

Cycles in Directed Graphs

- A cycle in a directed graph is a path that starts and ends with the same vertex

$$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_k \rightarrow v_0$$

$$A \rightarrow C \rightarrow E \rightarrow A$$

A directed graph has a directed cycle iff its dfs output tree has a back edge

Proof: $\rightarrow$

Suppose G has a cycle:

$$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_k \rightarrow v_0$$

A directed graph has a directed cycle iff its dfs output tree has a back edge

Proof: $\rightarrow$

Suppose G has a cycle:

$$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_k \rightarrow v_0$$

Suppose v_0 is the first vertex to be discovered.

(What does that mean about v_0 ?)

A directed graph has a directed cycle iff its dfs output tree has a back edge

Proof: $\rightarrow$

Suppose G has a cycle:

$$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_k \rightarrow v_0$$

Suppose v_0 is the first vertex to be discovered. (the vertex with the lowest pre-number.)

All other v_j are reachable from it and therefore, they are all descendants in the DFS tree.

A directed graph has a directed cycle iff its dfs output tree has a back edge

Proof: $\rightarrow$

Suppose G has a cycle:

$$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_k \rightarrow v_0$$

Suppose v_0 is the first vertex to be discovered. (the vertex with the lowest pre-number.)

All other v_j are reachable from it and therefore, they are all descendants in the dfs tree.



Therefore the edge (v_k, v_0) is a back edge.

A directed graph has a directed cycle iff its
dfs output tree has a back edge

Proof: ←

Suppose (b, a) is a back edge.

A directed graph has a directed cycle iff its dfs output tree has a back edge

Proof: $\leftarrow$

Suppose (b, a) is a back edge.

Then by definition a is an ancestor of b so there is a path from a to b in the DFS output tree.

A directed graph has a directed cycle iff its dfs output tree has a back edge

Proof: ←

Suppose (b, a) is a back edge.

Then by definition a is an ancestor of b so there is a path from a to b in the DFS output tree.

Along with the back edge, this path completes a cycle. 

Directed Acyclic Graphs (DAG)

- A directed graph without a cycle is called acyclic. (DAG)

Corollary:

A directed graph G is a DAG if and only if it's DFS output tree does not have any back edges.

How to spot a DAG?

Step 1: perform dfs on the graph

Step 2: loop through each edge to see if it is a back edge.

i.e.:

for each edge (u, v) ,

if $\text{pre}(v) < \text{pre}(u) < \text{post}(u) < \text{post}(v)$:

return “not DAG”

return “DAG”

Linearization aka Topological Sort

- Is it possible to order the vertices such that all edges go in only one direction?
- For what types of DAGs is this possible?
- How do we find such an ordering?