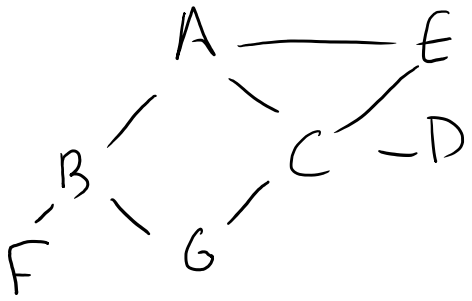
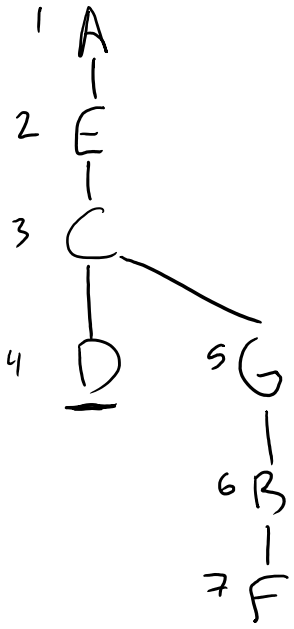
2018-11-01 Graphs 2 - DFS Trees

Thursday, November 1, 2018 2:57 PM

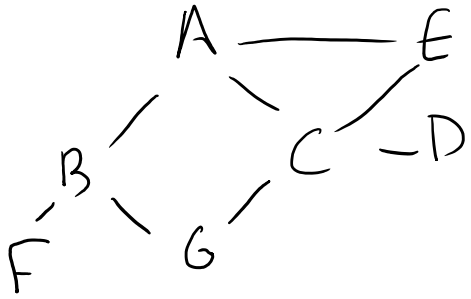
- DFS trees can be constructed by traversing a graph
- DFS trees describe a particular search trajectory through a graph

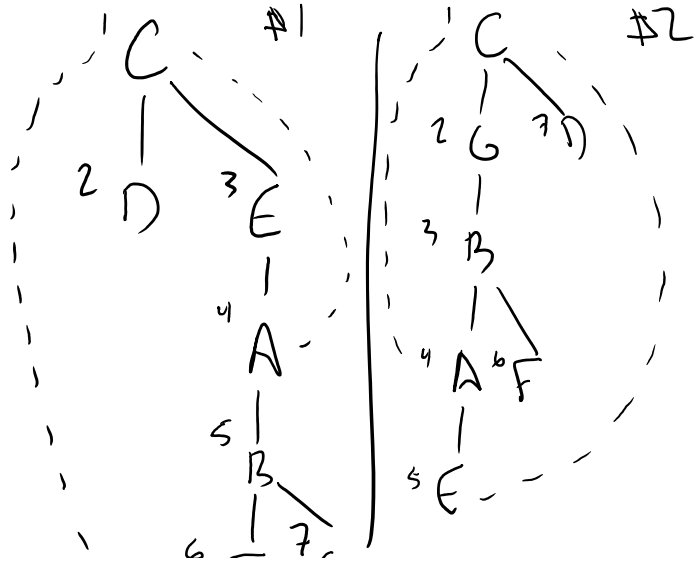


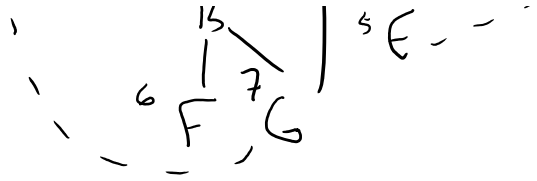
- What does the search pattern look like starting at A (DFS tree)?



