

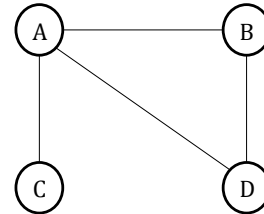
LECTURE 39: GRAPHS

Today

- Reading
 - JS Chapter 16
- Objectives
 - DFS for detecting cycles
 - Dijkstra's Algorithm
 - (Minimum spanning trees)

Detecting Cycles

- DFS Basic algorithm:
 - Push the start node
 - While the stack is not empty:
 - Pop a node
 - Check if node previously visited
 - If not, mark as visited and push all children
- Keep track of parents to detect cycles



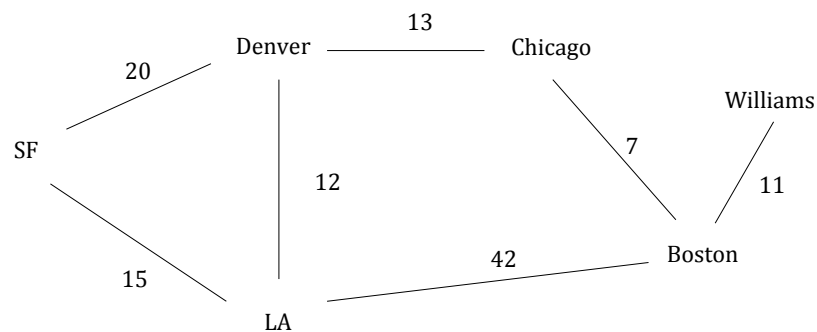
Detecting Cycles

- A cycle exists if,
 - A node in adjacency list has already been visited but it is not the node that added us to the stack
 - i.e. ancestor (not parent) in search tree already visited
 - Works for undirected graphs

Single Source Shortest Path

- Starting at node s , find shortest path to all other nodes
- If edges have no weight then can use BFS
 - Shortest path is defined to be the path with fewest edges
- If edges have (non-negative) weights, use Dijkstra's Algorithm
 - Dijkstra's Algorithm is BFS with a priority queue
 - The priority is the distance from the start node to current node
 - Keep track of parent node (i.e. preceding node in the path)

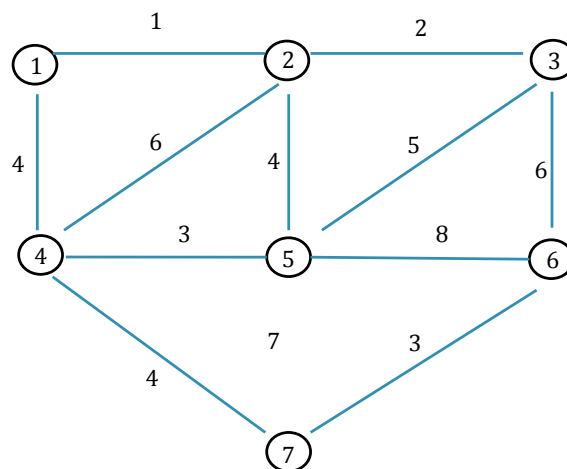
Single Source Shortest Path



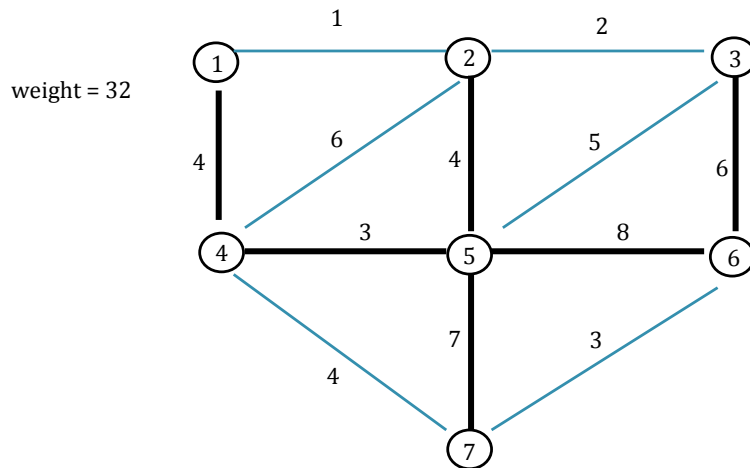
Minimum Spanning Trees

- $G' = (V', E')$ is a subgraph of $G=(V,E)$ if G' is a graph and V' is a subset of V and E' is a subset of E
- A spanning tree is a subgraph of G that is a tree and connects all of the vertices together
- A minimum spanning tree is a spanning tree whose weight is \leq the weight of any other spanning tree
- Weight is the sum of the weights of the edges

Graph



Spanning Tree



Minimum Spanning Tree

