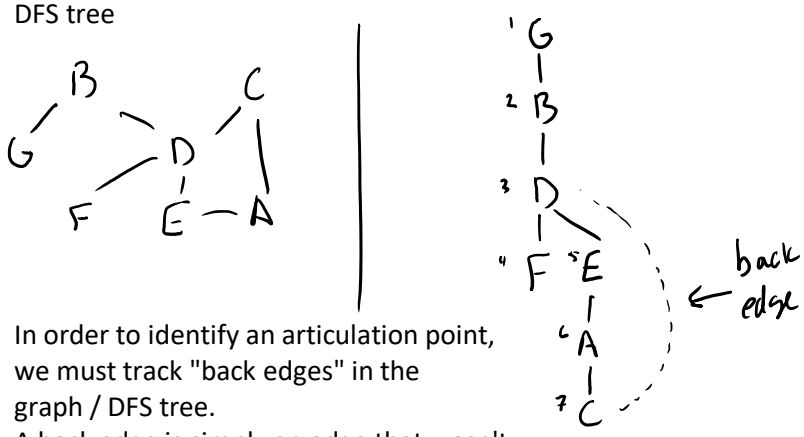
2019-04-25 Depth First Search / Articulation Points

Thursday, April 25, 2019 9:00 AM

Recall from last lecture

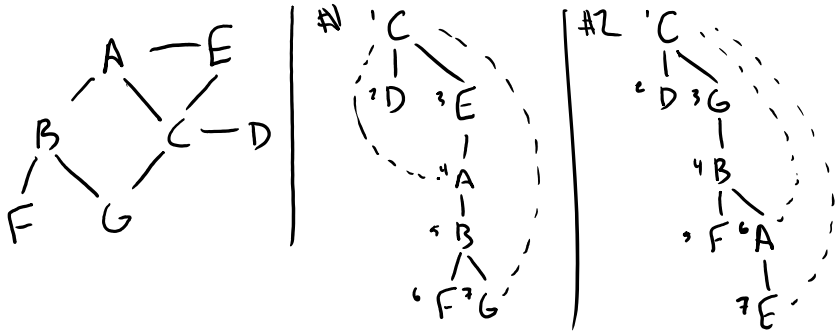
- Articulation points are "weak points" in the graph such that if removed, the graph would become disconnected
- Articulation points are algorithmically identified using a DFS tree



- In order to identify an articulation point, we must track "back edges" in the graph / DFS tree.
- A back edge is simply an edge that wasn't taken in our DFS
- Graphically, back edges are represented in the DFS tree with dotted lines
- Programmatically, they are housed in a separate data structure

---

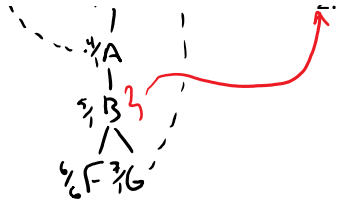
Class Exercise: Draw DFS tree w/ back edges starting at C



- To find an articulation point using a DFS tree w/ back edges we must add another piece of information to our tree
  - LOW value: The smallest visit order / ID of the node that is reachable by taking zero or more forward (solid) edges AND up to one back edge.

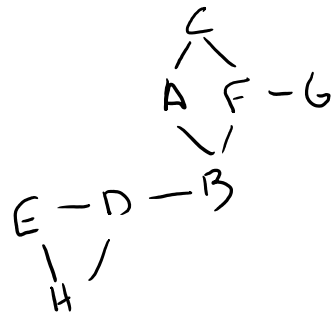
A vertex is an articulation point

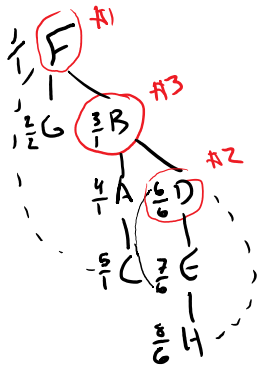
1. IF it is the root and it has more than one child
2. OR, not root and it has a direct child whose LOW value is greater than or equal to its ID value

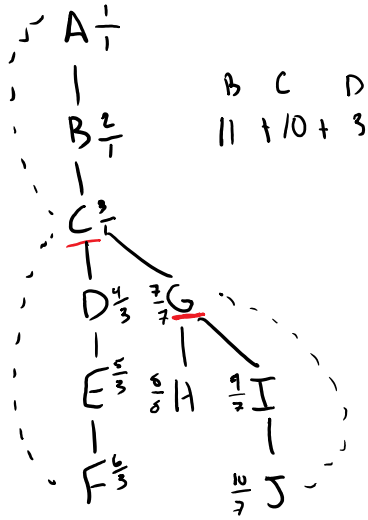


value is greater than or equal to its ID value

Articulation Point Class Exercise







B C D E F G H I J
11 + 10 + 3 + 2 + 1 + 4 + 1 + 2 + 1

Articulation Point Algorithm

1. Given a DFS tree w/ back edges but without low values
2. Set low value of root to 1
3. Assign low values by:
   a. Doing at DFS starting at each node, looking for the minimal ID possible by examining all of each node's back edges, all children and their back edges as well
4. Now, traverse the tree. A node is an articulation point
   a. IF it is the root with more than one child
   OR if it has a direct child whose LOW is greater than or equal to its ID value

Alternate algorithm for #3

1. Working in reverse sorted order of ID values
   a. Set LOW = MIN(SELF_ID, CHILDREN_LOW, BACKEDGE_ID)

Building a DFS tree from a graph