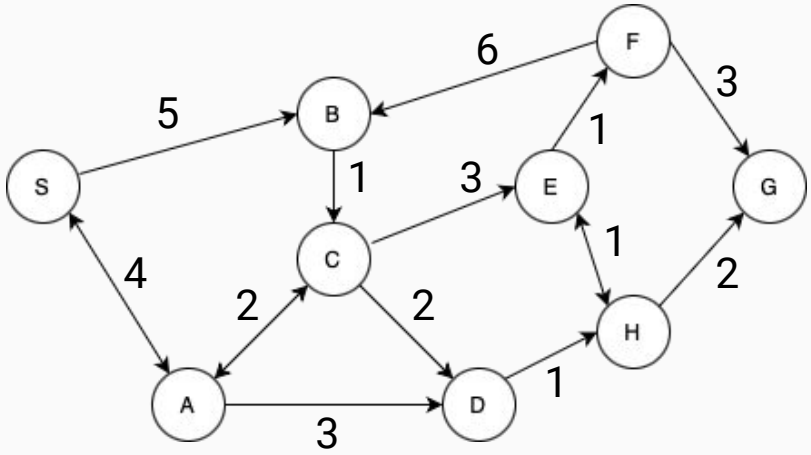
CSE 250 Recitation

3/30-3/31: Heaps, Dijkstra’s Algorithm

Graph Traversal



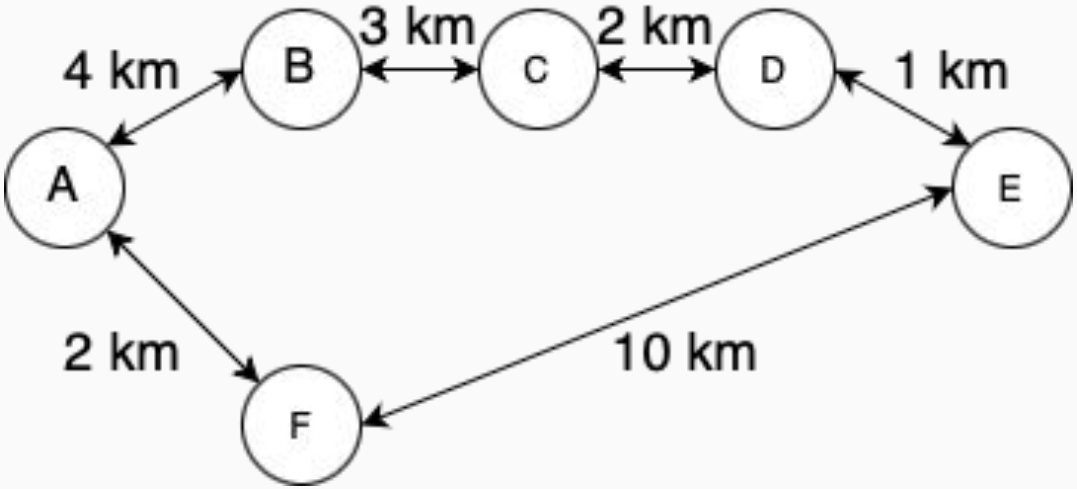
1. Insert an arbitrary starting node into the [TODO]
2. While the [TODO] is not empty:
   a. Remove a node from the [TODO]
   b. Mark the node as visited
   c. Insert all of the node's unvisited neighbors into the [TODO]

Try the above with TODO as a:

- **Stack (DFS)**
- **Queue (BFS)**
- **PriorityQueue (Djikstra's)**

Compare nodes being marked visited upon removal from TODO vs on add to TODO

Dijkstra’s Algorithm



- Path from C to F
- Path from A to E

Orderings

We know:

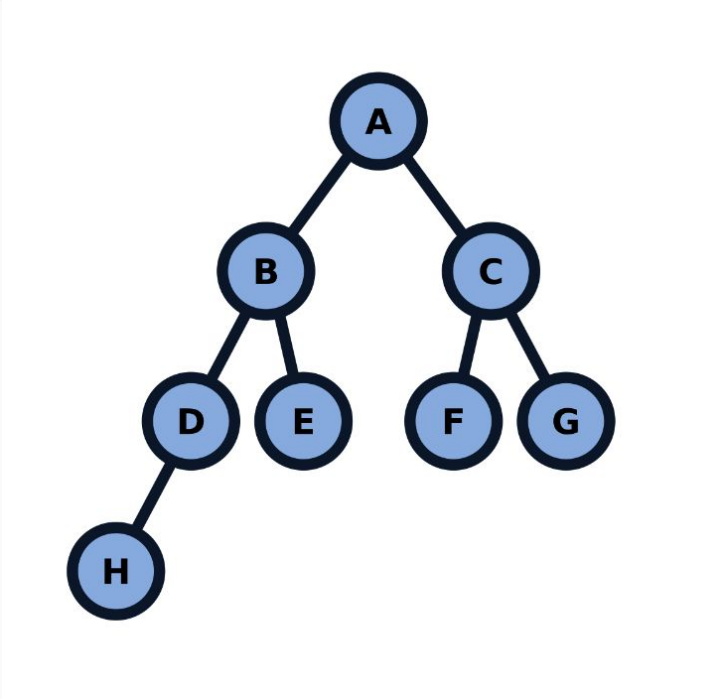
- A < B
- A < C
- B < D
- B < E
- C < F
- C < G
- D < H

- 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 max heap when the sequence is already sorted in **descending order**.