Breakfast Robot
There are some tasks that need to be done to eat breakfast:

get glass, pour juice, get bowl, pour cereal,

pour milk, get spoon, eat.

Some of the events must take precedence over others. For example, “get bowl” should precede “pour milk”.

The ordering of some other events is irrelevant, e.g., “get g/b/s”.

Example
Example

get glass
pour juice
get bowl
pour cereal
pour milk
get spoon
eat breakfast

Goal: Embed the partial order of events into a total order
Let $S$ be a set and $\leq$ a relation on $S$. Then $\leq$ is called a partial order if and only if for all $a$, $b$, $c$ in $S$, we have

- $a \leq a$ (reflexivity)
- if $a \leq b$ and $b \leq a$, then $a = b$ (antisymmetry)
- if $a \leq b$ and $b \leq c$, then $a \leq c$ (transitivity)

Any partial order can be embedded into a total order.
Let $\leq$ be a partial order on a set $S$. The cover relation $\ll$ of this partial order is defined as

$a \ll b$ if and only if $a \leq b$ and there doesn’t exist $x$ s.t. $a < x < b$.

Two elements are related under the cover relation iff their are immediate neighbors in the partial order.
Representation

Let \((S, \leq)\) be a partial order, and \(\prec\) its cover relation.

Let \(R\) be any relation on \(S\) such that

- \(\prec\) is contained in \(R\),
- \(R\) is contained in \(\leq\)

Then the reflexive and transitive closure of \(R\) is \(\leq\).

The relation \(R\) can be represented by a directed acyclic graph.
Topological Sorting

Let $G=(S,E)$ be a directed acyclic graph.

Then $G$ represents a partial order.

Goal: Find a total order $\leq$ on $S$ such that if $(u,v)$ in $E$, then $u \leq v$.

This can be solved, since any partial order can be embedded into a total order.
Example

1/6 get glass
2/5 pour juice
7/12 get bowl
8/11 pour cereal
9/10 pour milk
3/4 eat breakfast

get spoon ≤ get bowl ≤ pour cereal ≤ pour milk ≤ get glass ≤ pour juice ≤ eat breakfast
Topological Sorting Algorithm

**Input:** Directed acyclic graph \( G = (V,E) \)

1. Call DFS on \( G \) to compute \( \text{finish}[v] \) for all nodes \( v \)

2. After a node's recursive call finishes, insert it at the **front** of a linked list

3. return the linked list (so, events are ordered by decreasing finishing time).

**Running Time:** \( O(V+E) \)
Correctness

Let $e = (u,v)$ be an edge of the directed acyclic graph $G=(V,E)$.

- If $e$ is a forward or tree edge, then $\text{finish}[v] < \text{finish}[u]$.
- If $e$ is a cross edge, then $\text{finish}[v] < \text{disc}[u] < \text{finish}[u]$.
- The edge $e$ cannot be a back edge, since $G$ is acyclic.

Therefore, $\text{finish}[u] > \text{finish}[v]$ in all cases. Thus, the total order produced by DFS respects the partial order implied by $G$. 
Many thanks to Jennifer Welch for providing the breakfast example and breakfast priority graph.