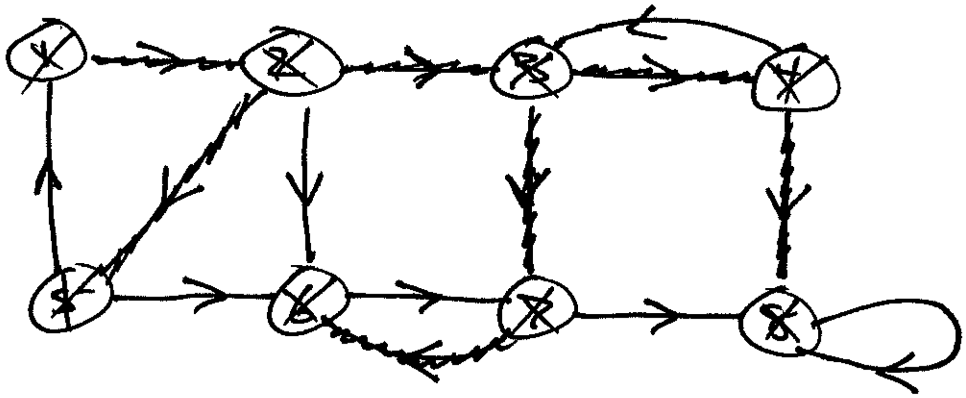
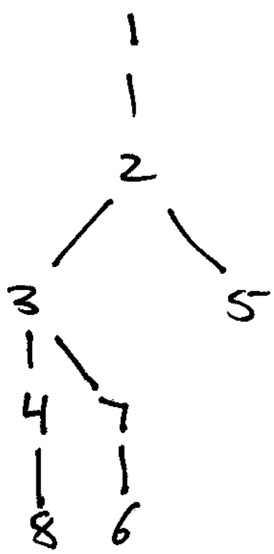
CSE 101-01 4-25-23

Ex.

