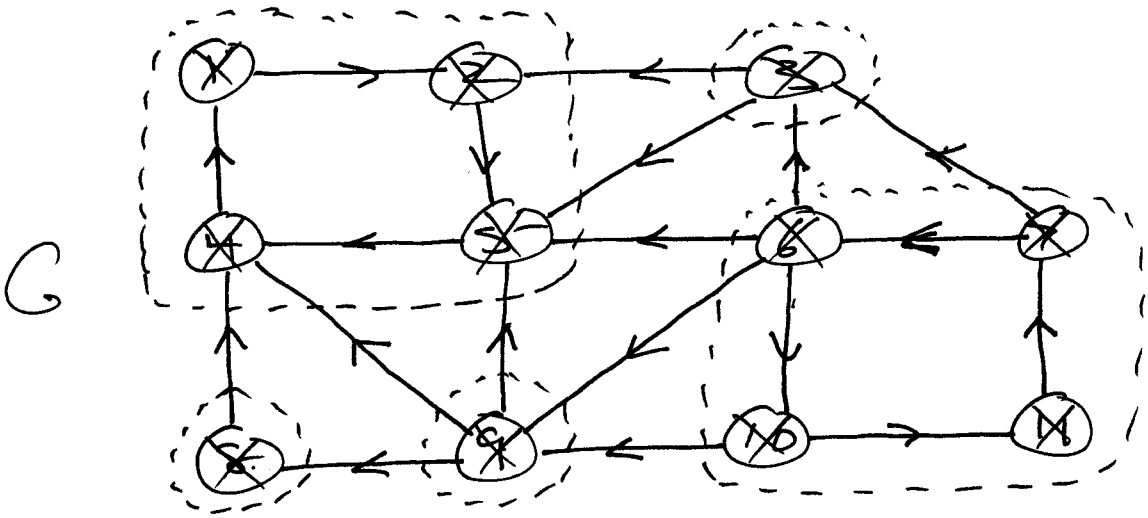
CSE 101 1-28-21

Pas: ext. 1 last day

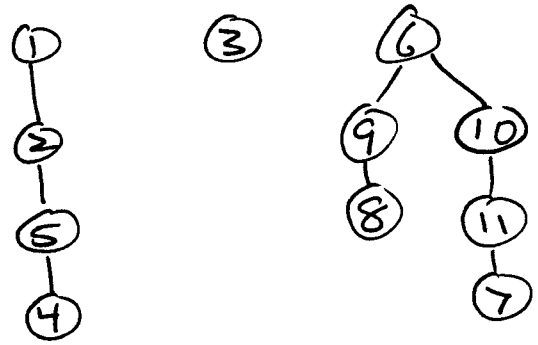
Pas example



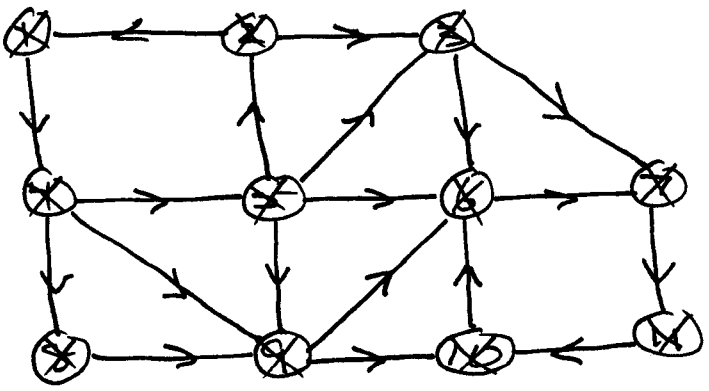
stack

6 ✓
10 ✓
11 ✓
7 ✓
9 –
8 –
3 –
1 –
2 –
5 –
4 –

forest



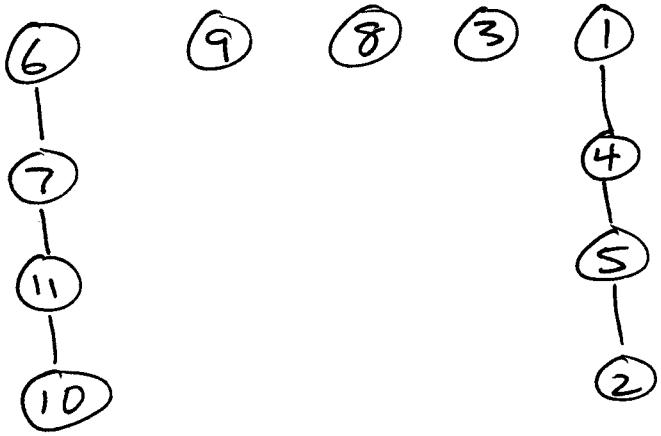
G^T



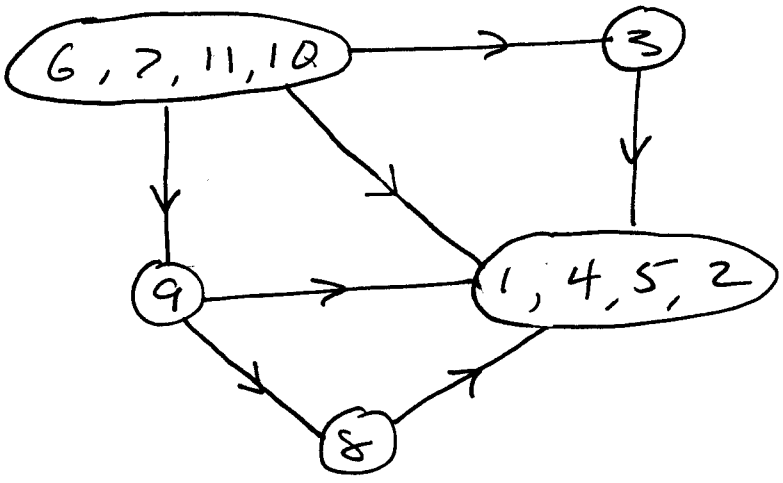
stack P

1.	n
4	
5	
2	
3	n
8	n
9	n
6	n
7	
11	
10	

forest

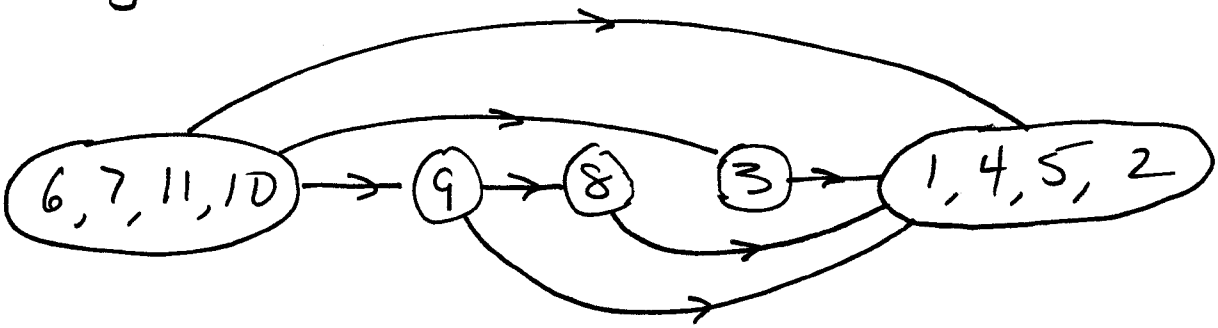


G^{SCC}



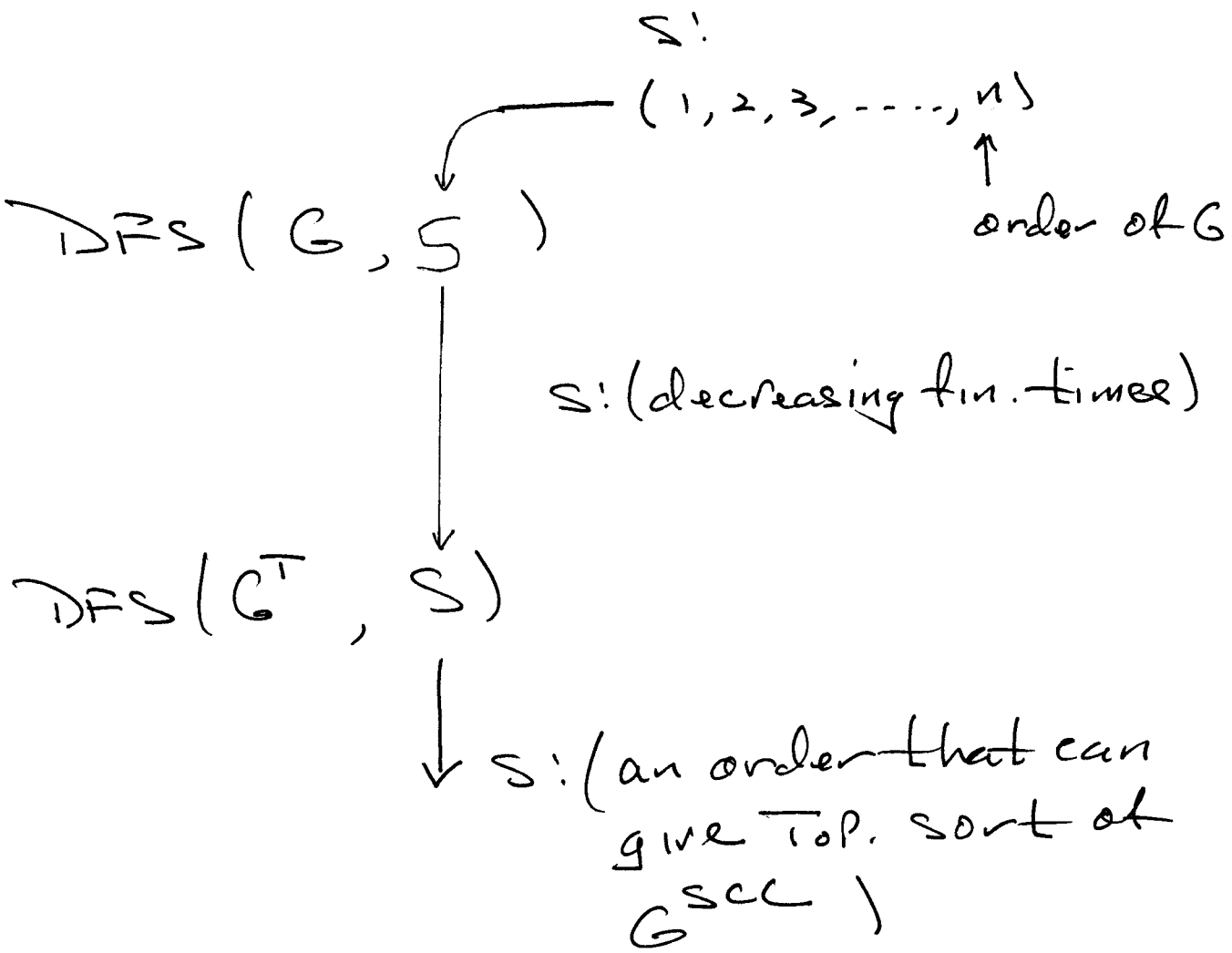
acyclic

Topological sort



output file

component 1 : 6 7 11 10
" 2 : 9
" 3 : 8
" 4 : 3
" 5 : 1 4 5 2


S:
(1, 2, 3, ..., n)
order of G
DFS(G, S)
S: (decreasing fin. times)
DFS(G^T, S)
S: (an order that can give Top. sort of G^SCC)

