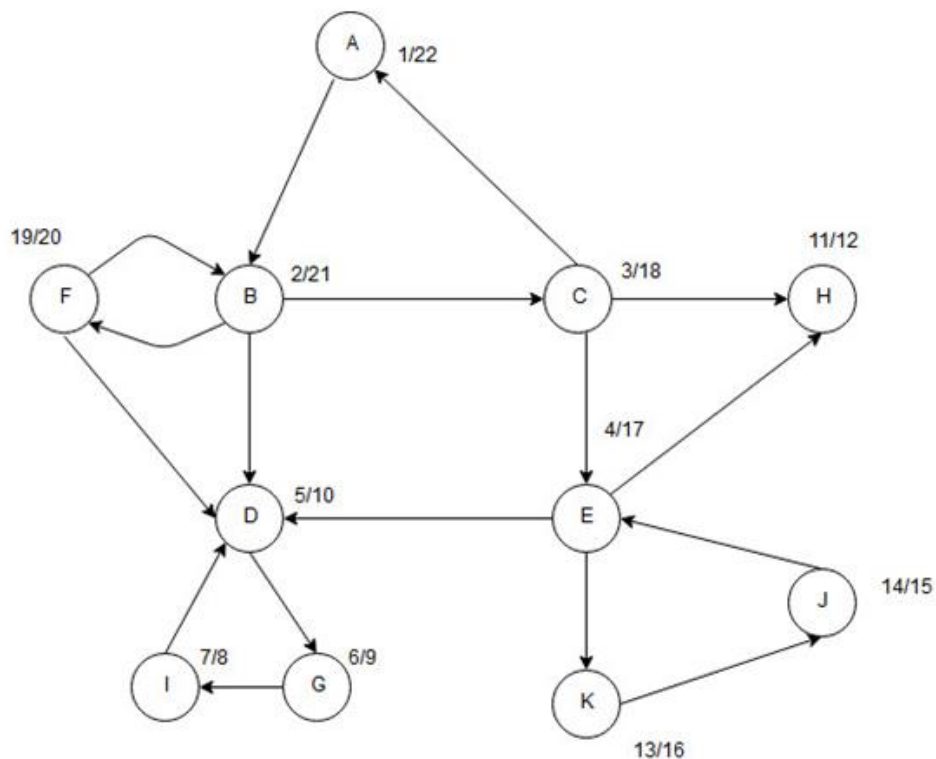


2.

- Tree Edge:** During the traversal of edge (x, y) if y is visited for the first time, then the edge is called tree edge.
- 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.
- 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.
- 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	$\text{Start}(x) < \text{Start}(y)$	$\text{End}(x) > \text{End}(y)$
Back Edge	$\text{Start}(x) > \text{Start}(y)$	$\text{End}(x) < \text{End}(y)$
Forward Edge	$\text{Start}(x) < \text{Start}(y)$	$\text{End}(x) > \text{End}(y)$
Cross Edge	$\text{Start}(x) > \text{Start}(y)$	$\text{End}(x) > \text{End}(y)$

DFS of graph with each vertex start/end time:



Tree Edges = $\{(A, B), (B, C), (B, D), (B, F), (C, E), (C, H), (E, K), (K, J), (D, G), (G, I)\}$

