Directed Graphs

Directed graph: $G = (V, E)$
- Edge $(u, v)$ goes from node $u$ to node $v$.

Example: Web graph - hyperlink points from one web page to another.
- Directedness of graph is crucial.
- Modern web search engines exploit hyperlink structure to rank web pages by importance.

Adjacency List
- Store the edges in the list in the direction that the edge points.
Finding Paths

- Use BFS or DFS
- No guarantee that there is a reverse path

TO

Strong Connectivity

Node u and v are mutually reachable if there is a path from u to v and also a path from v to u.

A graph is strongly connected if every pair of nodes is mutually reachable.

Directed Acyclic Graphs

A directed acyclic graph (DAG) is a directed graph that contains no directed cycles.

is a DAG

is NOT a DAG
Directed Acyclic Graphs

Uses for DAGs:
- Derives-from: You must buy food in order to make dinner.
- Timing: You should finish your homework before it is due.
- Is-bigger-than, is-faster-than, ...
- Any relationship that imposes a partial order.

Topological Order

A topological order of a directed graph $G = (V, E)$ is an ordering of its nodes as $v_1, v_2, \ldots, v_n$ so that for every edge $(v_i, v_j)$ we have $i < j$.

Computing the Topological Order

**Lemma**: If $G$ is a DAG, then $G$ has a node with no incoming edges.

**Proof?**
Algorithm

topological-order(v) {
    find a node v with no incoming edges
    delete v from G
    return v followed by topological-order (G - {v})
}
Algorithm
topological-order(v) {
    find a node v with no incoming edges
    delete v from G
    return v followed by topological-order(G - {v})
}

Order: 1, 5, 2, 4, 3