CS 430 Lecture 26 Activities

Opening Questions

1. What is the difference between a tree and a graph? All trees are graphs, but not all graphs are trees. In a tree, there is only 1 path between any 2 vertices. Trees are also acyclic since no item can have multiple parents.

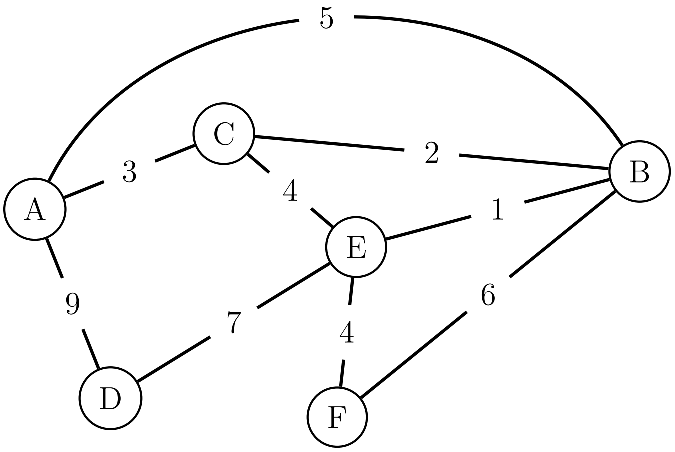
2. Give a recursive definition for a tree. Base case: single node with no children. A tree is … a node pointing to other trees with no cycles.

3. In a weighted undirected graph, what is the difference between a minimum spanning tree and a shortest path in a graph?

4. Since the shortest paths contain the shortest sub-paths (optimal substructure), name an algorithmic approach that we might try to find a shortest path in a graph.

Minimum Spanning Trees (MST)

1. Give a definition of a Minimum Spanning Tree, and find an MST of the below graph. Set of edges that connects all vertices with no cycles. $|V| - 1$ edges; not necessarily unique, you could have multiple MSTs for a graph.



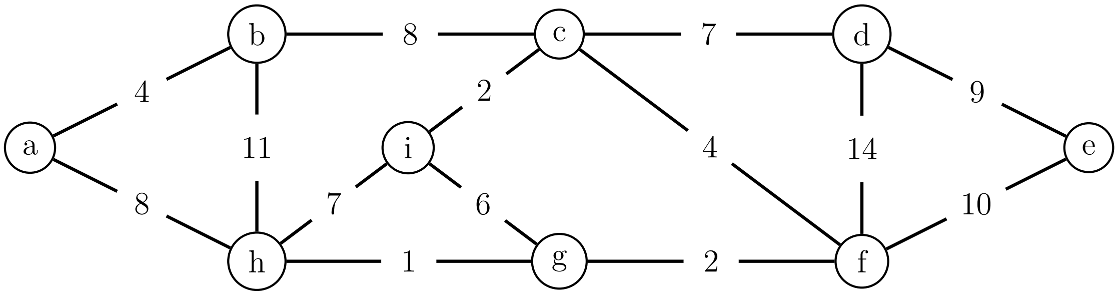
2. Prove a Minimum Spanning Tree has optimal substructure. Pick any subsets of adjacent vertices in an optimal MST. Those edges that connect that subset of vertices in the MST must also be MST

3. What are some possible greedy approaches to find a Minimum Spanning Tree? Prove correct or show counterexample.

   - Grow min edges first; no cycles.
   - Prune/remove max edge, but stay connected.
   - Creating edges from visited nodes to unvisited nodes.

- **Prim**: Min edge from visited vertex set to unvisited vertex set.
- **Kruskal**: Pick min edge that its vertices are not already in the connected component.

4. Demonstrate your MST algorithm on the following graph and write pseudocode.



Node	Visited?
a	
b	
c	Visited
d	
e	
f	
g	
h	
i	

Started at c.

Algorithm 26.1 Prim's Algorithm (MST)

```
function MST-Prim($G$, $w$, $r$)
    for all $u \in G.V$ do
        $u.key \leftarrow \infty$
        $u.\pi \leftarrow$ NIL
    end for
    $r.key \leftarrow 0$
    $Q \leftarrow G.B$
    while $Q \neq \emptyset$ do

    end while
end function
```

Algorithm 26.2 Kruskal's Algorithm (MST)

```
function MST-Kruskal

end function
```

Demonstration of Prim (Deleted): http://en.wikipedia.org/wiki/File:Prim-algorithm-animation-2.gif

Demonstration of Kruskal: https://www.cs.usfca.edu/~galles/visualization/Kruskal.html