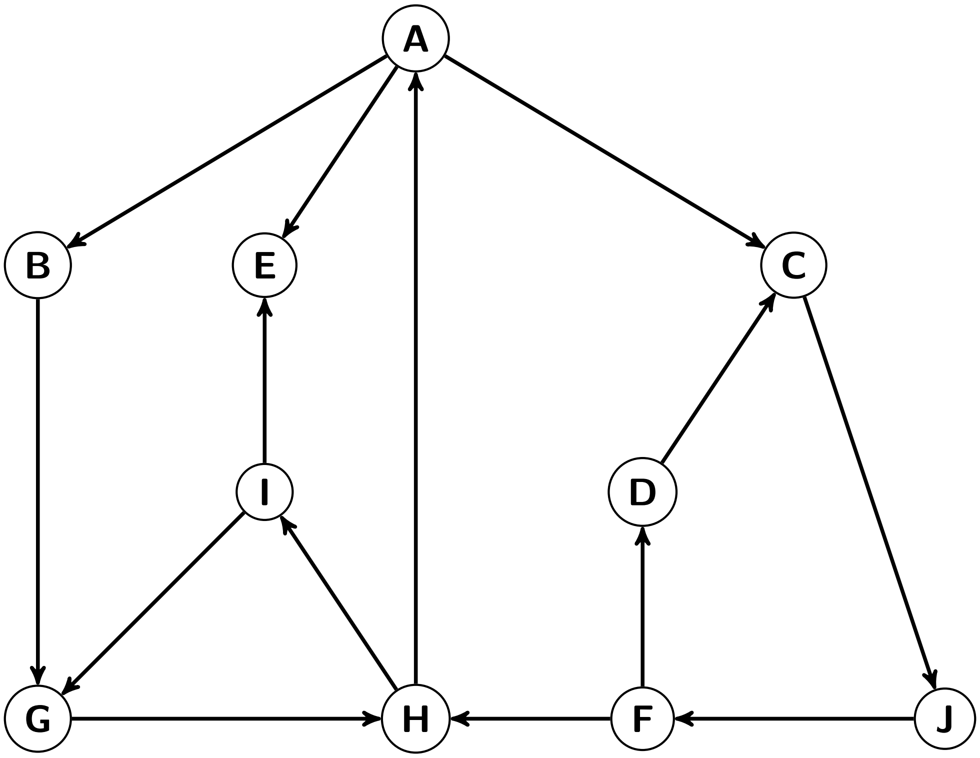
Graph Traversals



1. What are all the strongly connected components? (i.e. groups of vertices such that there exists a path between any two vertices in the group)
2. Perform DFS on the graph above starting from vertex A. Use lexicographical ordering to break vertex ties. As you go, label each node with the start time and the finish time. Highlight the edges in the tree generated from the search.
3. Perform BFS on the graph above starting from vertex A. Use lexicographical ordering to break vertex ties. As you go, label each node with the discovery order. Highlight the edges in the tree generated from the search.

True or False

1. If (u, v) is an edge in an undirected graph and during DFS and $finish(v) < finish(u)$, then u is an ancestor of v in the DFS tree.
2. In a directed graph, if there is a path from u to v and $start(u) < start(v)$ then u is an ancestor of v in the DFS tree.

3. Dijkstra's can be used for successfully finding the shortest path from a source to all other vertices in a graph with negative edges, but not with negative cycles.

4. Bellman-Ford's algorithm can be used for successfully finding the shortest path from a source to all other vertices in a graph with negative edges, but not with negative cycles.

5. To find the shortest path from one vertex to another in an unweighted graph, you should use Dijkstra's algorithm as it is the most efficient solution.

6. Adding a new positive edge to an undirected weighted graph with positive edges cannot lead to the output values of Dijkstra's increasing.

Russian Boxes

You have n boxes. The i-th box has dimensions $w_i \times h_i$. Box i can fit inside box j if and only if $w_i < w_j$ and $h_i < h_j$. A sequence of boxes $b_1, b_2, ..., b_k$ form a chain if box b_i fits inside box b_{i+1} for each $1 \leq i < k$. Design an algorithm which takes as input a list of dimensions $w_i \times h_i$ and returns the length of a longest possible chain of boxes. You must construct a directed graph as part of your solution.

Bipartite Graphs

A Bipartite Graph is a graph whose vertices can be divided into two independent sets, U and V such that every edge (u, v) connects a vertex from U to V or a vertex from V to U. A bipartite graph is possible if the graph coloring is possible using two colors such that vertices in a set are colored with the same color. In lecture, we saw an algorithm using BFS to determine where a graph is bipartite. Design an algorithm using DFS to determine whether or not an undirected graph is bipartite.

Source Vertices

A source vertex in a graph $G = (V, E)$ is a vertex v such that all other vertices in G can be reached by a path from v. Say we have a directed, connected graph that has at least one source vertex.

1. Describe a naive algorithm to find a source vertex.

2. Describe an algorithm that operates in $O(V + E)$ time to find a source vertex.