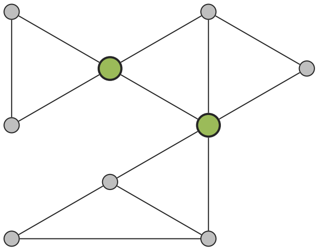
Finding Articulation Points and Bridges

Articulation Points

Articulation Point

Articulation Point

A vertex v is an *articulation point* (also called *cut vertex*) if removing v increases the number of connected components.



A graph with two articulation points.

Articulation Points

Given

- An undirected, connected graph $G = (V, E)$
- A DFS-tree T with the root r

Lemma

A DFS on an undirected graph does not produce any cross edges.

Conclusion

- If a descendant u of a vertex v is adjacent to a vertex w, then w is a descendant or ancestor of v.

Removing a Vertex v

Assume, we remove a vertex $v \neq r$ from the graph.

Case 1. v is an articulation point.

- There is a descendant u of v which is no longer reachable from r.
- Thus, there is no edge from the tree containing u to the tree containing r.

Case 2. v is not an articulation point.

- All descendants of v are still reachable from r.
- Thus, for each descendant u, there is an edge connecting the tree containing u with the tree containing r.

Removing a Vertex v

Problem

- v might have multiple subtrees, some adjacent to ancestors of v, and some not adjacent.

Observation

- A subtree is not split further (we only remove v).

Theorem

A vertex v is articulation point if and only if v has a child u such that neither u nor any of u's descendants are adjacent to an ancestor of v.

Question

- How do we determine this efficiently for *all* vertices?

Detecting Descendant-Ancestor Adjacency

Lowpoint

The *lowpoint* $\mathrm{low}(v)$ of a vertex v is the lowest depth of a vertex which is adjacent to v or a descendant of v. Formally,

$$\mathrm{low}(v) := \min\{\, \mathrm{depth}(w) \mid w \in N[u], u \text{ is decendent of } v \text{ (or equal } v) \,\}$$

Computing $\mathrm{low}(v)$ for all v

- Post-order traversal on DFS-tree T.

Theorem

A vertex v is an articulation point if and only if v has a child u with $\mathrm{low}(u) \geq \mathrm{depth}(v)$.

Algorithm

Procedure *FindArtPoints*(v, d)

  Set $\mathrm{vis}(v) := \mathrm{Ture}$, $\mathrm{depth}(v) := d$, and $\mathrm{low}(v) := d$.

  For Each $u \in N(v)$

    If $\mathrm{vis}(v) = \mathrm{False}$ **Then**

      FindArtPoints(u, $d + 1$)

    $\mathrm{low}(v) := \min\{\mathrm{low}(v), \mathrm{low}(u)\}$

    If $\mathrm{low}(u) \geq \mathrm{depth}(v)$ **Then**

      v is articulation point.

Special Case: Root of DFS-Tree

For the root r

- $\mathrm{low}(u) \geq \mathrm{depth}(r)$ for all $u \neq r$

Theorem

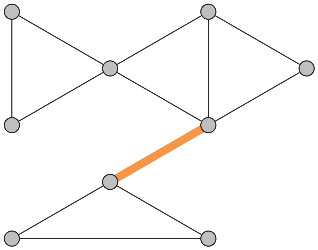
The root r is an articulation point if and only if it has at least two children in the DFS-tree.

Bridges

Bridge

Bridge

An edge is called *bridge* if removing it from the graph (while keeping the vertices) increases the number of connected components.



A graph with a bridge.

Finding Bridges

Lemma

An edge uv is a bridge if and only if $\{u, v\}$ is a block.

- Use articulation points algorithm to find blocks of size two.

Observations

- A bridge is part of every spanning tree.
- If u is parent of v in a rooted spanning tree, then uv is a bridge if and only if every vertex reachable from v not using u is a descendant of v.

Theorem

If u is parent of v in a rooted spanning tree, then uv is a bridge if and only if $\mathrm{low}(v) = \mathrm{depth}(u)$ and for all children w of v, $\mathrm{low}(w) = \mathrm{depth}(v)$.