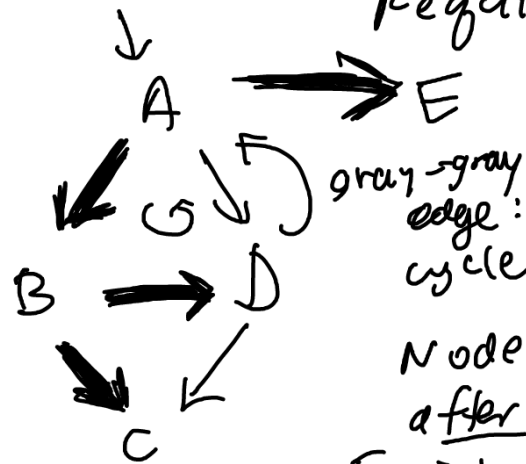


Topological Sort, Prelim Review

Depth-First Search.

```
void visit(Node g) {  
  for each (g → v) {  
    if (v.color == white) {  
      v.color = gray;  
      visit(v);  
    }  
  }  
  g.color = black;  
}
```



Exam rooms: Kennedy 5:30
netid A-F → Bailey 7:30
G-Z

Effect: mark as black
all nodes reachable
from v on all-white
paths.

Requires: g is gray

call tree creates
a tree w/in graph
(dark = edge)

Node finish (colored black)
after all descendants.

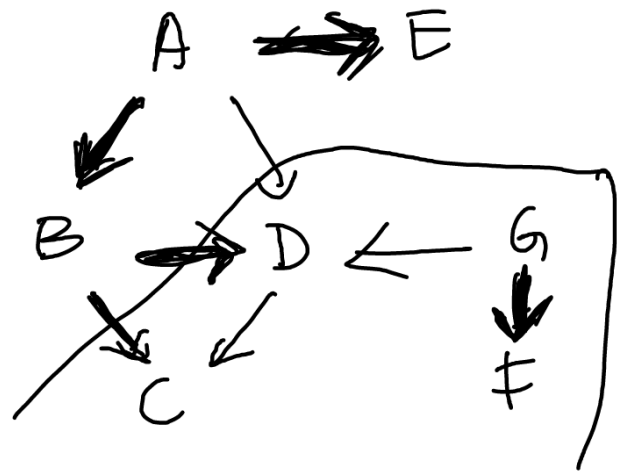
Finishing order: C, D, B, E, A

edge = dependency? v finishes after all of its dependents

topological sort: put node before all its dependents

toposort is just reverse of postorder traversal!

C D B E A | F G



G | F | A E B D C : topological sort
 $\Rightarrow$ run DFS repeatedly from arbitrary white node until all nodes reached.

Run time?

$O(V)$ calls to visit

$O(E)$ edges

$O(V+E)$

A E B G F D C

$$f(n) = n \lg n$$

is it

is it

$$O(n^2) \quad ?$$

$$O(n \lg n)$$

$$O(n \lg n)$$

$$\& \Omega(n \lg n)$$

A: Yes B: No

Yes: tighter bound.

$\Omega(n)$? Yes

is $f(n) = 10n + 100$ $O(n)$?

witness pair (k, n_0) such that ^{all} for $n \geq n_0$,
 $f(n) \leq k \cdot n$

$$10n + 100 \leq kn$$

$$10n + 100 \leq 20n$$

$$10n \geq 100$$

$$n \geq 10$$

$$k=20$$

$$k=100$$

$$(k, n_0) = (20, 10)$$

Painting in Swing

not called by
you, called
by event dispatch thread.

JComponent



```
paint(g) {
```

```
}
```

```
paintComponent(g);
```

```
}
```

```
repaint();
```

— called by you,
not overridden.
— notifies Swing
that paint is
needed.

```
MyComponent {  
    paint() {  
        // all-new  
    }  
}
```

```
paintComponent() {  
    :  
}
```

Thread-safe

safety $\Rightarrow$ synchronization.

— only one thread has access to mutable state at a time.

— locks ensure that only one thread is holding lock.

$\Rightarrow$ guard mutable state with a lock.

Monitors: guards private state (ivars) with a lock.
"synchronized" methods enforce thread safety.
all public methods are synchronized $\rightarrow$
blocking abstractions: need condition variable.

Condition variables (always have a mutex)

wait() : block current thread until notified.
requires: holding the mutex

```
synchronized (m) {
```

```
    while (!condition) m.wait();
```

```
}
```

↑ accesses mutable state guarded by m

← mutex held

← mutex released

← mutex held

notifyAll()

: All waiting threads on this CV stop waiting & try to acquire mutex

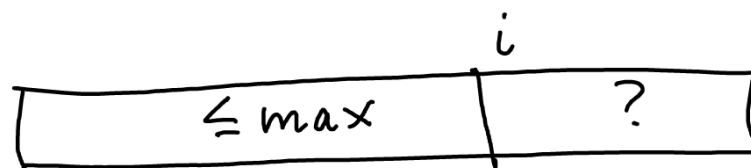
Requires: holding mutex

1. Establishment
2. Postcondition
3. Preservation
4. Termination.

$a[0] = \max(a[0..i])$
 $i \geq a.length \ \& \ i \leq a.length \Rightarrow i = a.length$
 So $\max = \max(a[0..a.length])$
 if $(\max < a[i])$ then $a[i] = \max(a[0..i])$
 else $\max = \max(a[0..i])$
 $DF = a.length - i$
 $DF' = a.length - (i+1) < DF$
 $DF' \leq 0 \Rightarrow$ loop guard false

```

int [] a = ... ; // assume non-empty.
int max = a[0];
int i = 1;
while (i < a.length) {
  if (max < a[i])
    max = a[i];
  i++;
}
  
```



$\max = \text{maximum of } a[0..i]$

$1 \leq i \leq a.length$