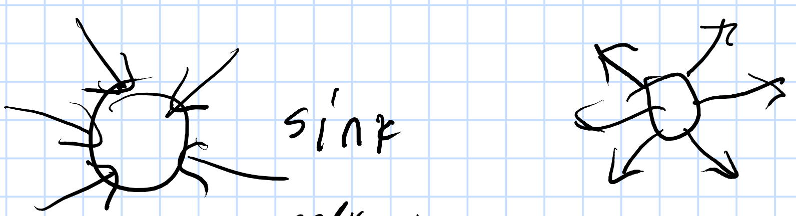
374 10/20/22

Lecture 18

DAGs
DFS pre-post numbering
Topological sort
SCC
SCC in linear time

DAG = directed acyclic graph

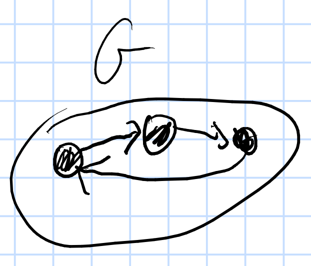
sink is a vertex with only incoming edges



source is a vertex with only outgoing edges

Claim
If G is a DAG then it has at least one source and at least one sink.

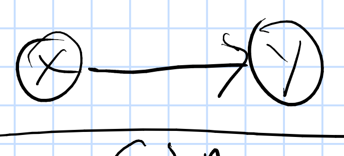
Proof



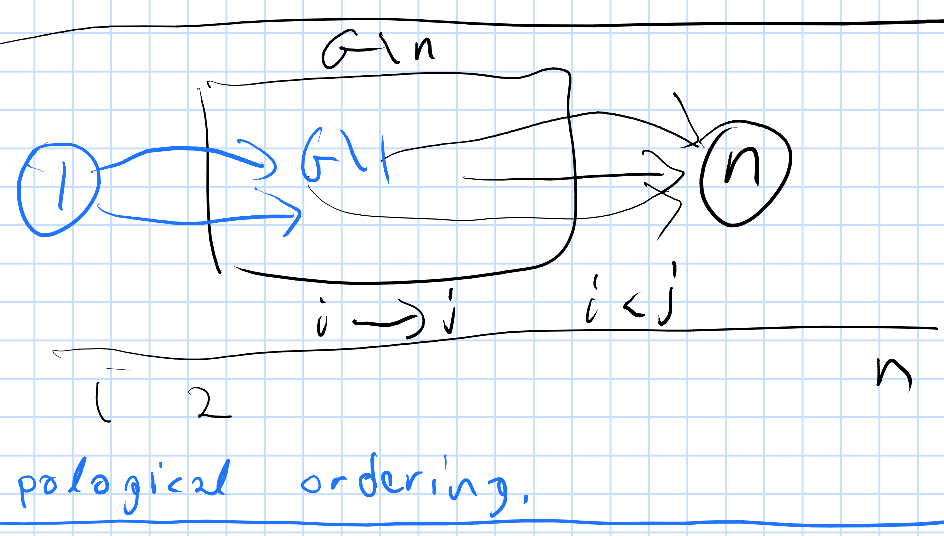
if go back in the walk to a visited vertex then found a cycle. Contradiction.

Claim If G is a DAG then any subgraph H of G has a sink/source.

$H=(V',E')$ is a subgraph of $G=(V,E)$ if $V'\subseteq V$ and $E'\subseteq E$.



X "<" Y



Topological ordering.

G is a DAG $\Longleftrightarrow$ G has a topological ordering

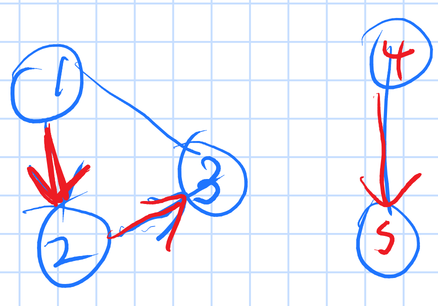
A topological ordering of a directed graph with n vertices, is a numbering of the vertices $\pi: V \to \{1,\dots,n\}$ s.t. $(u\to v)\in E(G)$ $\pi(u)<\pi(v)$.

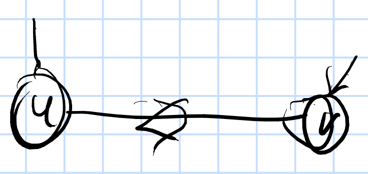
DFS (depth first search)
(greedy search)

undirected graph

```
DFS(G):
  ∀v∈V visited[v] ← false
  time ← 0
  while ∃v that is not visited do
      dfs_inner(v)
dfs_inner(v):
  visited[v] ← true
  pre[v] = ++time            // pre (order) numbering
  for u adjacent to v do
     if u is not visited then
         dfs_inner(u)                 ] 2m time
  post[v] ← ++time.          // post # of vertex
```