G



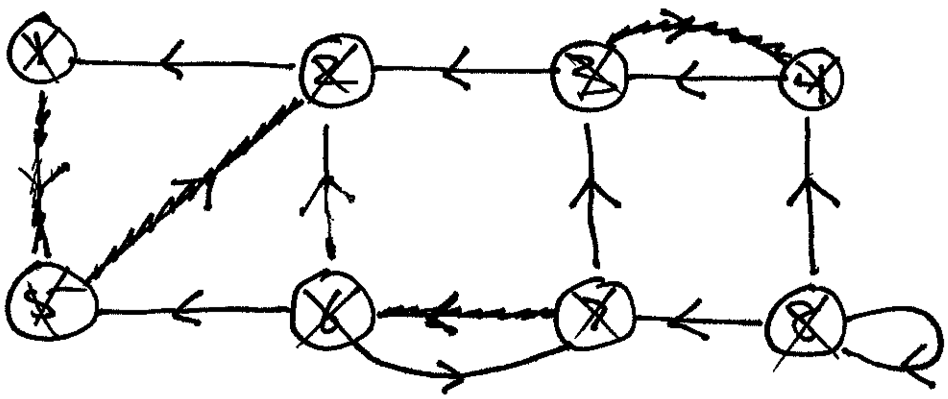
forest



stack

1
2
5
3
7
6
4
8

G^T



forest

1 — 5 — 2   3 — 4   7 — 6   8

stack

8
7
6
3
4
1
5
2

scc: $\{1,5,2\}, \{3,4\}, \{7,6\}, \{8\}$

$\quad = C_1 \quad = C_2 \quad = C_3 \quad = C_4$

Defn

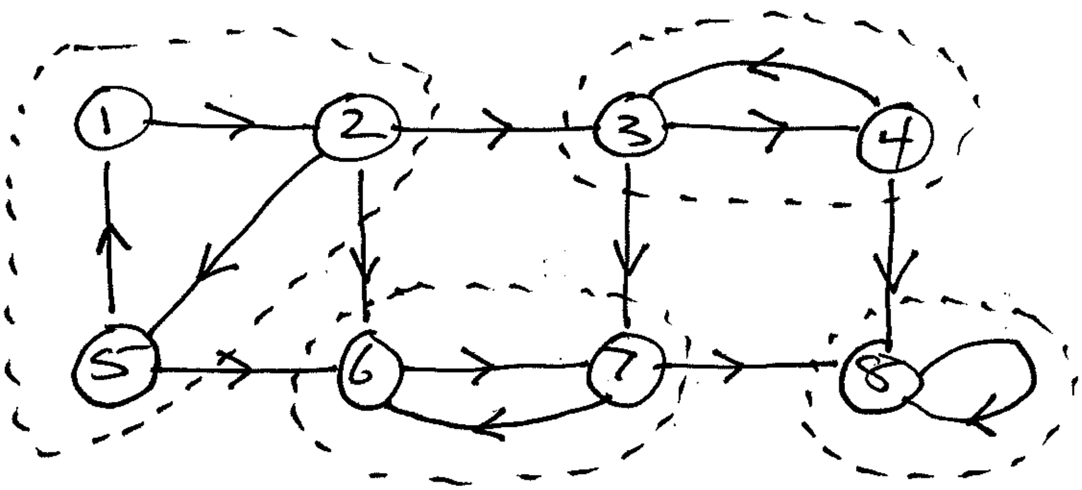
The component graph of G (also called the condensation graph), denoted G^{scc}, is the digraph with:

$$V(G^{scc}) = \{\text{strong component of } G\}$$

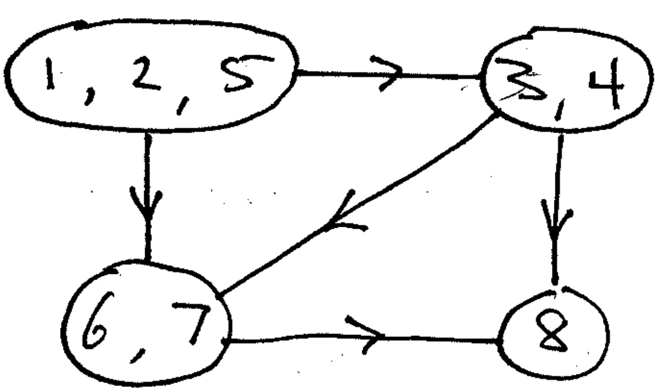
$$E(G^{scc}) = \{(C_i, C_j) \mid \text{there exists } x \in C_i \text{ and } y \in C_j \text{ s.t. } (x,y) \in E(G)\}$$

Ex.

G

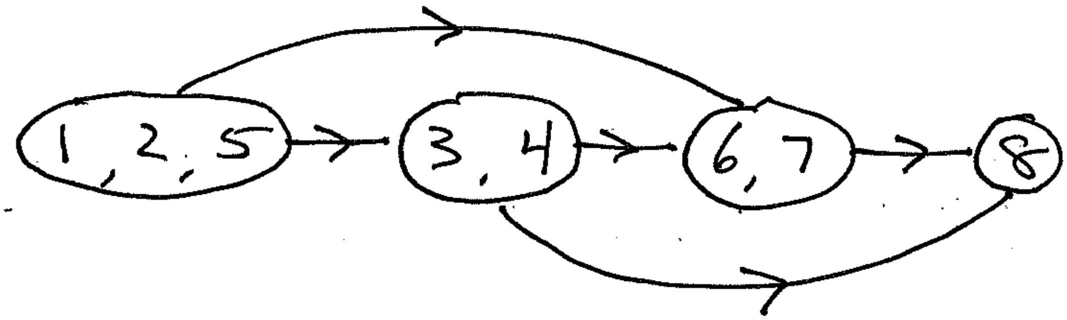


G^{scc}



This is a DAG, so has a topological sort.

Top. Sort!



note:

$$\left(G^{SCC}\right)^{T} = \left(G^{T}\right)^{SCC}$$

Stack	Print
8	Comp 1: 1 5 2
7	Comp 2: 3 4
6	Comp 3: 7 6
3	Comp 4: 8
4	
1	
5	
2	

DFS Prototype

void DFS(Graph G, List S);

$$
\begin{array}{l}
\quad\quad S: 1\ 2\ 3\ \ldots\ n \\
\downarrow \\
\text{DFS}(G, S) \\
\quad \downarrow\ S: \text{decreasing finish times} \\
\text{DFS}(T, S) \\
\quad \downarrow
\end{array}
$$

use order of S to obtain top. sort of G^{SCC}

Algorithm Runtime (handout)

