

# CSE 250 Recitation

April 3 - 7: PA3, Orderings, Priority Queues and Heaps

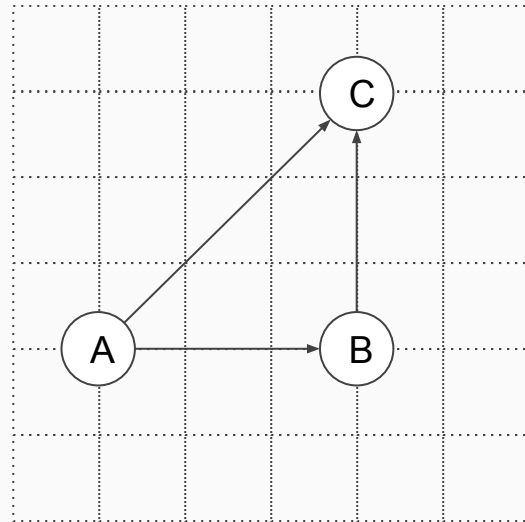


# PA3: Getting Started

- **PA3** is all about graph traversal
- As with **PA2**, you will start by first writing tests
- **REMEMBER:** You do not need to know how to implement a method to test that method!
  - For example, one of the methods finds the shortest path in a graph...you can still come up with example graphs, and know what the shortest paths are, without knowing how to find them algorithmically yet
- **That leads to our first exercise...coming up with good example graphs**

# PA3: Getting Started

- What is the adjacency list for the graph to the right?
- What should BFS find when start = A, end = C?
- **Start asking "what if?" questions. Try to think of things that might break, or issues your sample graphs and sample searches might not catch.**
  - For example, what if we did Dijkstra's instead of BFS...does the graph to the right differentiate between the two?



# Orderings

We know:

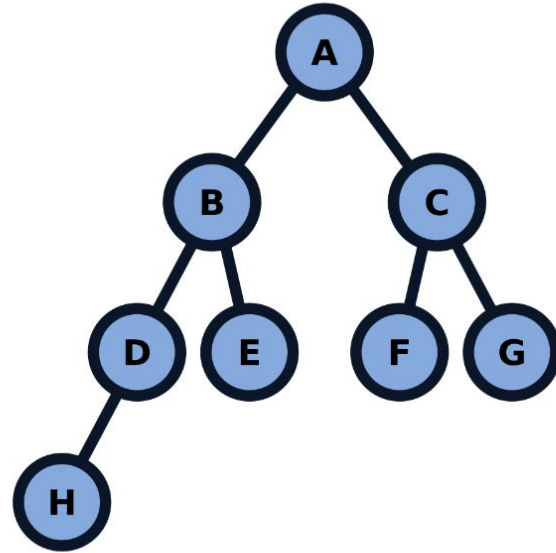
- $A < B$
- $A < C$
- $B < D$
- $B < E$
- $C < F$
- $C < G$
- $D < H$

- What other relationships can we infer?
- What is the smallest number of extra tests we need to...
  - Find the smallest value?
  - Find the second smallest value?
  - Find the third-smallest value?
  - Find the fourth-smallest value?

# Heaps

We know:

- $A < B$
- $A < C$
- $B < D$
- $B < E$
- $C < F$
- $C < G$
- $D < H$



# Heaps

Are the following arrays valid heaps?

9 7 4 5 6 2 3

# Heaps

Are the following arrays valid heaps?

20 7 15 2 5 12 9 6 4 1 3

# Heaps

Find tight bounds for inserting sequence of items into a heap when the sequence is already sorted in **descending order**.



# Heapify

Trace the execution of Heapify on the following array

9 6 8 1 5 4 15 3 7 14 11 10 2 13 12