Topological order - algorithms and examples

- Topological ordering of a directed graph is an ordering such that for every directed edge (u,v) the vertex u always comes before the vertex v.
- For example the vertices of the graph may represent tasks to be performed, and the edges may represent constraints that one task must be performed before another; in this application, a topological ordering is just a valid sequence for the tasks.
- A topological ordering is possible if and only if the graph is a directed acyclic graph (DAG).
- A directed acyclic graph (DAG) is a directed graph with no cycles, i.e. traversing the graph from one vertex to another in the direction of the arc connecting them will never form a closed loop.

Algorithms

Kahn's algorithm

1. Find a list of "start nodes" that have no incoming edges
2. Then for each node in "start nodes" append it to the final list, remove it from "start nodes" and remove it from the graph, along with all of the edges connected to that node.
3. Repeat the steps 1,2 until the "start nodes" list is empty
4. If there are still some vertices that are not in the final list then our graph is not a DAG

DFS

1. Create a list of visited nodes, and a list for the topological order

2. Run the DFS for each node in the graph with one modification: after visiting all neighbours of a node append that node to the topological order list
3. Reverse the topological order list.