Algorithm Efficiency & Runtime

How do we measure runtime?

Not in seconds!

Instead we count # of executions of some basic operation (barometer OP.)

Ex.

```
A(n)
1. for i = 1 to n
2.     for j = 1 to n
3.         print('blah')   ← basic operation
```

(# print stmts.) = n^2 ← function of input size n.

Ex.

```
B(n)
1. for i = 1 to n
2.     for j = 1 to i
3.         print('blah')   <-- basic op.
```

(# print stmts) $= 1 + 2 + 3 + \cdots + n$

$$= \frac{n(n+1)}{2} = \underbrace{\frac{1}{2}n^2}_{\text{high}} + \underbrace{\frac{1}{2}n}_{\text{low}}$$

notice: (1) term $\frac{1}{2}n$ is negligible for large values of n.

$$\frac{\frac{1}{2}n^2 + \frac{1}{2}n}{\frac{1}{2}n^2} = \left[1 + \left(\frac{1}{n}\right)\right] \longrightarrow 1$$

$\left(\frac{1}{n}\right) \downarrow 0$

so we might as well consider $\frac{1}{2}n^2 + \frac{1}{2}n$ to be $\frac{1}{2}n^2$

(2) now compare A (runtime n^2)
to B (runtime $\frac{1}{2}n^2$)

we seek a measure of runtime that is independent of computing device. Since A & B can be equalized by running A on a 2x ~~faster~~ machine, we shouldn't consider n^2 & $\frac{1}{2}n^2$ different. i.e. we don't distinguish between A and B.

we call our measure of runtime: Asymptotic runtime.

informal definition

- define basic op.
- determine # basic ops. on input of size n
- drop 'lower order' terms
- replace coeff. of 'highest order' term by 1.