Class Exercise: Draw tree starting at C

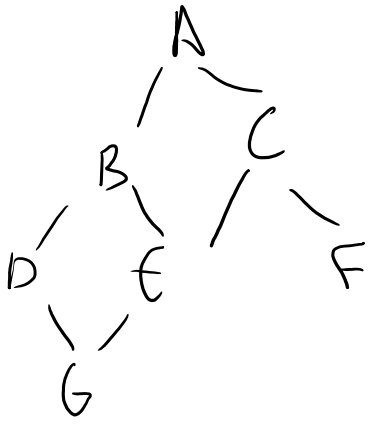


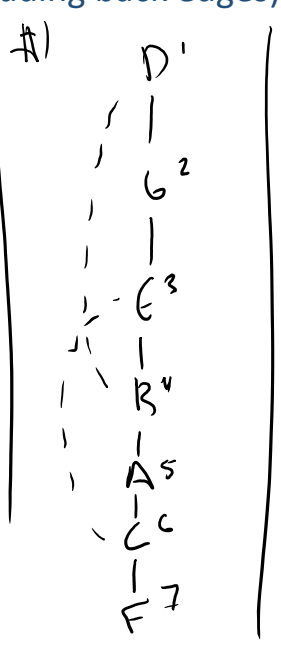


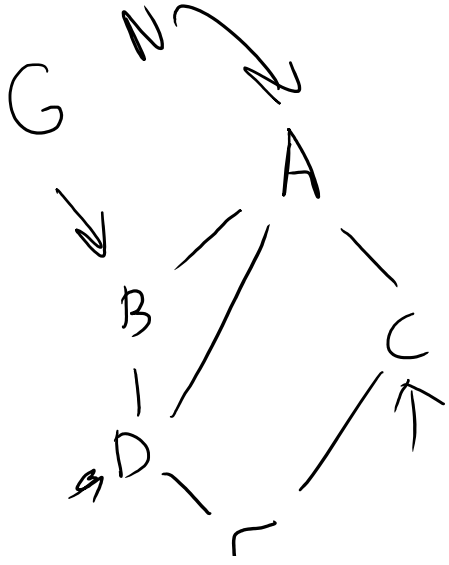


- DFS trees are really "trees" in the strict sense because they also track "back edges"
- A back edge is an edge in the graph that could have been used but wasn't
  - Back edges are represented using dotted lines

Class Exercise: Draw DFS tree (including back edges) for the following graph starting at D:







How would we algorithmically construct a DFS tree?

Function BuildDfsTree
INPUT: Graph g, Node n, VisitedNodes visited, TreeNode tree
RETURNS TreeNode

1. Add n to new TreeNode
2. Add n to visited
3. For each outgoing node "outgoing" in g[n] (graph at node n):
   a. If outgoing is in visited:
      i. Connect n and outgoing using a back edge
   b. ELSE:
      i. Tree->addChild(BuildDfsTree(g, outgoing, visited, null))
4. Return TreeNode created in #1

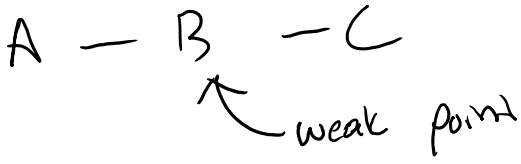
i. Connect n and outgoing using a back edge
   b. ELSE:
      i. Tree->addChild(BuildDfsTree(g, outgoing, visited, null))
4. Return TreeNode created in #1

Visited:	A	B	D	E	C

Post Processing Algorithm
Walk tree. For each node: if child has a back edge to us, remove that back edge.

Finding articulation points in a graph

- An articulation point is a "weak point" in a graph.
  - Defined: a node, which if removed from the graph, would result in a disconnected graph



