

CptS 223 Homework #4 - Graphs

Please complete the homework problems on the following page using a separate piece of paper. Note that this is an individual assignment and all work must be your own. Be sure to show your work when appropriate.

1. [13] Define these terms as they relate to graph and graph algorithms:
Use mathematical terms where appropriate.

Graph An ADT that consists of a finite set of ordered pairs, called edges or arcs, of certain entities called nodes or vertices

Vertex A node that is the fundamental unit of which graphs are formed

Edge A line segment between two nodes connecting them, one of two fundamental things in a graph along side a Vertex, An edge may be directed or undirected

Undirected Graph A graph that has bidirectional edges

Directed Graph A graph that has edges that point in a specific direction (directional edges)

Path A finite or infinite sequence of edges which connect a sequence of vertices that are all distinct from one another

Loop An edge that connects a vertex to itself

Cycle A graph that consists of a sequence of vertices starting and ending at the same vertex, with each two consecutive vertices in the sequence adjacent to each other in the graph

Acyclic _____ A graph in which there is no cycle or closed path

Connected _____ A graph in which there is a path to every vertex in the graph, all vertices are reachable, in a graph with only one vertex it is _____ considered _____ connected

Sparse _____ A Sparse graph is a graph in which the ratio of vertices to edges is very low relative to the maximum ratio of vertices to edges.

The _____ opposite _____ of _____ a _____ dense _____ graph

Weight _____ The association of a value with an edge to calculate path

lengths, _____ optimal _____ routes, _____ etc....

2. [4] Under what circumstances would we want to use an adjacency matrix instead of an adjacency list to store our graph?

- Adjacency Matrix is good for lookups and check presence or absence of a specific edge but is slow to iterate over all edges.

- Adjacency List is good fast for iterating over all edges but is slow for lookups and check presence or absence of a specific edge.

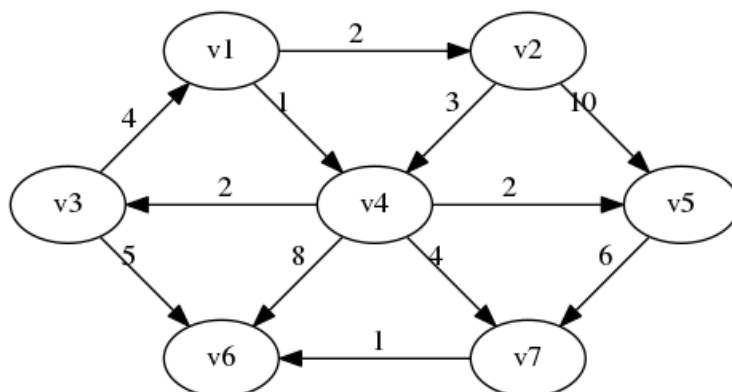
We would want an Adjacency matrix if we are doing a lot of lookups and checks for presence or absence of a specific edge, instead of using an adjacency list.

3. [6] Name three problems or situations where a graph would be a good data structure to use:

- Mapping Roads
- Studying Neural Networks
- Electrical Wiring

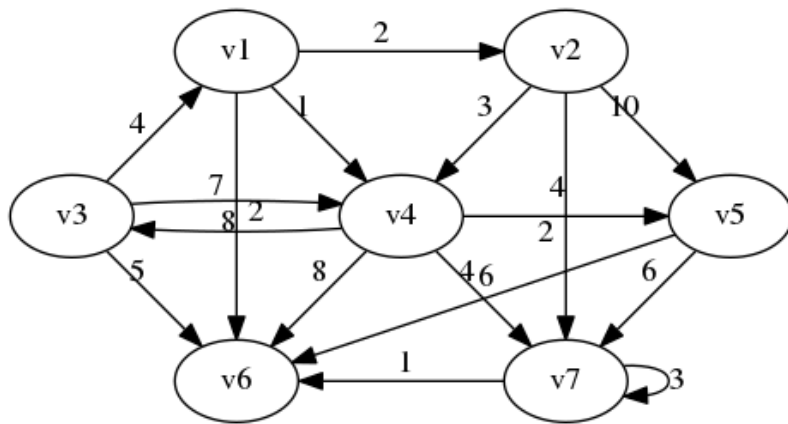
4. [4] What kind of graph is this?

A directional connected cyclic Graph



5. [4] Identify the loop in this graph:

V7->V7 and the looped edge has a weight of 3.

