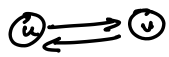
Lecture #17

Chat Mods: Emerson & Pooja

Topics: - Directed Acyclic Graphs
- Topological ordering/Sorting
- Depth-First Search
  - with pre-/post-#ing
- DFS & cycle detection

Undirected

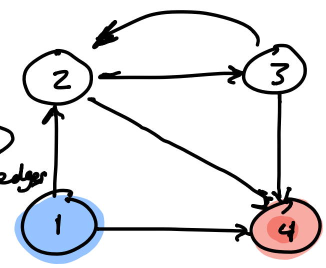
Directed



Directed Acyclic Graphs (DAG)

At least 1 source (vertex where all edges depart from)

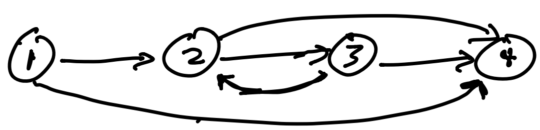
" 1 sink (vertex where all edges end at)



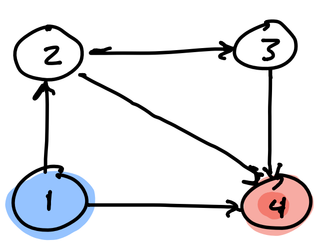
Topological Ordering:

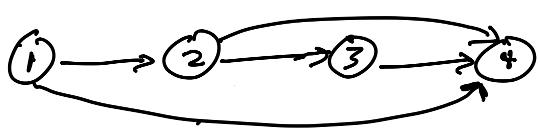
Order of a set is a binary relation between any two values

- transitive $x \prec y$ $y \prec z$ $x \prec z$
- anti-symmetric $x \prec y$ $y \prec x$ $x = y$
- reflexive $a \in A$, $(a,a) \in P$



Lemma: A graph can be topologically ordered iff it's a DAG.





Top_ordering algorithm:

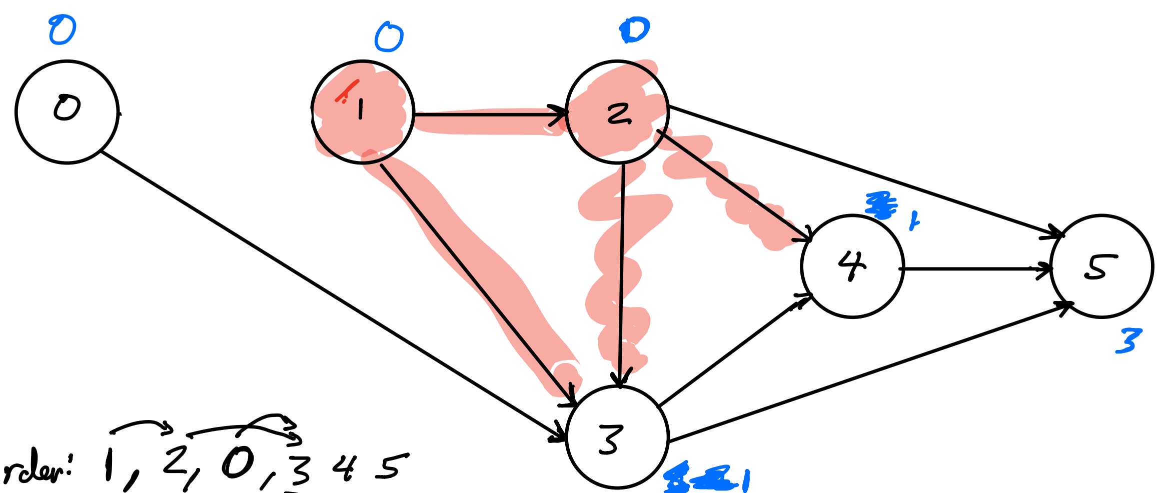
Step 1: for every count # of incoming edges

Step 2: Put all vertices into list with the same #

0: 1   1: 2, 3   2: ~~3~~ 3: 4

Step 3: At least 1 vertex(v) has a degree of 0, then that's a source. Add v to our order

Step 4: Delete v and reduce the edge count of the corresponding vertices



Order: 1, 2, 0, 3 4 5

Depth-First Search

DFS (G, u) time++ pre[u] = t

Mark u as visited

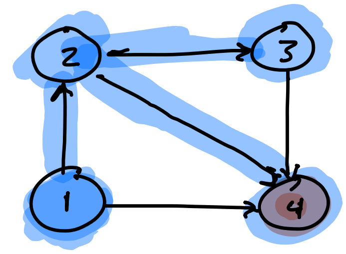
For each v in Adj(u)