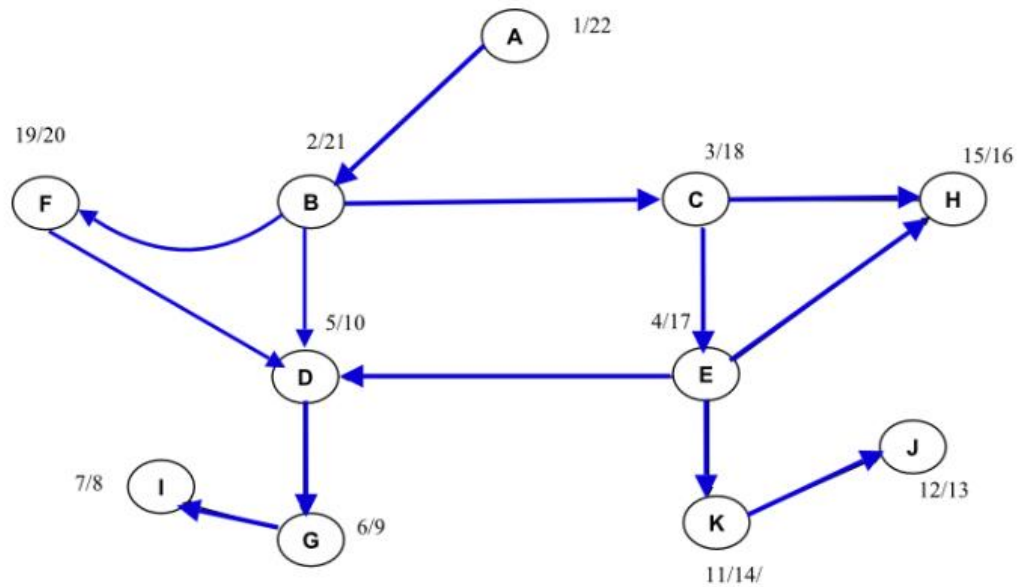
Back Edges = $\{(F, B), (I, D), (J, E)\}$

Forward Edges = $\{(E, H)\}$

Cross Edges = $\{(F, D)\}$

3.

- a. 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:

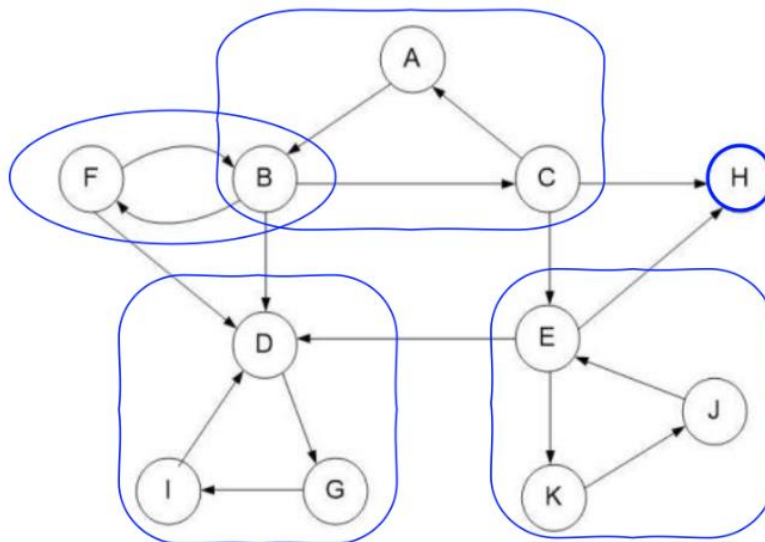
A, B, F, C, E, H, K, J, D, G, I

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

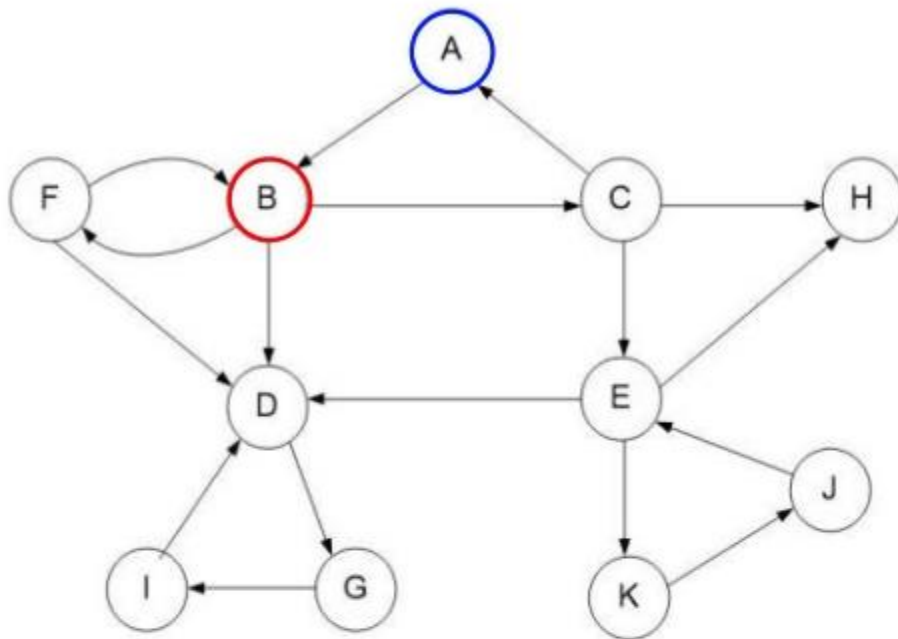
4.