Ex. algorithms A, B

A : # basic ops $= n^3$

B : # basic ops $= 10n^2$

Asymptotic Growth of fcns

Defn

let $f(n), g(n)$ be func. write

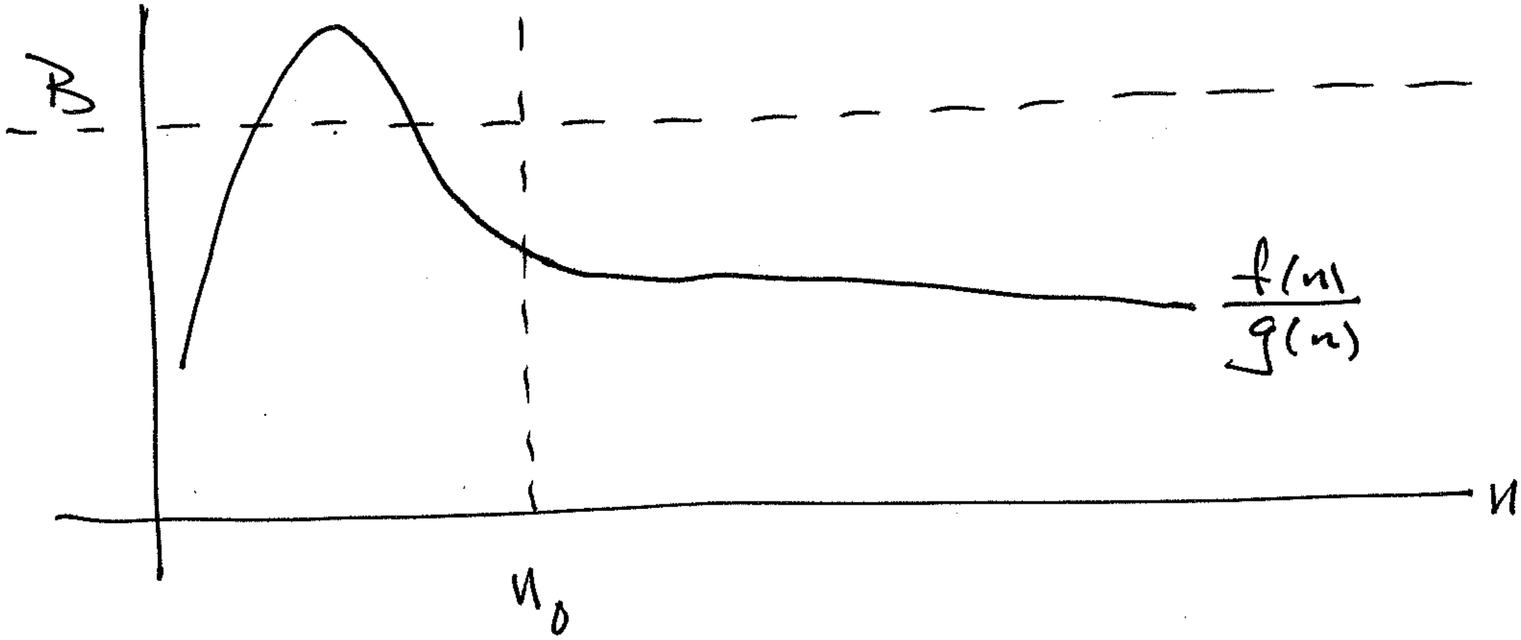
$$f(n) = O(g(n))$$

iff there exist $B > 0$, $n_0 > 0$ s.t.
for all $n \geq n_0$:

$$\frac{f(n)}{g(n)} \leq B$$

we say:

- $g(n)$ is an asymptotic upper bound for $f(n)$



Ex. let $f(n) = 2n+5$, $g(n) = n$. Then

$$\frac{f(n)}{g(n)} = \frac{2n+5}{n} = 2 + \frac{5}{n}$$

let $B=3$, $n_0 = 5$. Then:

$$\boxed{\text{if } n \geq 5, \text{ then } 2 + \frac{5}{n} \leq 3}$$

Proof: $\boxed{2n+5 \leq 3n} \Longleftrightarrow \boxed{5 \leq n}$

$$\boxed{2 + \frac{5}{n} \leq 3}$$