SSSP in DAGs (directed acyclic graphs)

- **DFS (Depth First Search)**

  ```
  dfs (vertex v) {
      v.visited = TRUE;
      for each w adjacent to v do
          if (!w.visited) then dfs(w);
  }
  ``

- If $G = (V, E)$ not (strongly) connected => may have DFS forest.

- **Topological Sort**: ordering of vertices in a DAG based on precedence: if path $v_i \rightarrow v_j$, then $v_i$ before $v_j$ in ordering.

- Not unique:

  ![Diagram](image)

  $v \in V$, indegree $(v)$: # edges $(u, v) \in E$

- Can use DFS to obtain TS order.

- Given directed graph $G$, is it a DAG?

  - Can use TS to answer it
    (cannot find vertex with indegree 0)
TS algorithm:
- output (number) \( v \in V \) with indegree 0
- (remove \( v \) and its outgoing edges)
- repeat until no vertex left

How to find \( v \in V \) with indegree \( (v) = 0 \) ?
- use queue:
  - each time incoming edge at \( u \in V \) is removed, decrease indegree \( (u) \)
  - if indegree \( (u) = 0 \), place \( u \) in queue
  - \( O(|V|+|E|) \) time

  * hint: find \( v \in V \) with indegree \( (v) = 0 \) in \( O(|V|) \) time.

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Indegree before dequeue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
</tbody>
</table>

Enqueue

Dequeu
Lemma 1: A DAG has a sink and a source.

Proof:

Let \( P \) be the longest path in \( G \), \( u \) mins \( v \).

Claim: \( u \) is a source.

If not, \( \exists (w, u) \). If \( w \notin P \Rightarrow P = \{w, u\}\) then
is longer than \( P \). If \( w \in P \Rightarrow \text{cycle}:

\[
\begin{array}{c}
\text{w} \\
\text{u} \\
\text{w} \\
\text{v}
\end{array}
\]

A similar argument shows \( v \) is a sink.

Theorem 1: A directed \( G \) has a TS \( \iff \) \( G \) is a DAG.

Proof:
Assume \( G \) has TS and a cycle \( C \). Let \( v_i \) be the vertex with smallest index assigned by \( T \).

There is then an edge \( (v_{j'}, v_i) \), \( v_j \in C \), with \( j' > i \) = contradiction with TS assignment.

Now assume \( G \) acyclic \( \Rightarrow \) use induction.

Define \( P(n) \): a directed acyclic graph with \( n \) vertices has a TS.

\[ \Rightarrow P(1) \text{ true} \]

Assume \( P(m) \) is true. Let \( G \) have \( m+1 \) vertices \( \Rightarrow \) source \( v_0 \). \( G \setminus \{v_0\} \) has TS
\( v_1, v_2, \ldots, v_m \Rightarrow G \) has TS \( v_0, v_1, v_2, \ldots, v_m \).
Use vertex selection rule: select in TS order.

- **SP** well defined, even if negative weight edges (negative weight cycles cannot exist)

\[
\text{DAG-SP} (G, w, s)
\]

- \text{topologically sort vertices of } G

\[
\text{Init-Single-Source} (G, s)
\]

- \text{for each } u \in V, \text{ in TS order do}

\[
\text{[for each } v \in \text{Adj}[u] \text{ do}
\]

- \text{Relax} (u, v, w)

- \text{O}(|V| + |E|) \text{ time.}

- **Critical path in a DAG**: longest path through the DAG

  - Can find one by:

    1. negating edge weights and running \text{DAG-SP}, or
    2. run \text{DAG-SP} with modifications:

      - replace \text{"\in\in\in"} by \text{"-\in\in\in"} in \text{Init-Single-Source}
      - replace \text{"\rightarrow\rightarrow\rightarrow"} by \text{"\leq\leq\leq"} in \text{RELAX}

  \text{Note: Longest simple path in unweighted graph } \rightarrow \text{NP-complete}
Example DAG $\rightarrow$ SP:

Rule: $d(u) = \min \{ d(w), d(v) + w(v, u) \}$

<table>
<thead>
<tr>
<th>$v$</th>
<th>$d(v)$</th>
<th>$P(v)$</th>
<th>Queue: 1, 2, 5, 4, 3, 7, 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>nil</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$\infty$</td>
<td>3</td>
<td>$\infty$ 4</td>
</tr>
<tr>
<td>4</td>
<td>$\infty$</td>
<td>1</td>
<td>$\infty$ 1</td>
</tr>
<tr>
<td>5</td>
<td>$\infty$</td>
<td>12</td>
<td>$\infty$ 2</td>
</tr>
<tr>
<td>6</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$ 4</td>
</tr>
<tr>
<td>7</td>
<td>$\infty$</td>
<td>5</td>
<td>$\infty$ 5</td>
</tr>
</tbody>
</table>