- a. The strongly connected components of the graph are as follows:



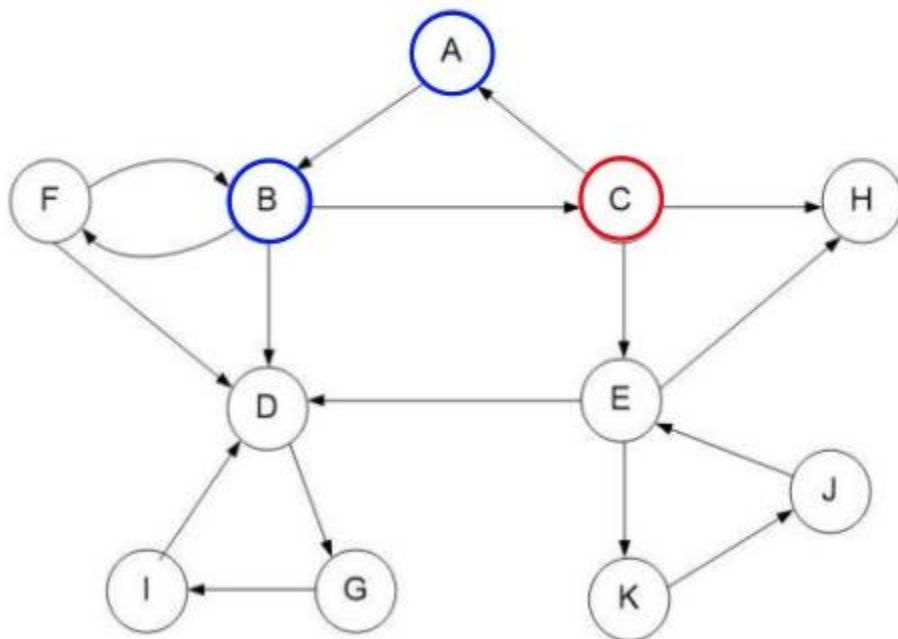
b. The final Depth First Search of the transpose graph is as follows:

Step 1:



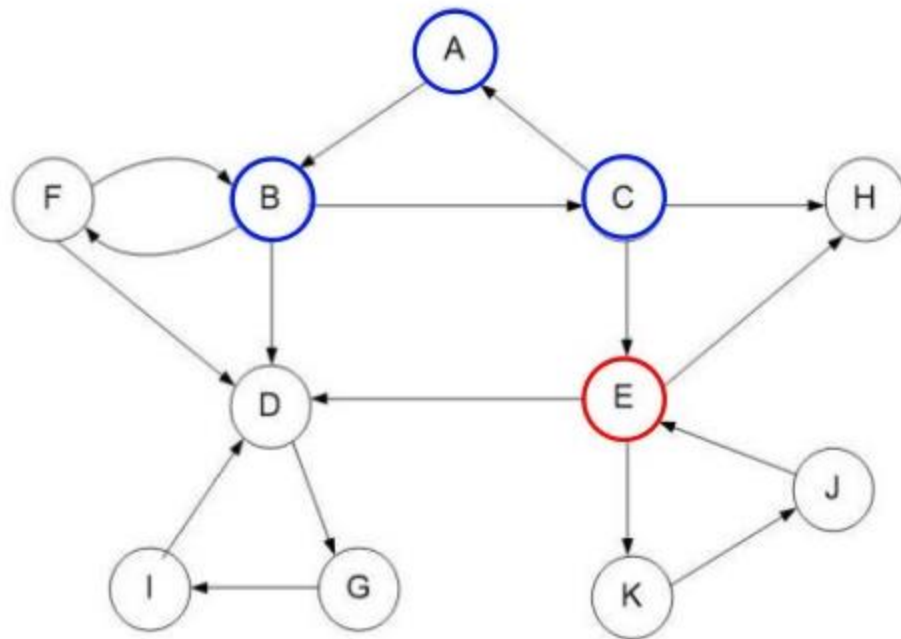
Output: A B

Step 2:



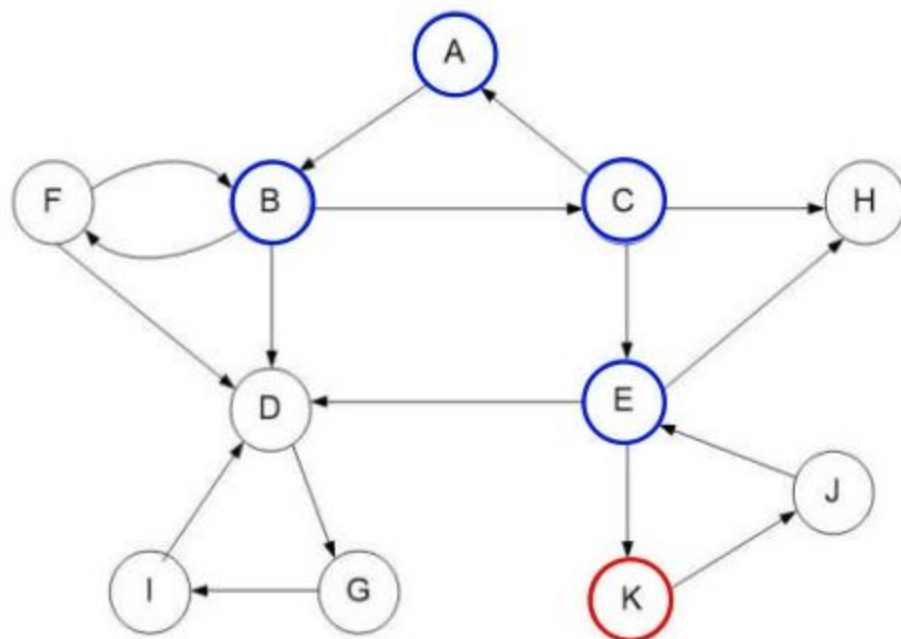
Output: A B C

Step 3:



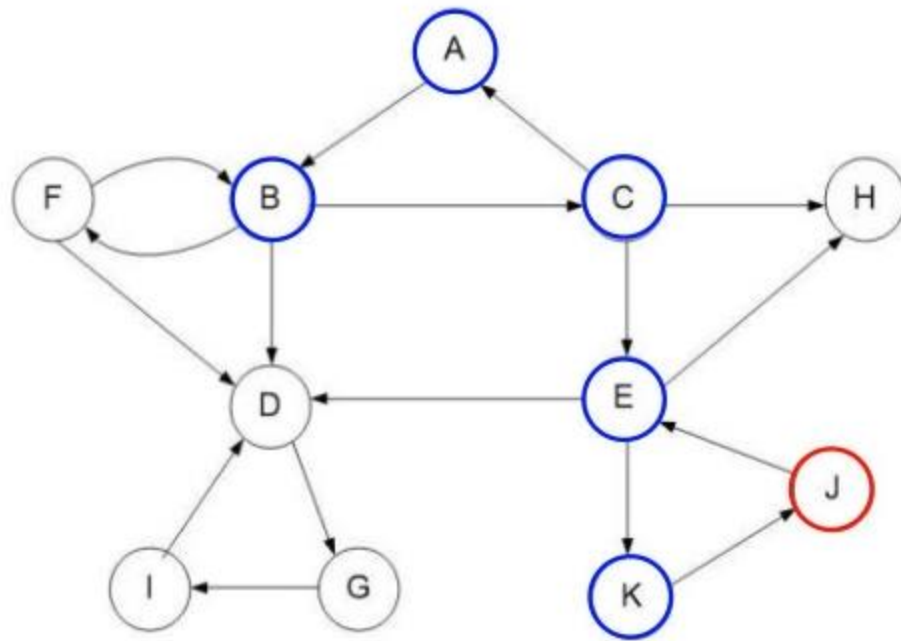
Output: A B C E

Step 4:



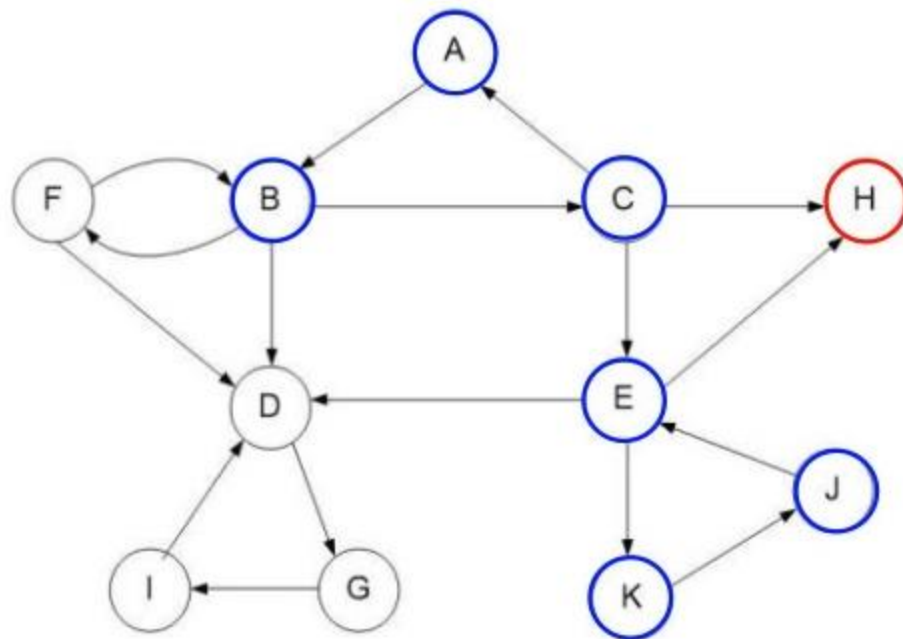
Output: A B C E K

Step 5:



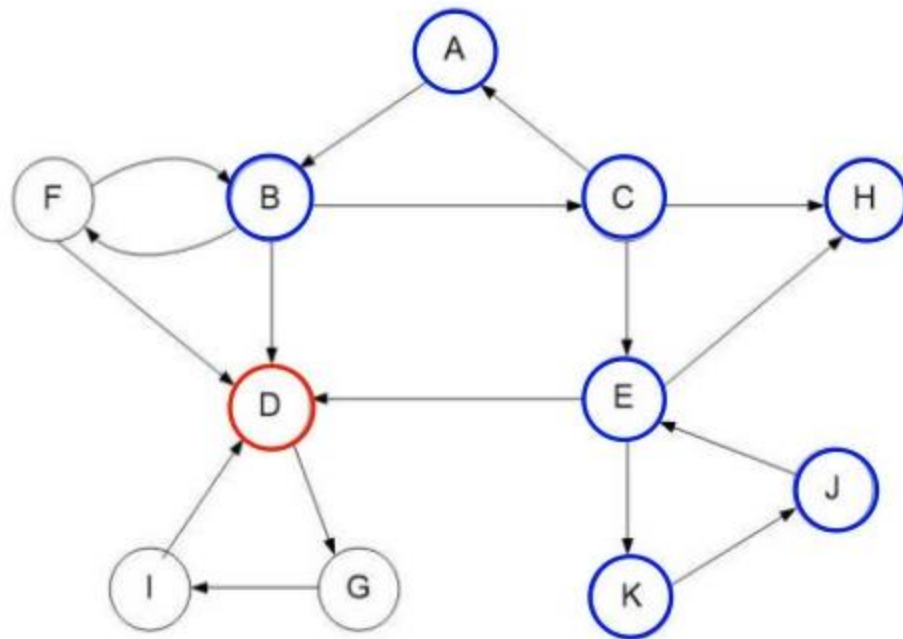
Output: A B C E K J

Step 6:



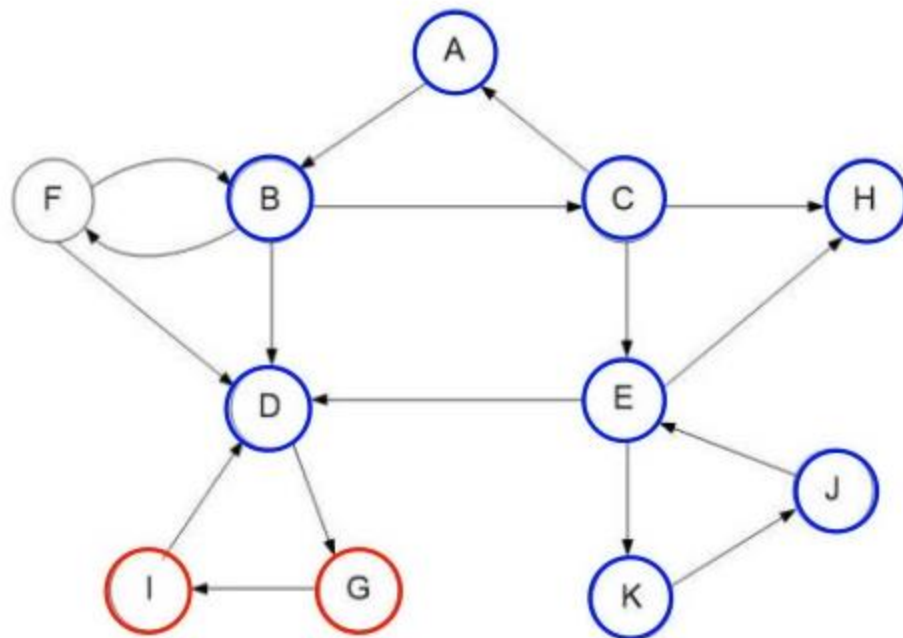
Output: A B C E K J H

Step 7:



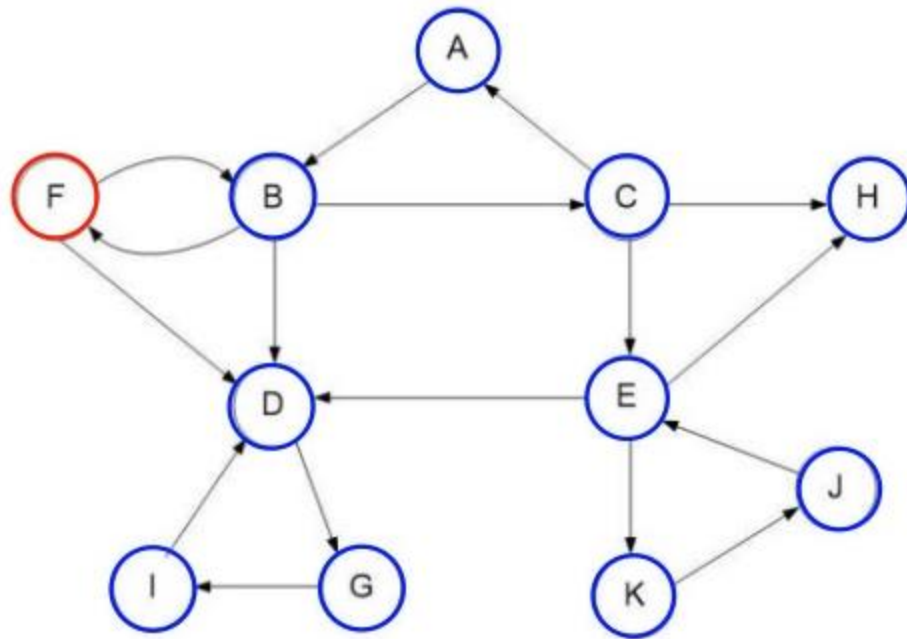
Output: A B C E K J H D

Step 8:



Output: A B C E K J H D G I

Step 9:



Output: A B C E K J H D G I F

The DFS is A, B, C, E, K, J, H, D, G, I, F