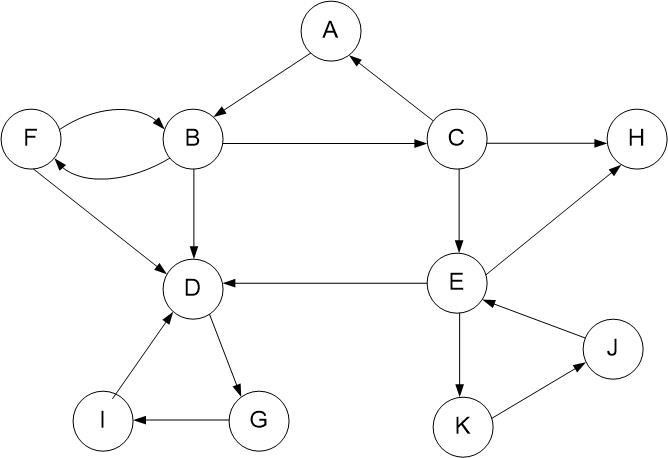
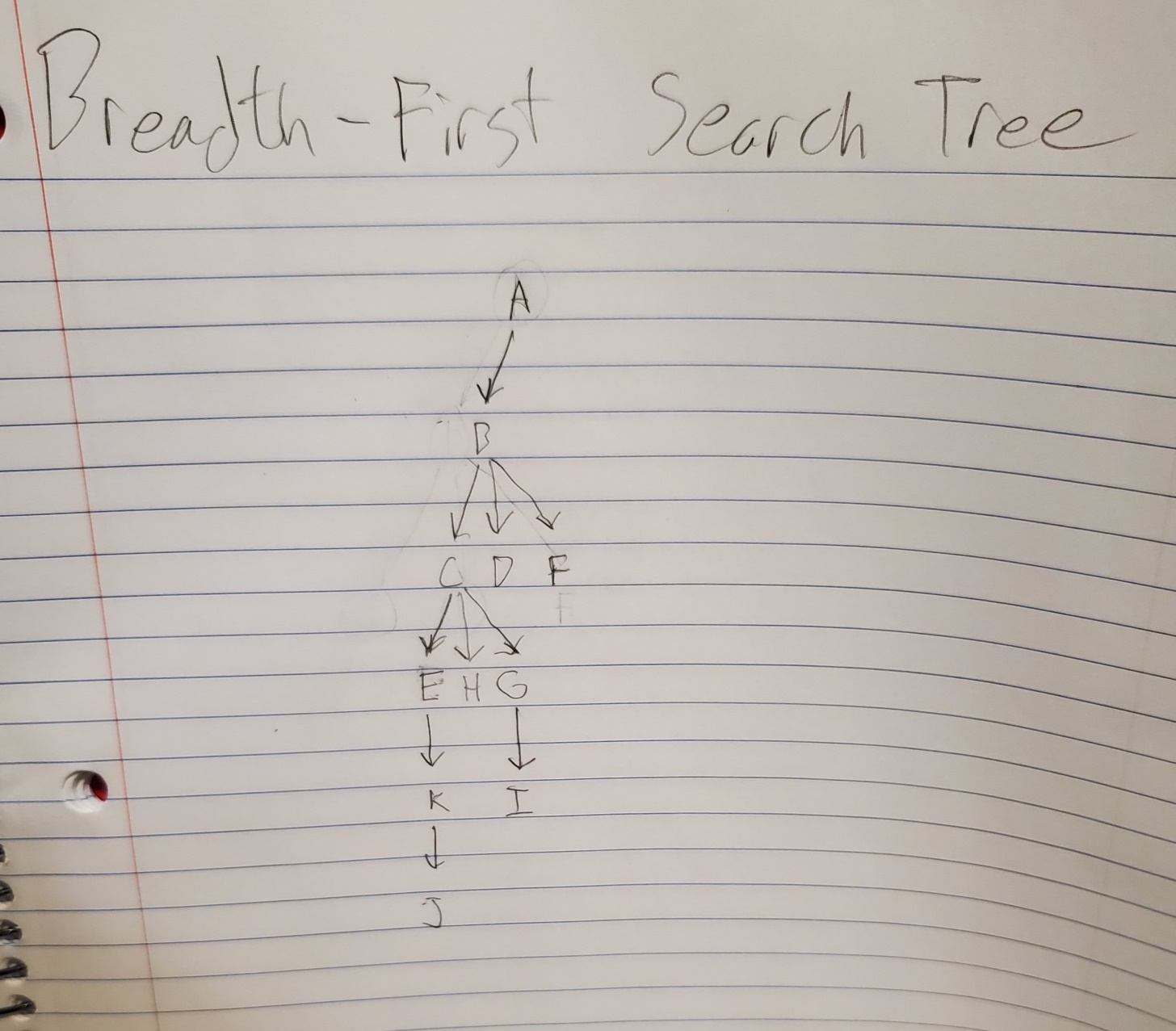
# CMSC 451 Homework 4

