6.1 Topological Sort

Thursday, October 17, 2024 11:35 AM

- Application of Decrease and Conquer

- Sorting a digraph.

digraph :- directed graph.

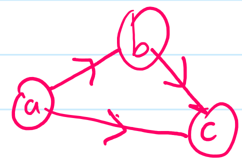
a graph in which edges have direction

e.g

$G = \langle V, E \rangle$

$V = \{a, b, c\}$

$E = \{ \langle a, b \rangle \ \langle b, c \rangle \ \langle a, c \rangle \}$



- DFS and BFS both work in digraphs
- Digraphs can also be stored as
  - adjacency matrices
  - adjacency Lists

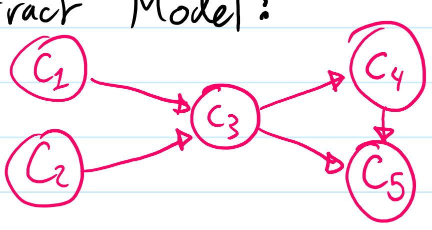
Example:

Suppose you have course C_1, C_2, C_3, C_4, C_5

- C_1 and C_2 have no prerequisits
- C_3 requires C_1 & C_2
- C_4 req. C_3
- C_5 req C_3 and C_4

In which order should you take the classes?

Abstract Model:

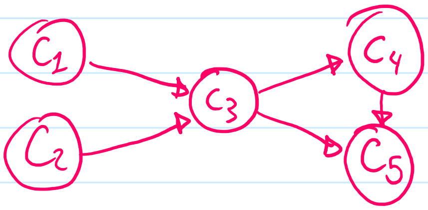


Problem: (re-stated)

order the vertices $\langle V_0, V_1, V_2, V_3, \ldots V_k \rangle$
in such a way that for every edge $\langle V_a, V_b \rangle$
$a < b$ in the order.

Note: not possible if the digraph has a cycle.
Note: digraphs without cycles are called DAG's
Directed Acyclic Graphs

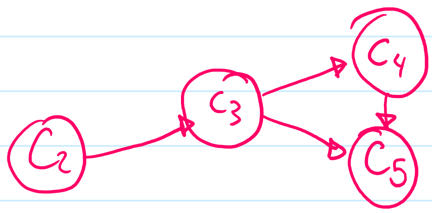
Solution: Decrease and conquer



Size = 5, How can I use somebody who can solve size = 4

(1) Get rid of one node; one with no incoming edges, "source" node.

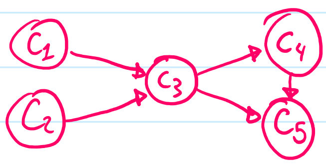
$\langle C_1,$ then:



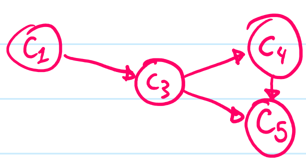
$\rangle$

Algorithm:
- Select a source node
- place it next in the order
- Simplify the graph

Trace

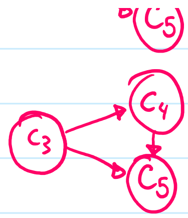


$\langle C_2$

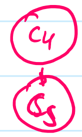


$\langle C_2, C_1$

$\langle C_2, C_1$



$\langle C_2, C_1, C_3$



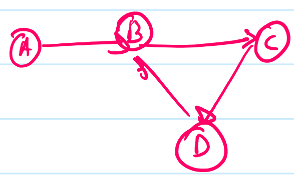
$\langle C_2, C_1, C_3, C_4$

$\langle C_2, C_1, C_3, C_4, C_5 \rangle$

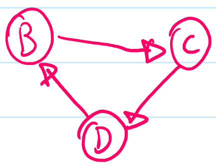
- What if the digraph has cycles?

  the algorithm will fail

e.g



$\langle A,$



Fail!

EOF.