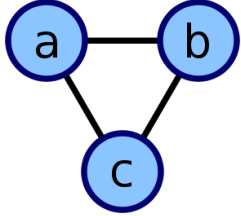
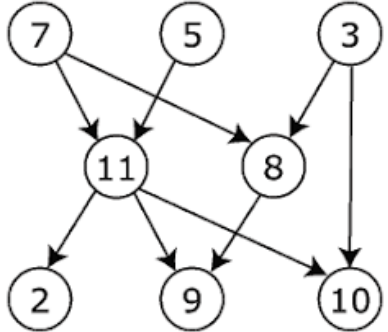


6. [4] How many vertices and edges are in this graph:

	<p>Vertices <u>7</u></p> <p>Edges <u>17</u></p>
--	---

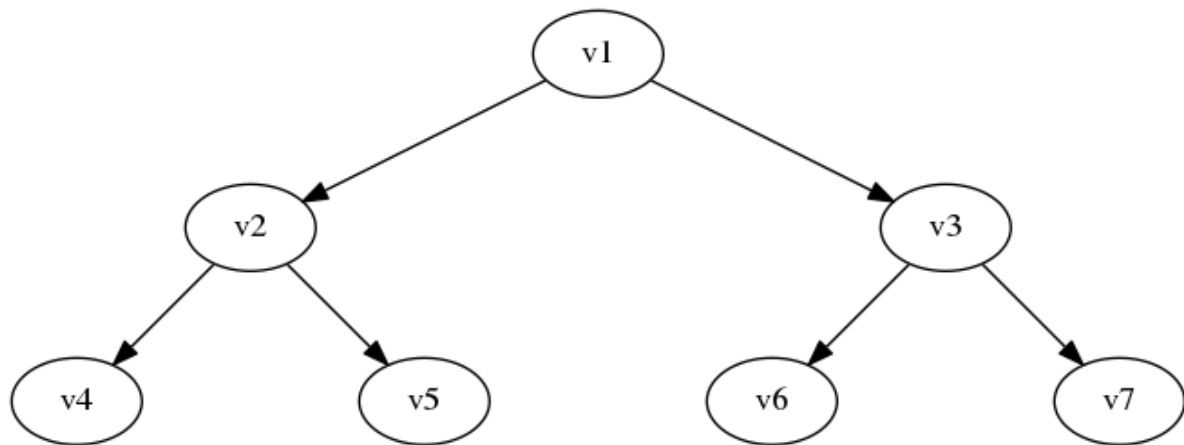
7. [6] Are these cyclic or acyclic graphs?

	<p>Cyclic?</p> <p>Yes <input type="radio"/> No <input checked="" type="radio"/></p>
--	---

	Cyclic? <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; border-radius: 50%; padding: 10px; text-align: center;">Yes</div> <div>No</div> </div>
	Cyclic? <div style="display: flex; justify-content: space-around; align-items: center;"> Yes <div style="border: 1px solid black; border-radius: 50%; padding: 10px; text-align: center;">No</div> </div>

8. [5] A tree is a particular kind of graph. What kind of graph is that?

•A tree is an undirected Graph in which any two vertices are connected by exactly one path. Any acyclic connected graph is a tree.



9. [4] What is the difference between a breadth-first search and a depth first search?

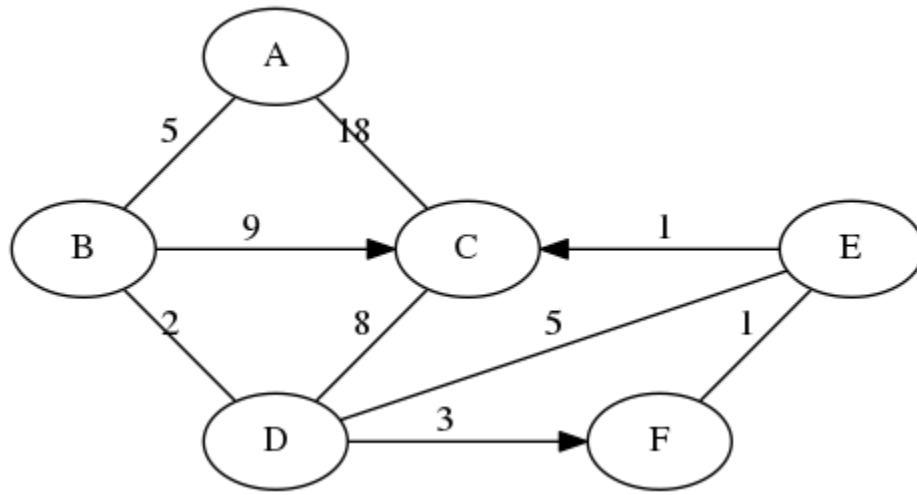
•Breadth-first search starts at the root and looks at the nodes closest to the root first. Implemented with a Queue.