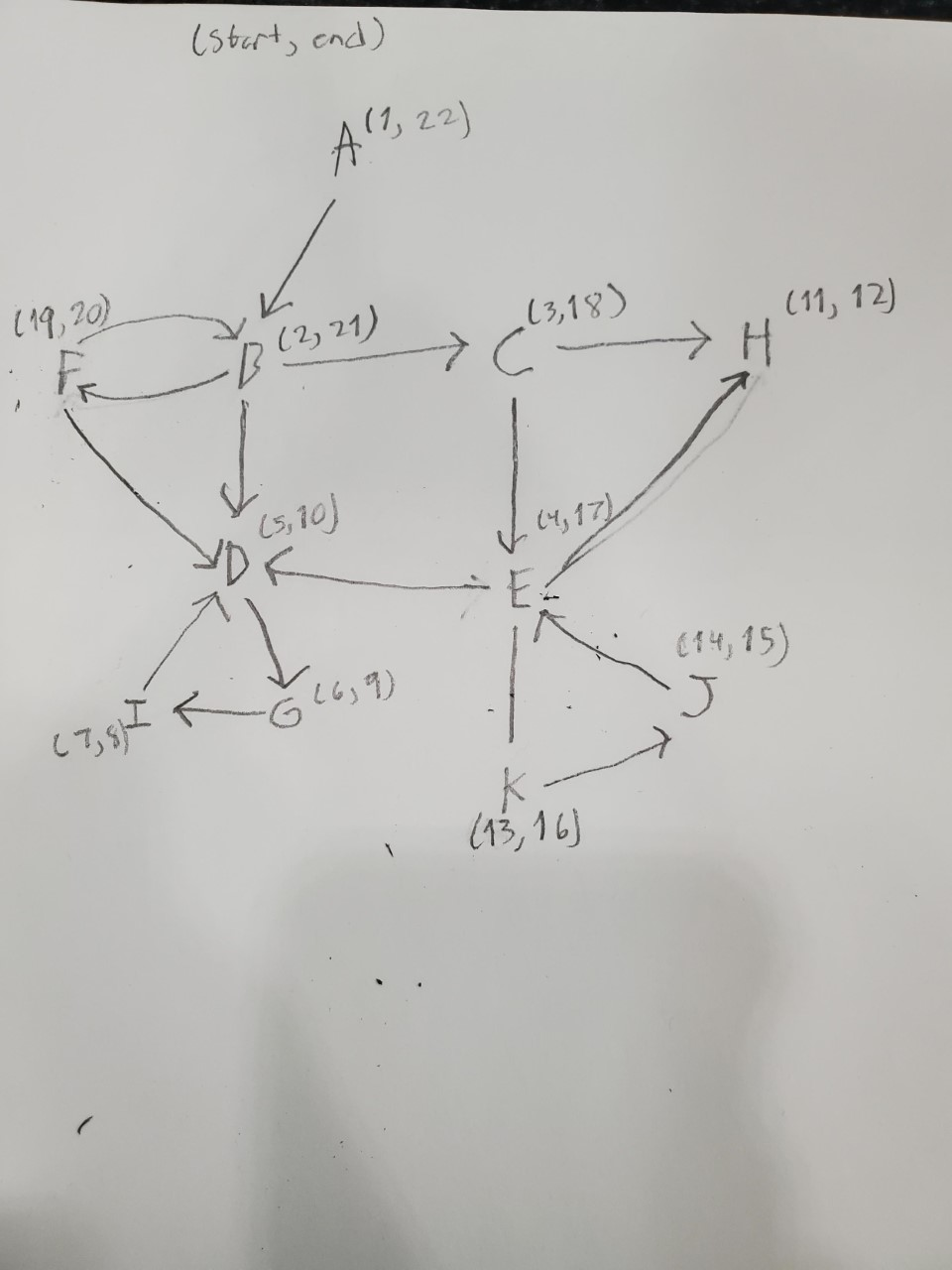
Consider the following directed graph for each of the problems:



1. Perform a breadth-first search on the graph assuming that the vertices and adjacency lists are listed in alphabetical order. Show the breadth-first search tree that is generated.
2. We start with vertex A, which is the current queue.
3. Pop vertex A. Enqueue and visit unvisited adjacent nodes of A. Our queue now lands on B.
4. Pop vertex B. Enqueue and visit unvisited adjacent nodes of B. Our queues are now C, D, and F.
5. Pop vertex C. Enqueue and visit unvisited adjacent nodes of C, resulting in queues of D, F, E, and H.
6. Pop vertex D. Enqueue and visit unvisited adjacent nodes of D. We now have F, E, H, and G as our queue.
7. Pop vertex F. There are no unvisited adjacent nodes of F to be visited as those nodes have already been encompassed. E, H, and G are our queues.
8. Pop vertex E. Enqueue and visit unvisited adjacent nodes of E. H, G, and K are the queues now.
9. Pop vertex H. We have no unvisited adjacent nodes for H. The queues are G and K.
10. Pop vertex G. Enqueue and go through unvisited adjacent nodes of G. Now, our queues are K, and I.
11. Pop vertex K. Enqueue and go through unvisited adjacent nodes of K. I and J are our queues.
12. Pop vertex I. We don’t have any unvisited nodes of I. J is the queue.
13. Pop vertex J. We have no more unvisited nodes of the tree, as we have reached the end of our traversal.



1. Perform a depth-first search on the graph assuming that the vertices and adjacency lists are listed in alphabetical order. Classify each edge as tree, forward, back or cross edge. Label each vertex with its start and finish time.



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

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

Forward Edge = { E, H) }

Cross Edge = { F, D) }

1. Remove all the back edges from the graph so it becomes a DAG. Perform a depth-first search recording the start and finish times. Using those finish times, provide the topological order that is produced. Provide one breadth-first topological order for that graph.

Counting the vertices that finish:

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

Depth-First Topological order:

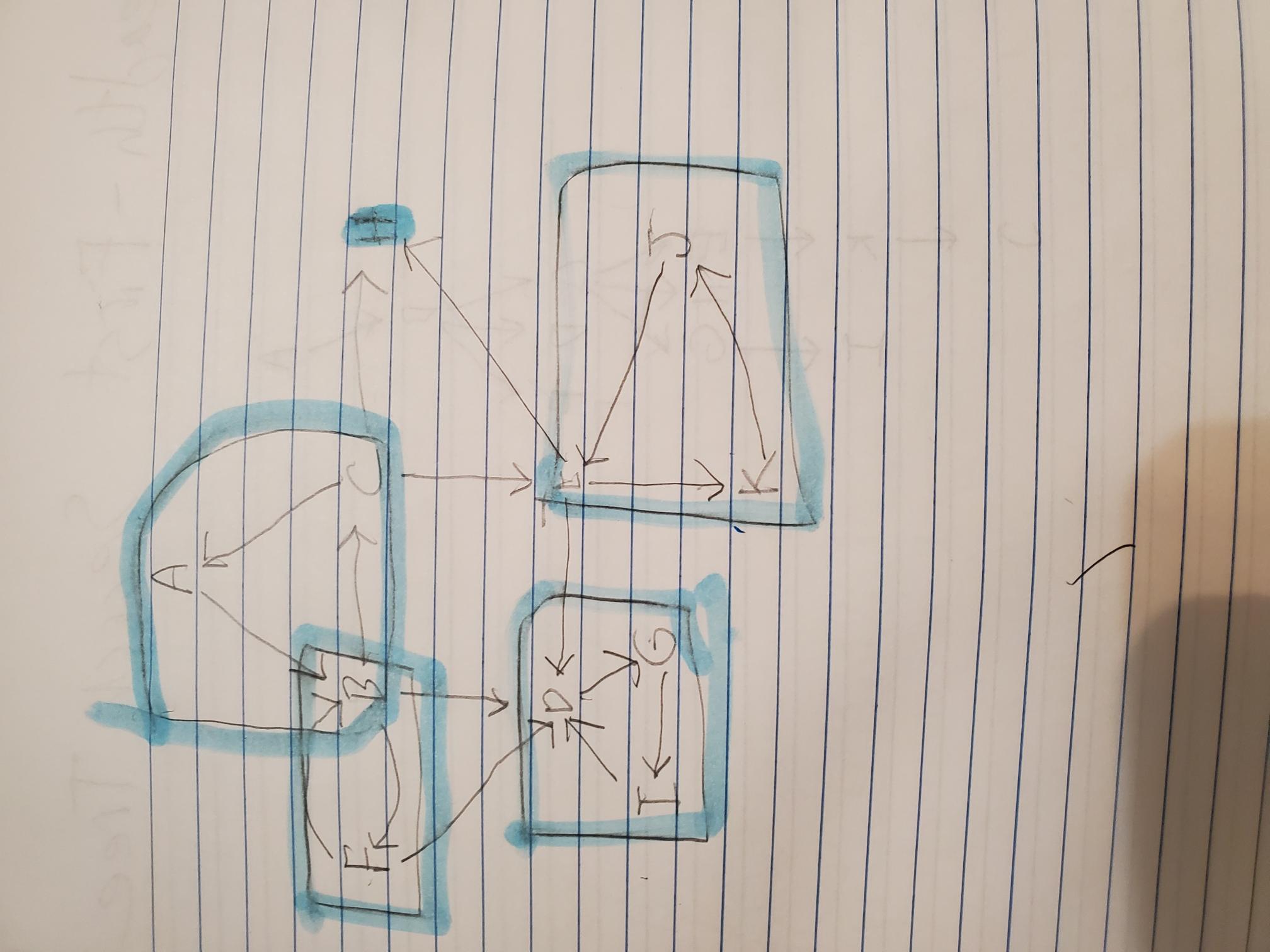
A, B, F, C, E, H, K, J, D, G, I

