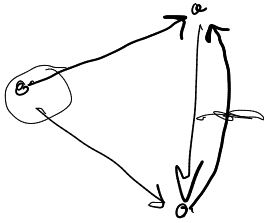


Directed graph

Def'n A directed graph is just like an undirected graph, but edges are ordered pairs of vertices, $E \subseteq V \times V$.



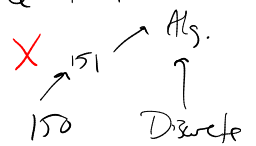
Examples:

- Roads
- Communication links
- Chunks + ladders
- Paths w/ directionality (rivers, conveyors, ...)
- Classes + prerequisites
- Markov processes
- Module imports
- Tasks, recipes
- Flowcharts

Dir. Cycles?



airline routes



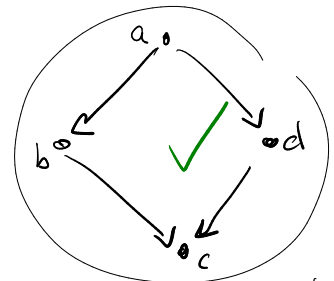
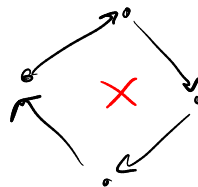
Generalizing undirected $\rightarrow$ directed:

path $\rightarrow$ directed path,
(Follow edges in correct direction)

degree $\rightarrow$ indegree/outdegree
 $\uparrow$ # edges pointing in $\uparrow$ # pointing out.

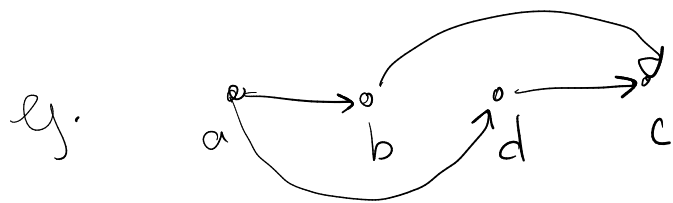
DFS, BFS still work!

Def'n A directed acyclic graph (DAG) is a directed graph with no directed cycles.



Def'n A topological ordering (topological order, topological sort, topsort) of a directed graph is an ordering of vertices $v_1, \dots, v_n$ such that for every edge $(v_i \rightarrow v_j)$, $i < j$.

i.e. all edges point to the right.



Thm. A directed graph G has a topological ordering
iff G is a DAG.

Proof. $(\Rightarrow)$ If G has a top. ordering, it cannot have any directed cycles, since any directed path can only go to the right through the ordering, so can't return to the start.

$(\Leftarrow)$ Suppose G is a DAG; we will show it has a top. ordering.

Lemma Any DAG has a vertex with indegree 0.

Proof Pick a vertex + follow edges backwards. You will eventually get stuck @ vertex w/ indegree 0. If not, you would eventually have to repeat a vertex (since G is finite) which would make a directed cycle.

Proof by induction on # of vertices / recursive algorithm

Base case: if $n=1$, one vertex, hence top. ordering.

if $n > 1$, find a vertex [✓] w/ indegree 0, put it first.

Note $G - \{v\}$ is still a DAG, so by IH it has a top. ordering, which can go after v .