•Depth-first search starts at the root and traverses all the way down as far away from the root as possible. Implemented with a Stack.

10. [10] Dijkstra's Algorithm. Use Dijkstra's Algorithm to determine the shortest path starting at **A**. Note that edges without heads are bi-directional. To save time, you do not have to add items to the "priority queue" column after it has been discovered (listed in the "distance" column). Use the table below to show your work.

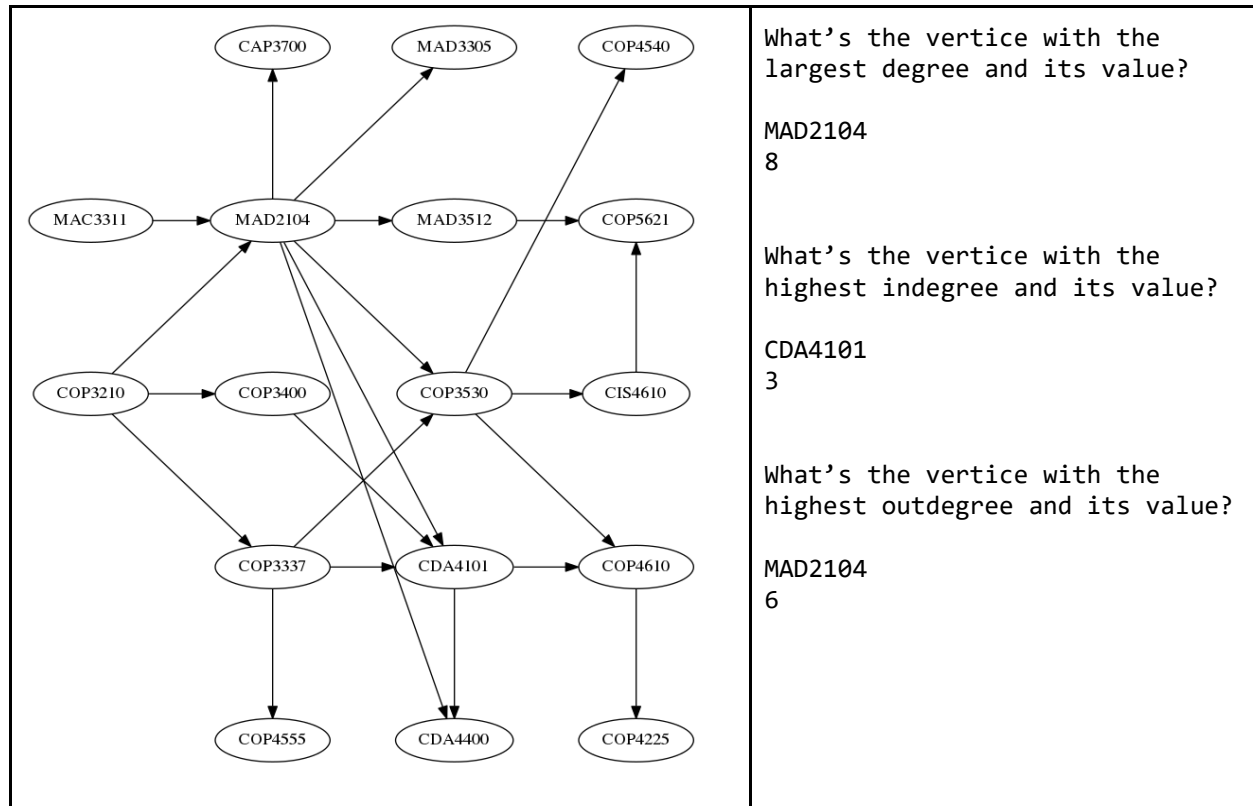
What's the shortest route (by weight) from A to C?

A->B->D->F->E->C



Node: Distance	Priority Queue
A:0	B, C
B:5	D, C
D:7	F, E, C
F:10	E, C
E:11	C
C:12	

11. [10] Topo sort. Show the final output of running Topo Sort on this graph:



Topo sort output:

MAC3311

COP3210

COP3337

COP3400

MAD2104

CAP3700

COP4555

MAD3305

MAD3512

COP3530

CDA4101

CDA4400

COP4540

CIS4610

COP4610

COP5621

COP4225