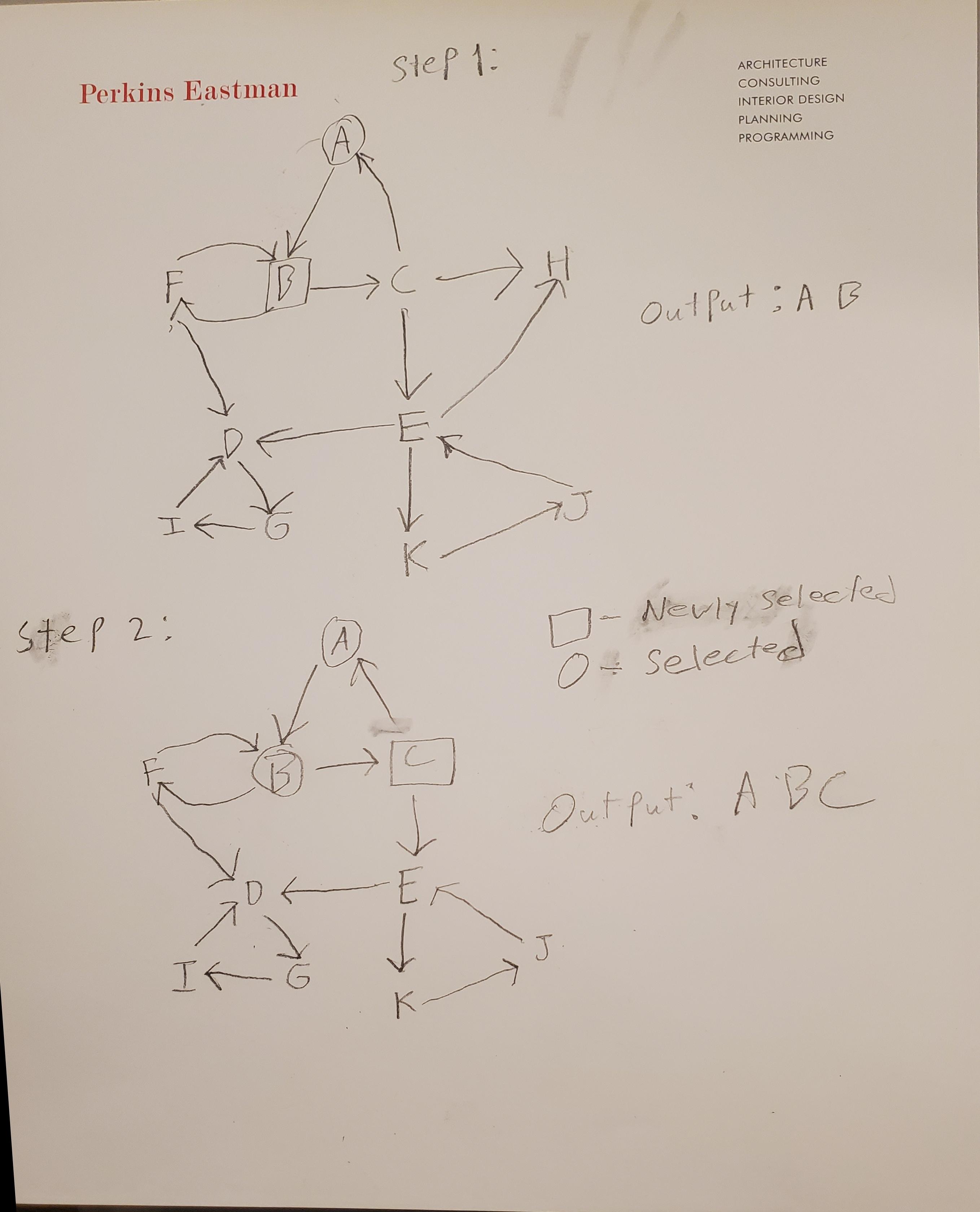
Breadth-First Topological Order:

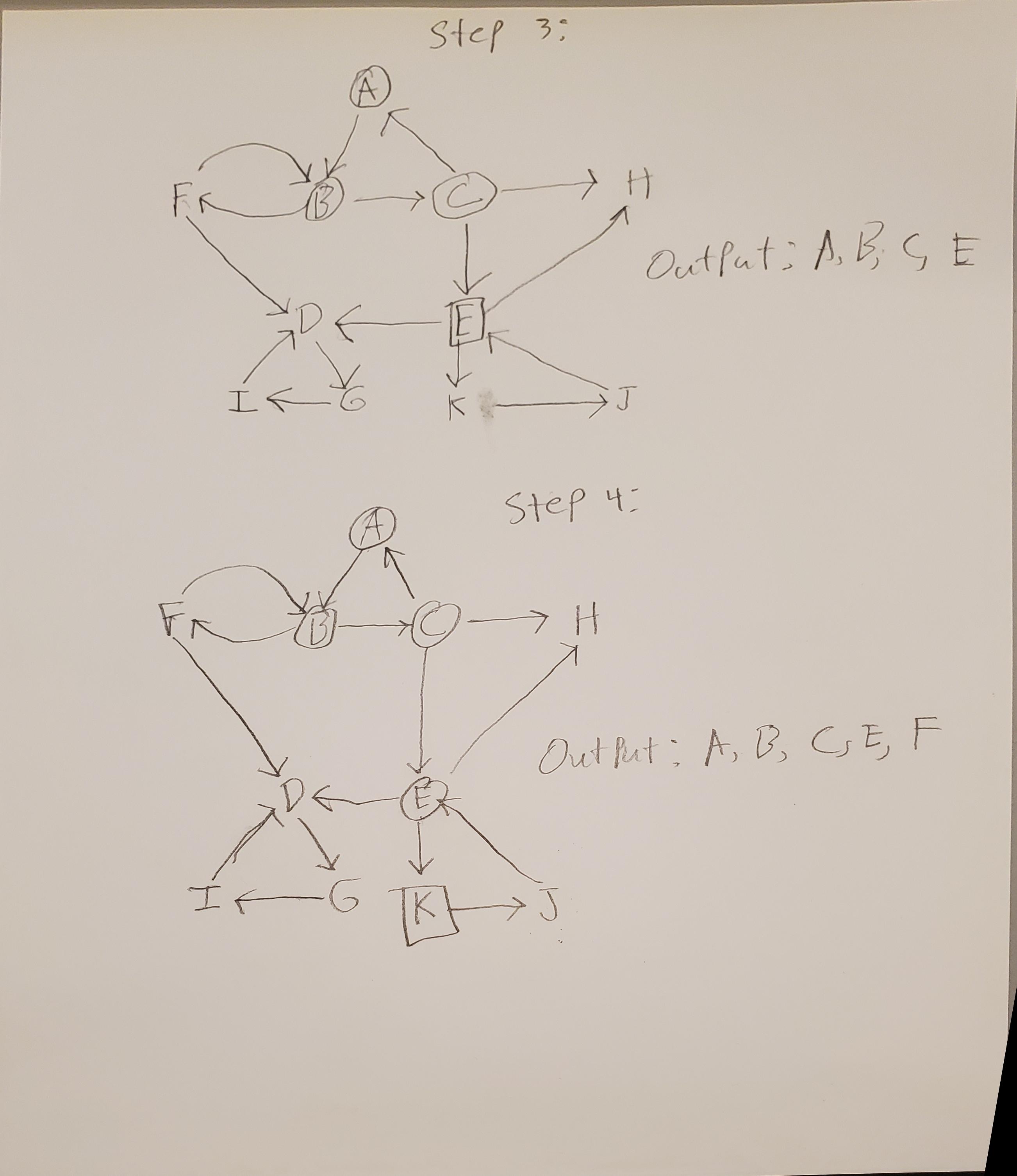
A, B, C, D, F, E, H, G, I, K, J

1. Determine the strongly connected components of the graph using the algorithm provided in the sample problems. Show the final depth-first search of the transpose graph labeled with its start and finish times. Identify the strongly connected components based on that search.

Strongly Connected Components of the Graph







# Grading Rubric

|  |  |  |
| --- | --- | --- |
| **Problem** | **Meets** | **Does Not Meet** |
| **Problem 1** | **10 points** | **0 points** |
|  | |
| Breadth-first search was done correctly assuming alphabetical vertex and adjacency lists (5) | Breadth-first search was not done correctly assuming alphabetical vertex and adjacency lists (0) |
| Tree provided is a correct breadth-first search tree (5) | Tree provided is not a correct breadth- first search tree (0) |
| **Problem 2** | **10 points** | **0 points** |
|  | |
| Performed a correct depth-first search (3) | Did not perform a correct depth-first search (0) |
| Tree edges correctly labeled (1) | Tree edges not correctly labeled (0) |
| Back edges correctly labeled (1) | Back edges not correctly labeled (0) |
| Forward edges correctly labeled (1) | Forward edges not correctly labeled (0) |
| Cross edges correctly labeled (1) | Cross edges not correctly labeled (0) |
| Provided correct start and finish times (3) | Did not provide correct start and finish times (0) |
| **Problem 3** | **10 points** | **0 points** |
|  | |
| Correctly transformed the graph into a DAG (2) | Did not correctly transform the graph into a DAG (0) |
| Provided correct start and finish times (3) | Did not provide correct start and finish times (0) |
| Provided correct topological order from finish times (3) | Did not provide correct topological order from finish times (0) |
| Provided correct breadth-first topological order (2) | Did not provide correct breadth-first topological order (0) |
| **Problem 4** | **10 points** | **0 points** |
|  | |
| Correctly transposed graph (2) | Did not correctly transpose graph (0) |
| Provided correct start and finish times (4) | Did not provide correct start and finish times (0) |
| Provided correct strongly connected components (2) | Did not provide correct strongly connected components (0) |