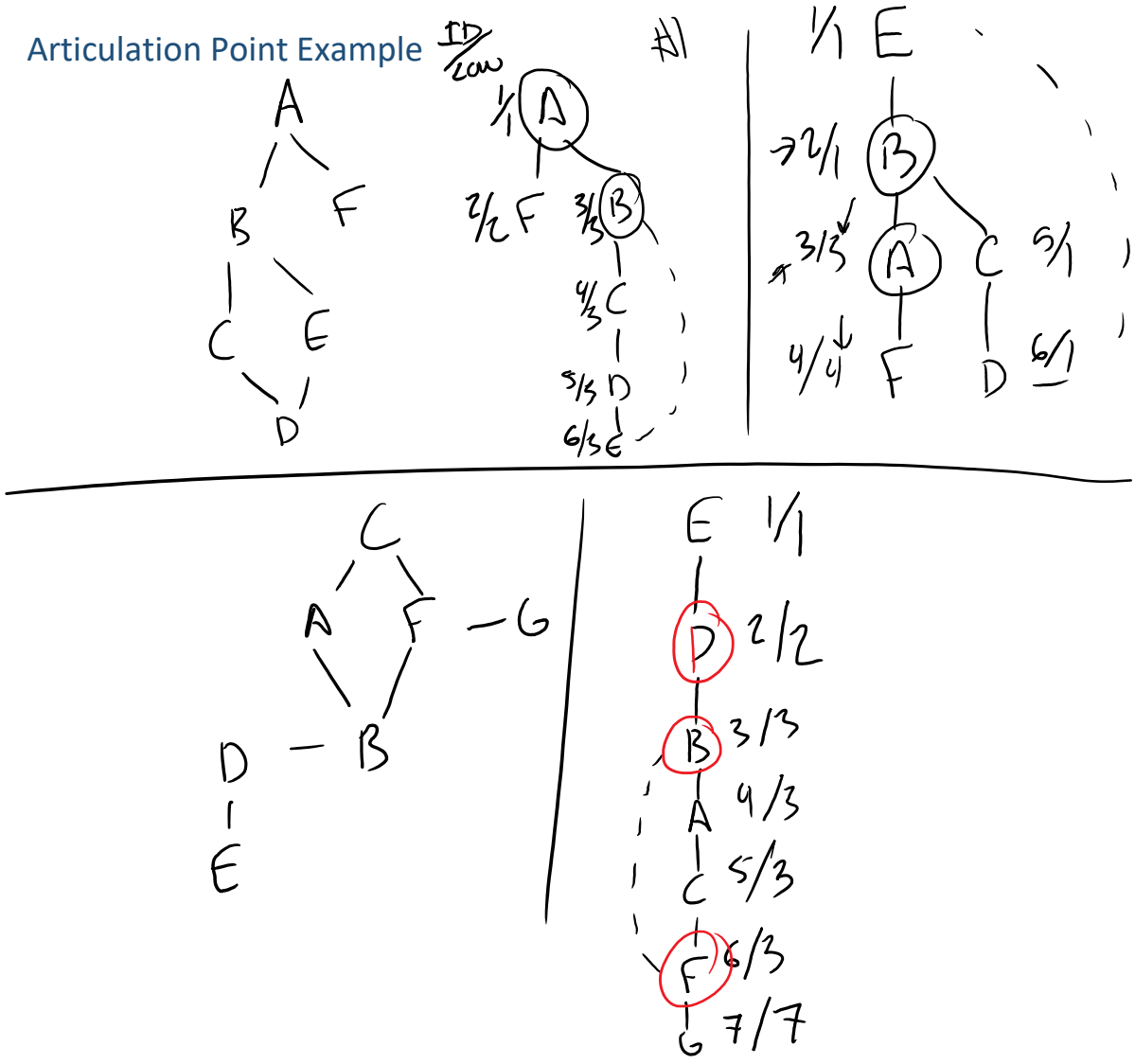
- Formal term for "weak point" is articulation point
- In a social graph, if one person were to go away (e.g. die), who would no longer be friends?
- In network analysis, if this switch, router, or sever were to go down, who would not have internet access?
- Traffic patterns: In case of a natural disaster, will some people get stranded?
- Military conflict: What bridge will allow the enemy to bypass our defenses?

Algorithm for Articulation Points

1. Build DFS tree w/ back edges. Make sure to label each node's order of visit. Call this **ID**.
2. Next, for each node in the DFS tree, find the lowest possible ID reachable by taking zero or more forward edges (solid line) **and up to one** back edge (dotted line). Let be called the **LOW** value.
3. A node is an articulation point:

a. IF it is the root and the root has more than one child
b. Or, when not the root and node's child LOW value >= node's ID value

Articulation Point Example



Analyzing Efficiency

Algorithm for Articulation Points

N 1. Build DFS tree w/ back edges. Make sure to label each node's order of visit. Call this **ID**.

N^2 2. Next, for each node in the DFS tree, find the lowest possible ID reachable by taking zero or more forward edges (solid line) **and up to one** back edge (dotted line). Let be called the **LOW** value.

N 3. A node is an articulation point:
   a. IF it is the root and the root has more than one child
   b. Or, when not the root and node's child LOW value >= node's ID value

$O(N^2)$ → too slow

- How might we increase the efficiency of step #2?
  - By working backwards, we only touch each node once. Thus, #2 becomes N