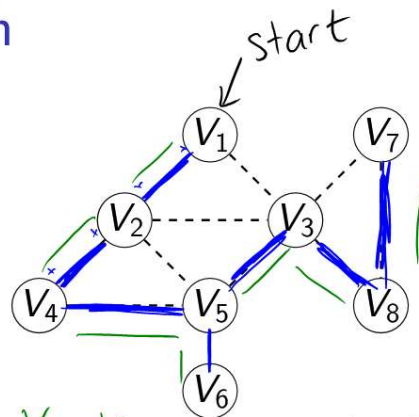
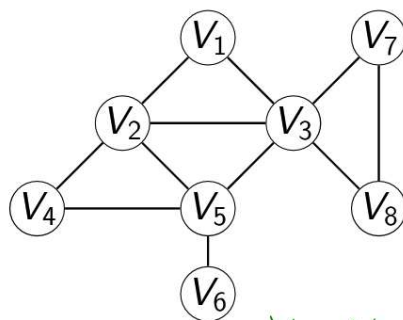


CompSci 161

Spring 2021 Lecture 4:

Graphs: DFS and Topological Sort

2 Depth-First Search



Depth-First
Traversal:

$V_1 \ V_2 \ V_4 \ V_5 \ V_6 \ V_3 \ V_8 \ V_7 \ V_2$
 $V_1 \ V_2 \ V_4 \ V_5 \ V_3 \ V_8 \ V_7 \ V_6$

Tree: n vertices, $n-1$ edges
 acyclic, connected

3

Running Time of DFS?

n : #vertices
 m : #edges

DFS-iterative(s)

$\forall v$ discovered[v] = false } $O(n)$ to initialize

$S \leftarrow$ empty stack

push s to S

while S is not empty **do** } Total iterations: $O(m)$

$u \leftarrow \text{pop}(S)$ } $O(m)$ total pops

if discovered[u] = false **then** } degree of u ,
 $\#$ edges incident

discovered[u] = true } $O(n)$ total

for all edges (u, v) **do** } $\delta(u)$

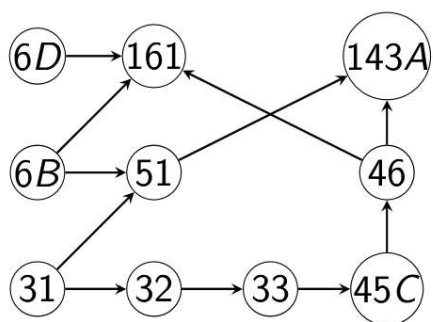
push(S, v) $\leftarrow O(1)$

Total: $\sum \delta(u)$
 is $O(m)$

Total work done
 is $O(nm)$

4

Prerequisite Graph



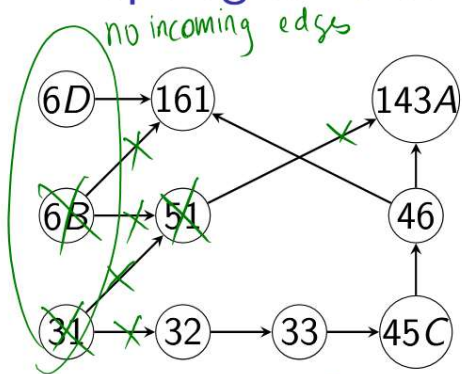
Edge (u, v) : must take u before v .

What would the neighbor set $N(u)$ indicate?

5

Topological Order

Directed Acyclic Graph ("DAG")



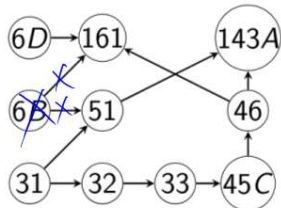
6B 31 51

$[u \text{ before } v] \rightarrow \text{no path } v \text{ to } u$

- What must be true for topological order?
acyclic (no cycles)
- Find a topological order of this graph.

6

Topological Sort



- How do we topological sort?

Bag b;

std::vector<unsigned> count(n,0)
for each $e=(u,v)$ } $O(m)$
 $\text{count}[v]++$

for each $v \in V$ } $O(n)$
 if $0 == \text{count}[v]$
 $b.add(v)$

while ! b.empty()

{
 $x \leftarrow b.remove()$; out x ; } $O(1)$

for each $y \in \text{adj}[x]$
 $\text{count}[y]--$

if $0 == \text{count}[y]$ } $O(\delta(x))$
 $b.add(y)$
}

$O(n+m)$

(out, add to
vec to return,
etc