Depth-First Search Applications

Topological Sort

TOPOLOGICAL-SORT(G)

1 call DFS(G) to compute finish times $v.f$ for each vertex v
2 as each vertex is finished, insert it onto the front of a linked list
3 **return** the linked list of vertices

Strongly Connected Component Decomposition

STRONGLY-CONNECTED-COMPONENTS(G)

1 call DFS(G) to compute finish times $u.f$ for each vertex u
2 create G^{T}
3 call DFS(G^{T}), but in the main loop of DFS, consider the vertices
in order of decreasing $u.f$ (as computed in line 1)
4 output the vertices of each tree in the depth-first forest formed in line 3 as a
separate strongly connected component