The solutions to a and b are straightforward, so we focus on c. If \((u, v)\) is a cross edge, it means \(v\) must already be black when \(u\) is visited, otherwise, \((u, v)\) would be tree edge. Therefore, we have \(v.d < v.f < u.d < u.f\). On the other hand, if we have \(v.d < v.f < u.d < u.f\), according to a and b, \((u, v)\) cannot be tree edge, forward edge nor back edge, therefore, it has to be cross edge.

Note that there are only tree edges and back edges in undirected graph. Therefore, if \(v\) was first discovered by exploring edge \((u, v)\), then \((u, v)\) is encountered earlier than \((v, u)\), which means \((u, v)\) is a tree edge; if \(v\) was first discovered by exploring another edge, then \((v, u)\) is encountered earlier than \((u, v)\), which means \((u, v)\) is a back edge.

Consider a graph with 4 nodes a, b, u, v, and edge set \(E = (a, b), (b, u), (u, b), (b, v)\). Nodes are visited in alphabetical order, we can see a path from u to v: u-b-v, and u.d < v.d, but v is not a descendant of u.

Use the same counterexample given in 22.3-8, we can see v.d < u.f.

Set \(k=1\), simply increase \(k\) by 1 once a new DFS is started.

Sort the nodes in topological order, and apply Viterbi-style algorithm. The algorithm visits each node and edge at most once.

A undirected graph is acyclic if and only \(|E| \leq |V| - 1\). Simply count edges, if there are more than \(|V| - 1\) edges, then the graph contains a cycle.

Use adjacent-list, visit each node and edge at most once, so it takes \(O(V+E)\). If the graph contains cycles, at a certain step, there exists no vertex with 0-degree incoming edges.

Incorrect, easy to find counter-example.

Let \((V^T, E^T)\) and \((V, E)\) be the vertex and edge set of the two component graphs. It is obvious that \(V^T = V\), and now we prove \(E^T = E\). For any \(e \in E^T\), we have \(e \notin (G^T)^{SCC}\), and therefore \(e \in G^{SCC}\), so \(E^T \subset E\). It is similar to prove from the other way, so \(E \subset E^T\). Therefore we have \(E^T = E\), and we have proved \(((G^T)^{SCC})^T = G^{SCC}\).

Obtain \(G^{SCC}\) and topologically sort the node of \(G^{SCC}\). Suppose \(v_1, v_2, \ldots, v_k\) are the \(k\) nodes of the graph in topological order. \(G\) is semiconnected if and only if there exists a chain through all nodes in \(G^{SCC}\), i.e., there exist \(k - 1\) edges \((v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k)\). The complexity is \(O(V+E)\).

1. a. If \((u,v)\) is a forward edge, then \(u\) is the closest ancestor of \(v\) in the BFS tree, however, \(u\) has only one closest ancestor in the BFS tree, so \((u,v)\) could either be a tree edge or cross edge. Similarly, we can prove \((u,v)\) cannot be back edge. If \((u,v)\) is a tree edge in BFS tree, then \(u\) is the closest ancestor of \(v\) in the tree and \(v\) has to be discovered by \((u,v)\), so \(v.d = u.d + 1\); if \((u,v)\) is a cross edge in BFS tree, and suppose \(v.d - u.d > 1\), then \(v\) would be discovered by \(u\) earlier through \((u,v)\), and therefore \((u,v)\) becomes a tree edge.

b. 1, 2 and 3 can be proved similarly with those in a, and if \((u,v)\) is a back edge and \(u.d < v.d\), then \((u,v)\) would be tree edge because \(v\) would be discovered through \((u,v)\).