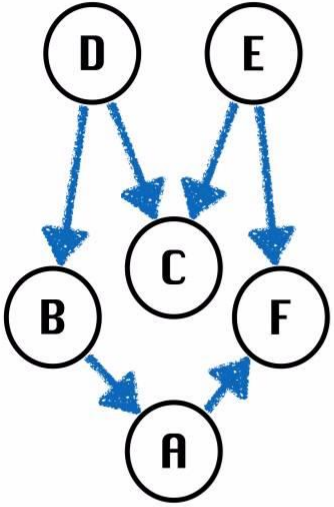
Topological Sort

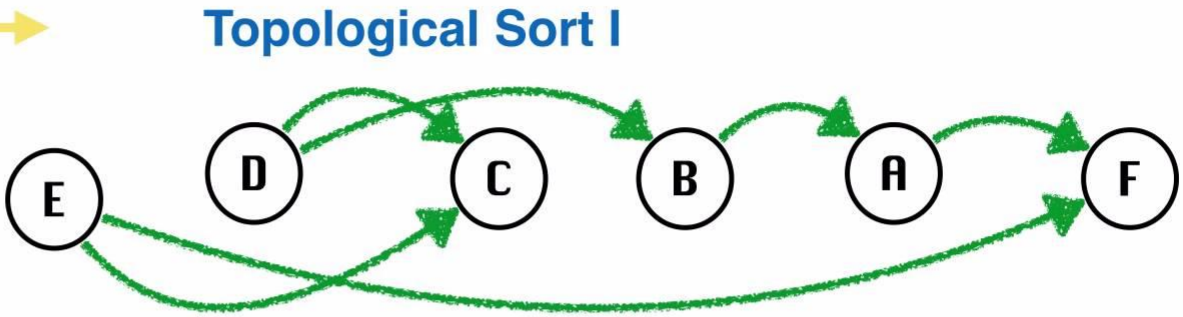
2017-02-08

What is it?

- Ordering of vertices of a directed graph so that for every edge uv from vertex u to v, u is before v
- Every DAG has at least one topological ordering
- O(n) algorithms exist

Topological Sort





Directed Graphs

- AddDirectedEdge(u, v : Vertex)
- AdjacentTo(v : Vertex) : []Vertex
- RemoveEdge(u, v : Vertex)
- EdgeCount() : int
- Keep track of degrees of Vertices
- Adjacency list is good choice.

Strategy for Contests

- Building graph should be hard part
- Once graph is built run through easy to implement algorithm

Kahn's Algorithm

```
input G # Graph to sort
local L # Contains sorted Vertices
local S # Vertices with no incoming edges
while !S.isEmpty()
    n = S.pop()
    L.append(n)
    for m in G.adjacent(n)
        G.removeEdge(n, m)
        if m.inDegree == 0
            S.push(m)
if G.edgeCount() > 0
    error # G has a cycle
return L
```

Something new

- Link to “Almost Done” solution
- https://goo.gl/HrKlyJ

Problem

- https://open.kattis.com/problems/reactivity