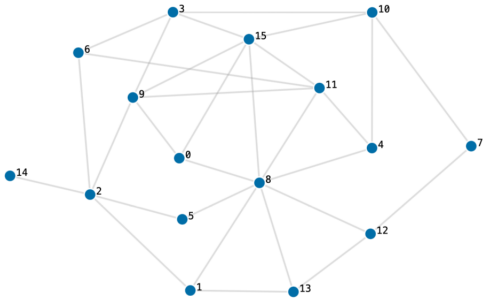
CSC 404 - Activity/Project 9 - Name:

Problem 1. Consider the following graph (16 Nodes and 30 Edges) – Give it a fun back story/relevance :-).



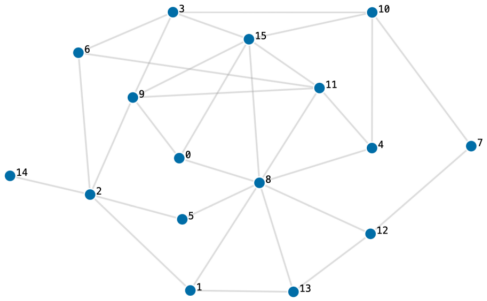
$d = \{0 : [8, 9, 15], 1 : [2, 8, 13], 2 : [1, 5, 6, 9, 14], 3 : [6, 9, 10, 15], 4 : [8, 10, 11], 5 : [2, 8], 6 : [2, 3, 11], 7 : [10, 12], 8 : [0, 1, 4, 5, 11, 12, 13, 15], 9 : [0, 2, 3, 11, 15], 10 : [3, 4, 7, 15], 11 : [4, 6, 8, 9, 15], 12 : [7, 8, 13], 13 : [1, 8, 12], 14 : [2], 15 : [0, 3, 8, 9, 10, 11]\}$

a. Using 0 as your source, construct a spanning tree for the given graph using a breadth-first process. Identify which nodes can be reached within 1, 2, 3, and 4 steps from 0.

b. Using 1 as your source, construct a spanning tree for the given graph using a breadth-first process. Identify which nodes can be reached within 1, 2, 3, and 4 steps from 1.

c. Using 8 as your source, construct a spanning tree for the given graph using a breadth-first process. Identify which nodes can be reached within 1, 2, 3, and 4 steps from 8.

Problem 2. Consider the following graph



a. Using 0 as your source, construct a spanning tree for the given graph using a depth-first process.

b. Using 1 as your source, construct a spanning tree for the given graph using a depth-first process.

c. Using 8 as your source, construct a spanning tree for the given graph using a depth-first process.

Problem 3. (Bonus) Implement a breadth and/or depth first search to determine all of the vertices/nodes that can be reached starting at 0 (or general i). Return a list of visited vertices as well as a data structure to that encodes the resulting tree.