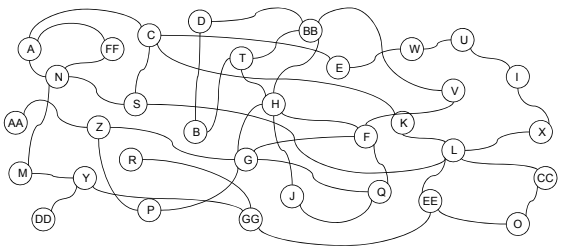
COMP261 Lecture 8

Articulation Points 1 of 2 (Ideas)

Victoria
UNIVERSITY OF WELLINGTON
Te Whare Wānanga o te Ūpoko o te Ika a Māui
CAPITAL CITY UNIVERSITY

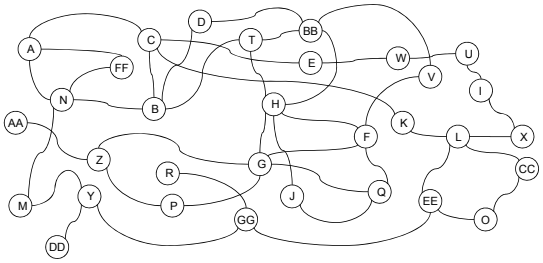
Connectedness

- Is this graph connected or not?



Articulation points

- This graph is connected, but is it "fragile"?
  Would deleting one node disconnect it?



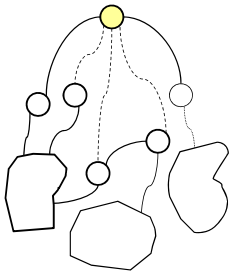
- *Articulation point: node whose removal would disconnect part of the graph.*

Articulation Points: a bad algorithm

- Take each node out in turn, and test for connectedness

```
articulationPoints ← empty set
for each node in graph
    node.visited ← true;
    for all other nodes nd : nd.visited ← false
    recDFS(first neighbour of node)
    for each remaining neighbour of node
        if not neighbour.visited then
            add node to articulationPoints

recDFS (node ):
    if not node.visited then
        node.visited ← true,
        for each neighbour of node
            if not neighbour.visited
                recDFS(neighbour )
```



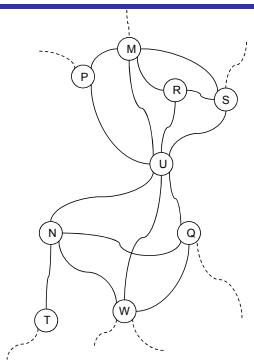
Why is it bad?

- Cost of DFS: O(e) = $O(n^2)$ for very dense graphs
- Cost of Alg: O(ne) = $O(n^3)$ for very dense graphs
- Why do we have to traverse the whole graph n times, once for each node?
- Why not do a single traversal, identifying all articulation points as we go?

Articulation Points.

- What are we looking for?

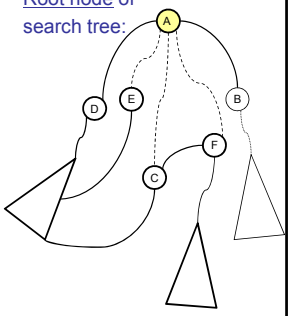
Nodes in a graph that separate the graph into two groups, so that all paths from nodes in one group to nodes in the other group go through the node.



Articulation points: DFS

- Use depth first search, keeping track of the depth of each node in the search tree
- At root:
  if there is more than one edge to an unvisited node,
  then root is an articulation point.
- At other node:

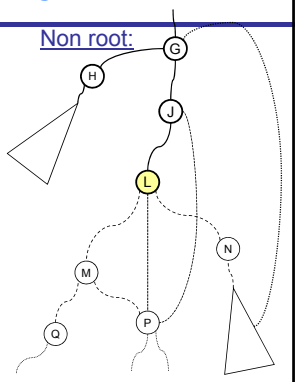
Root node of search tree:



Articulation points: DFS

- Use depth first search, keeping track of the depth of each node in the search tree
- At root:
  if there is more than one edge to an unvisited node,
  then root is an articulation point.
- At other node:
  If there is a subtree that has no edge up to an ancestor node
  then node is an articulation point.

Non root:



Articulation Points

- Key ideas of algorithm:
  - record depth of nodes as you search (or record order in which visited)
  - From each recursive search of a subtree, return the minimum depth that the subtree node that the subtree can "reach back" to.
    (ie, the minimum depth of any previously visited neighbour
  - Compare the "reach back" of each subtree to depth of this node
  - Can use depth to record whether visited