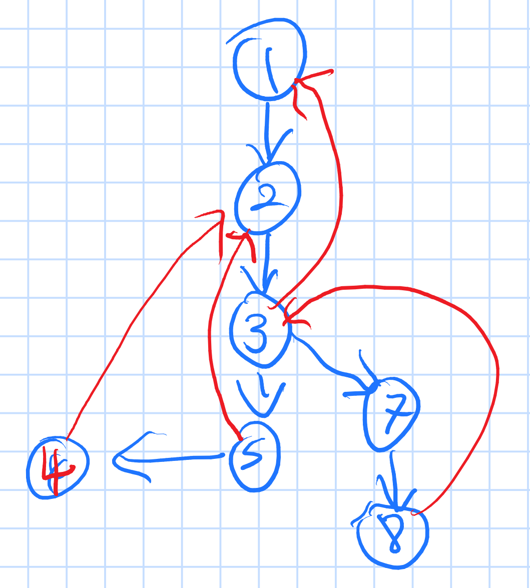
O(n)





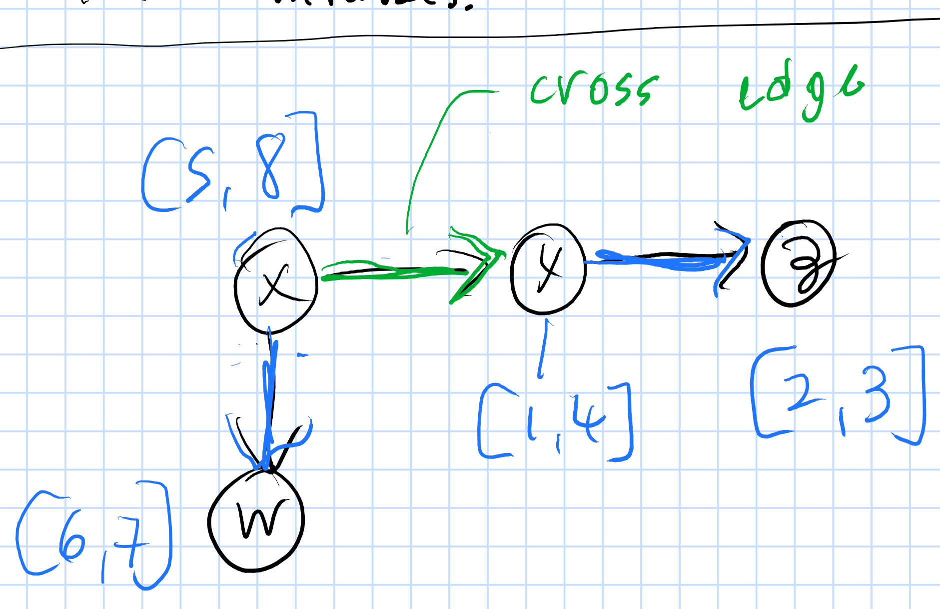
O(n+m)

n−1

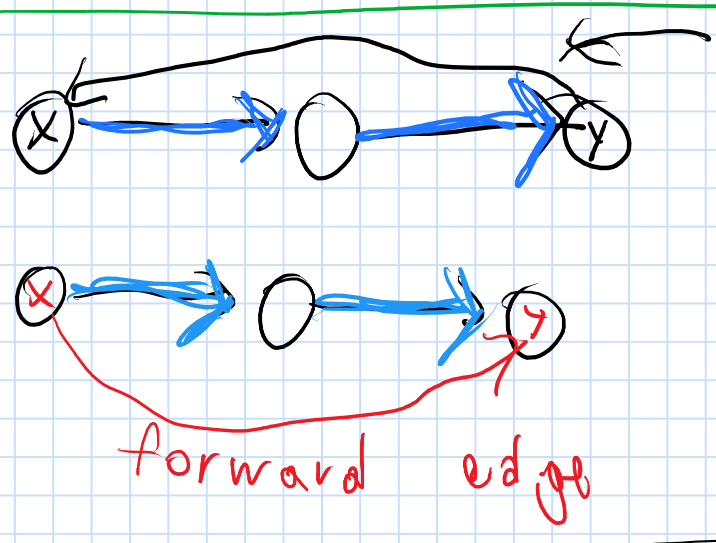


[pre(v), post(v)] − time interval when dfs(v) was running
[pre(u), post(u)]

$xy\in E(G)$
− $xy\in E$(DFS tree)
− back edge can decide if x is an ancestor of y (or vice versa) by inspecting time intervals.



if $pre(x) \overset{?}{<} pre(y)$
$\Rightarrow$ dfs_inner(y) was done inside dfs_i(x)
$\Rightarrow$ $pre(x) > pre(y)$
$T(x) > T(y)$



back edge
$T(x) \subseteq T(y)$
[pre(x), post(x)]
$T(x) \supseteq T(y)$

Lemma
A dir graph G has a cycle $\Longleftrightarrow$ there is a back-edge.

Proof