DAG: directed acyclic graph

no (directed) cycles

not a DAG
Topological sort (on a DAG)

"Sort" all vertices (place in line) s.t. all directed edges are →

V₄ → V₅ → V₁ → V₃ → V₂

implied
Topological Sort (on a DAG)

"Sort" all vertices (place in line) s.t. all directed edges are implied.
DFS tree from $V_1$: $V_1 \xrightarrow{} V_2 \xrightarrow{} V_3$

directly gives us some info, but...

"Sort" all vertices (place in line) s.t. all directed edges are

Topological Sort (on a DAG)
DFS tree from $v_1$: $v_1 \rightarrow v_2 \rightarrow v_3$

- directly gives us some info, but...

  - notice, we visited $v_2$ before $v_3$

  - Otherwise
DFS tree from $v_1$:

$V_1 \rightarrow V_2 \rightarrow V_3$

Directly gives us some info, but...

Notice, we visited $v_2$ before $v_3$

Otherwise

We need this order:

$V_1 \rightarrow V_3 \rightarrow V_2$
DFS tree from $v_1$:

- Directly gives us some info, but...
- Notice, we visited $v_2$ before $v_3$.

Rule: sort/output by finish time. $v_2$ finishes first. Then $v_3$. Then $v_1$.
OR

Continue DFS: search \( v_4, v_5 \)
OR

\[ \begin{align*}
V_1 & \rightarrow V_3 \rightarrow V_2 \\
V_1 & \rightarrow V_4, V_5 \\
\end{align*} \]

Continue DFS: search \( V_4, V_5 \)

\[ \begin{align*}
V_4 & \rightarrow V_5 \\
\end{align*} \]

group 1

group 2

group 1 finished before group 2
group 1 finished before group 2, so:

$V_4 \rightarrow V_5$

$V_1 \leftrightarrow V_3 \rightarrow V_2$
group 1 finished before group 2, so:

- Continue DFS: search $v_4$, $v_5$
  - $v_4 \rightarrow v_5$
  - $v_1 \leftarrow v_3 \rightarrow v_2$

  can add these; $v_4$ found $v_3$, $v_1$, but they were marked
group 1 finished before group 2, so:

We could have had other groups or DFS trees, but each would give a valid topological sort.
Intuition

DFS trees
Intuition

impossible

DFS trees
Intuition

If $x \rightarrow y$ is implied in a DFS tree then $y$ was explored after $x$. So $y$ finished first.
Intuition

If $x \rightarrow y$ is implied in a DFS tree then $y$ was explored after $x$. So $y$ finished first.

DFS trees

we output correctly
If $x \rightarrow y$ is implied in a DFS tree then $y$ was explored after $x$. So $y$ finished first.

If $x \rightarrow z$ not in tree & not implied, then ?
Intuition

If $x \rightarrow y$ is implied in a DFS tree then $y$ was explored after $x$.

So $y$ finished first

If $x \rightarrow z$ not in tree & not implied, then $x$ was explored after $z$.

(subtree) (subtree)

(otherwise we would have $x \rightarrow z$)

So $?$
Intuition

If $x \rightarrow y$ is implied in a DFS tree then $y$ was explored after $x$.

So $y$ finished first

If $x \rightarrow z$ not in tree & not implied, then $x$ was explored after $z$.

(subtree)

(otherwise we would have $x \rightarrow z$)

(subtree)

So $z$ finished first

we output correctly
Intuition

If $x \rightarrow y$ is implied in a DFS tree then $y$ was explored after $x$.

So $y$ finished first

If $x \rightarrow z$ not in tree & not implied, then $x$ was explored after $z$.

(subtree)

(otherwise we would have $x \rightarrow z$)

So $z$ finished first

In both cases, we output correctly
Run DFS in any order.
Run DFS in any order.

When a vertex $v$ has been processed entirely, add it to a list.
Summary

Run DFS in any order.

When a vertex $v$ has been processed entirely, add it to a list.

Every vertex reachable from $v$ will be done before $v$, so it will already be in the list.
Run DFS in any order.

When a vertex \( v \) has been processed entirely, add it to a list.

Every vertex reachable from \( v \) will be done before \( v \), so it will already be in the list.

Similarly, \( v \) will be ahead of any vertex that can reach it.