**DFS: Undirected Graph**

**Theorem:** In an undirected graph G, a DFS produces only tree and back edges.

**Theorem:** An undirected graph is *acyclic* iff a DFS yields no back edges.
- If acyclic, there are no back edges (back edge implies a cycle)
- If no back edges, then graph is acyclic because
  o DFS will produce only tree
  o Trees are by definition acyclic

**DFS vs. BFS Uses**
- DFS can be used to detect *cycles*
- BFS will yield *shortest path*

**Directed Acyclic Graphs (DAG)** - A directed graph with no directed cycles

```
Example of:  Tree   DAG   Cyclic Graph
```

A **Partial Order** R on a set S is a binary relation such that:
- For all a in S, a R a is false
- For all a, b, c in S,

**Theorem:** A directed graph is acyclic iff a DFS yields no back edges
- If G is acyclic:
  o No back edges (trivial)
- If G has a cycle, there must exist a back edge

**DFS, DAG, and Strongly Connected Components**

**Topological Sort** - Linear ordering of all vertices in graph G such that vertex u comes before vertex v if edge (u, v) in G
- Real World Application: Scheduling a dependent graph, finding a feasible course plan for university studies
A graph that can be topologically sorted as A,B,C or A,C,B

Topological-Sort()
{
1. Call DFS to compute finish time f[v] for each vertex
2. As each vertex is finished, insert it onto the front of a linked list
3. Return the linked list of vertices
}

Runs in $O(V+E)$ time

Lemma: $(u,v)$ in G implies $f[u] > f[v]$
- When $(u,v)$ is explored, $u$ is gray, consider
  - $v$ is gray. $(u,v)$ is back edge. Can’t happen if G is a DAG
  - $v$ is white. $v$ becomes descendent of $u$. $f[v] < f[u]$
  - $v$ is black. $v$ already finished. $f[v] < f[u]$
- In other words, since there is no path from $u$ back to $v$ (DAG, by definition), $u$ will always finish later.

Strongly Connected Directed Graphs - Cyclic directed graph in which every vertex can be reached from every other vertex
G is strongly connected if, for every $u$ and $v$ in V, there is some path from $u$ to $v$ and from $v$ to $u$

Strongly connected components - a maximal subset of nodes, along with their associated edges, that is strongly connected. Nodes share a strongly connected component if they are inter-reachable. Can be viewed as “communities”

Example of two sets of three Strongly Connected Components in a directed graph