if v not visited

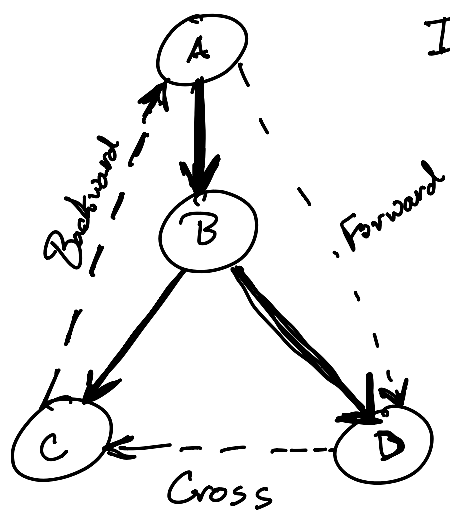
add edge(u,v) to T

set pred(v) to u

DFS(v)



post[u] = t



If we have a backward edge we have a cycle

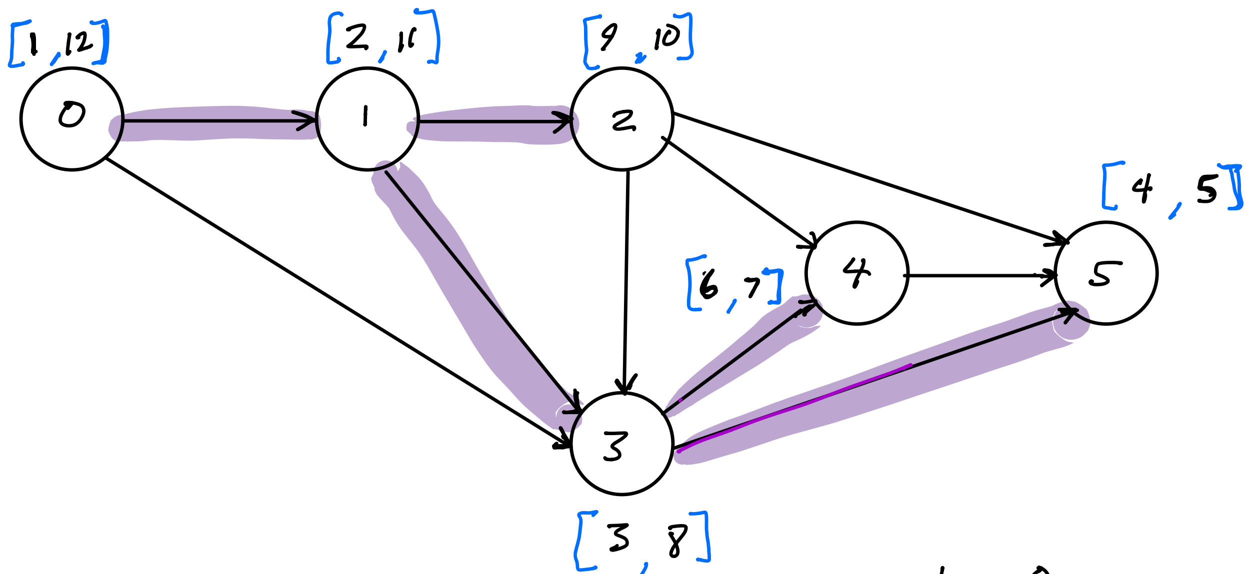
Forward exist if (x,y)

$pre(x) < pre(y) < post(y) < post(x)$

Backward

$pre(y) < pre(x) < post(x) < post(y)$

[pre, post]



Order by post #   5, 4, 3, 2, 1, 0

Order by decreasing post #: 0, 1, 2, 3, 4, 5

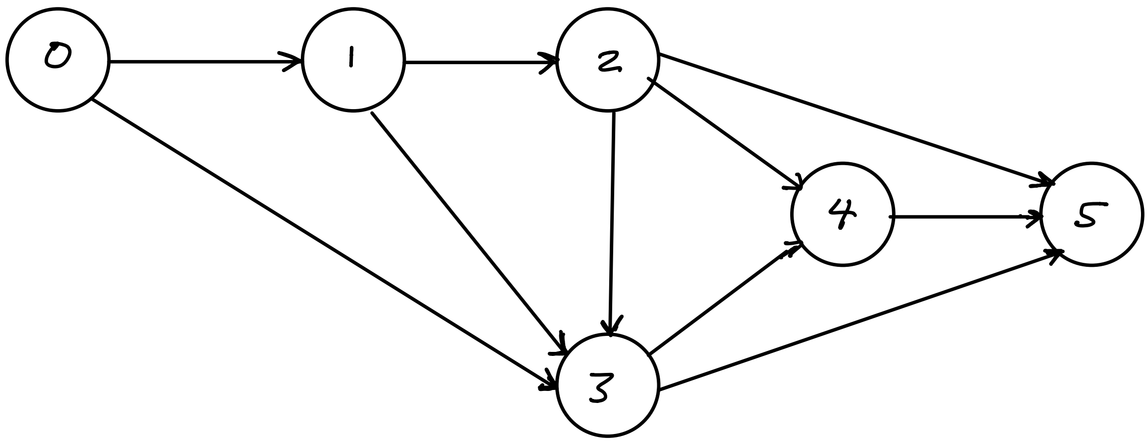
Topological Order

Run DFS w/ pre/post $O(n+m)$

Order by decreasing post $O(n\ \ )$

= Topological order

Topological Sort Example:



Topological Sort = 0, 1, 2, 3, 4, 5