

## CSE 101 – Nov 22, 2019 (Week 8)

Notes provided by Ben Sihota bsihota@ucsc.edu

---

### Find a topological sort of a DAG

1. Run  $\text{DFS}(G)$
2. As vertices finish, push them onto a stack (equivalent: sort by decreasing finish time)
3. When complete, the stack is a topological sort

### Theorem

Last bullet point is true. Proof in book.

**Exercise:** Figure out how to sort an array using topological sort

### Strongly connected components

Let  $G = (V, E)$  be a digraph. Recall a S.C.C. of  $G$  is a subset  $U \subseteq V$  satisfying

1.  $U$  is strongly connected
2.  $U$  is maximal W.R.T (1)

### DFS can be used to find SCCs of $G$

1. Call  $\text{DFS}(G)$ , as vertices, push them onto a stack
2. Compute the transpose:  $G^T$ . (Reverse all directed edges). **Note:**  $A(G^T) = A(G)^T$
3. Call  $\text{DFS}(G^T)$ , process main loop of DFS by popping vertices off the stack

### Theorem

Trees ? 2nd DFS are the SCCs of  $G$

### Component Graph: $G^{SCC}$

$$V(G^{SCC}) = \{ \text{SCC's of } G \}$$

$$E(G^{SCC}) = \{ (c_i, c_j) : \exists x, y, x \in C, y \in C \text{ s.t. } (x, y) \in E(G) \}$$