Topological Sorting

Thursday, October 26, 2023 4:00 PM

- Some things hv to occur before everything else.
- eg. Class prerequisites, program build dependency

- Can find in O(V+E) time.

Note: Topological orders are not unique.

- Not every graph has a topological order (graphs with cycles)

Only DAGs

- Every tree has a topological order.
  - ↳ cherry picking from the leaf.
    - Once all leaves are gone, node becomes available.

Add the node to the order while backtrack.

- Can be done with both DFS & BFS
- DFS inserts a node in the ordering when its outdegree becomes 0.
- BFS inserts a node in the ordering when its indegree becomes 0.

Kahn's algo

↳ repeatedly remove nodes w/o dependencies.

we use a queue & push elements when its indegree becomes 0.

Shortest/longest path in a graph

- Linear time algo to find shortest path between two nodes. O(V+E). (also works with -ve edges)
- Next fastest algo → Djikstra's algorithm.

find topological order & process sequentially.

1. Works because we only add node to queue when all indegree becomes 0.
   - ↳ Makes sure that all paths to node are explored & min is selected.

Main reason this works is because, by the time you get to a node, all the nodes which precede it have been processed.

Why use Djikstra?

- Topsort algo fails when graphs have cycles.
- Graph must be a DAG

No order is possible.