CptS 223 Homework #4 - Graphs

Due Date: Dec 11 2020 @ 11.59 pm

Please complete the homework problems using a separate document. 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

Vertice A node of a graph

Edge A line that joining two nodes

Undirected Graph A graph with bidirectional edges

Directed Graph A graph with directional edges

Path A group of edges connect vertices

Loop An edge end with same vertex

Cycle Closed Walk without repeating edge or vertex

Acyclic A graph without graph cycle

Connected A path on every pair vertices

Sparse A graph of edges is less than its possible edge

Weight Branch with labeled numbers.

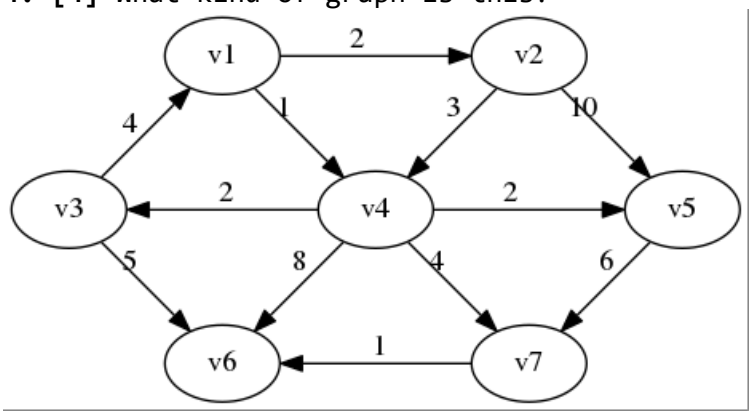
2. [4] Under what circumstances would we want to use an adjacency matrix instead of an adjacency list to store our graph?

With adjacency matrix, it performs a quick insertions and deletions of the edges. If the graph is not sparse, adjacency matrix will have a better performance compared to adjacency list.

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

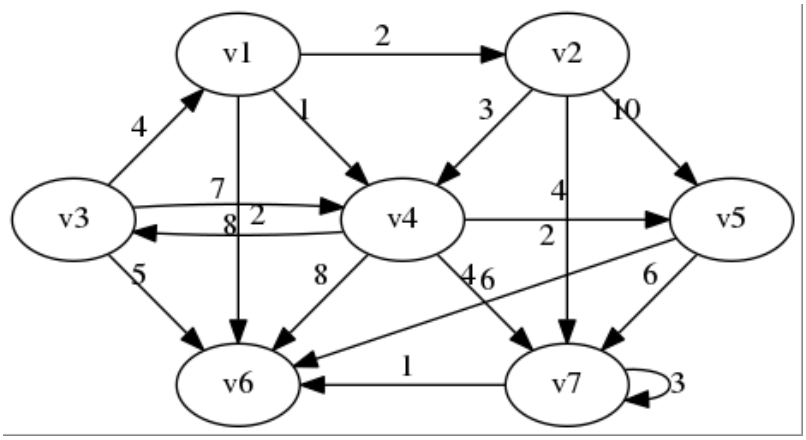
Maps, networks, Facebook's graph API

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



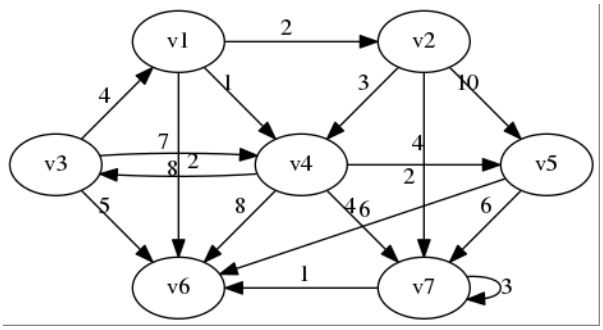
Directed Cyclic Graph

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



$V_1 - V_4 - V_3 - V_1$

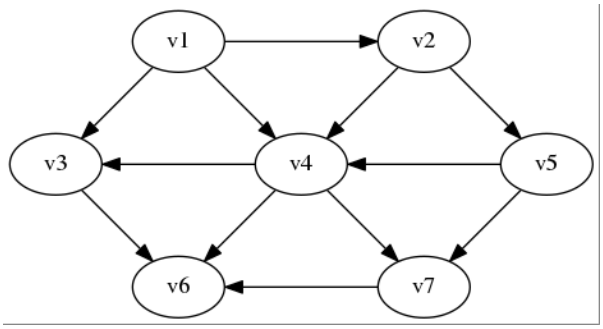
6. [4] How many vertices and edges are in this graph:



Vertices 7

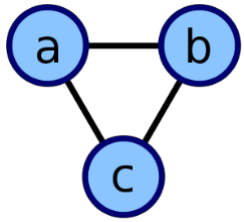
Edges 17

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



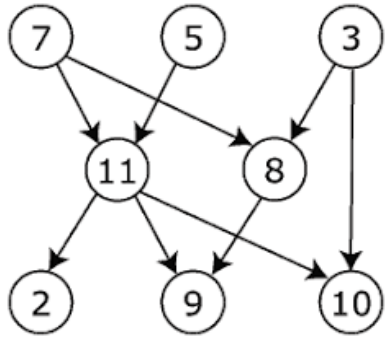
Cyclic?

Yes (No)



Cyclic?

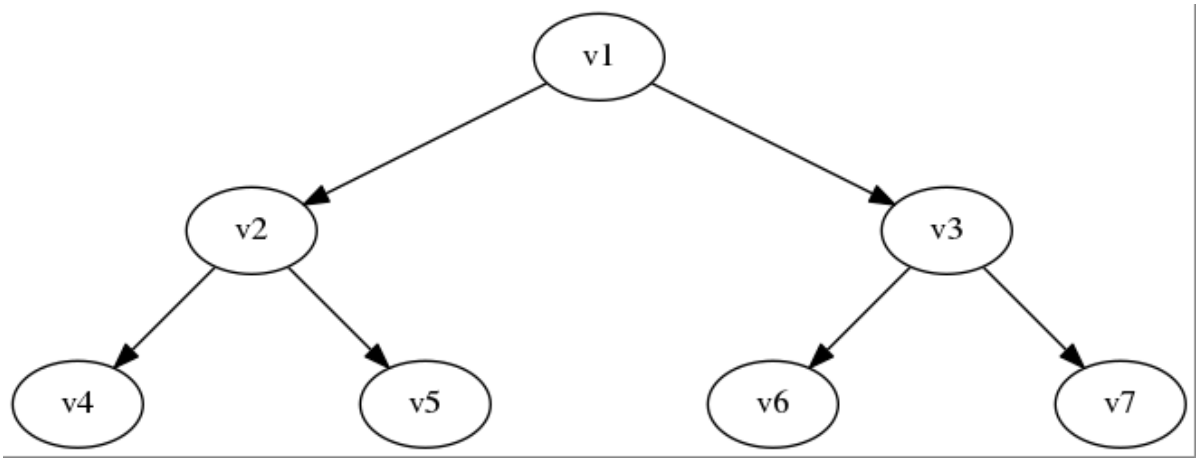
(Yes) No



Cyclic?

Yes (No)

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



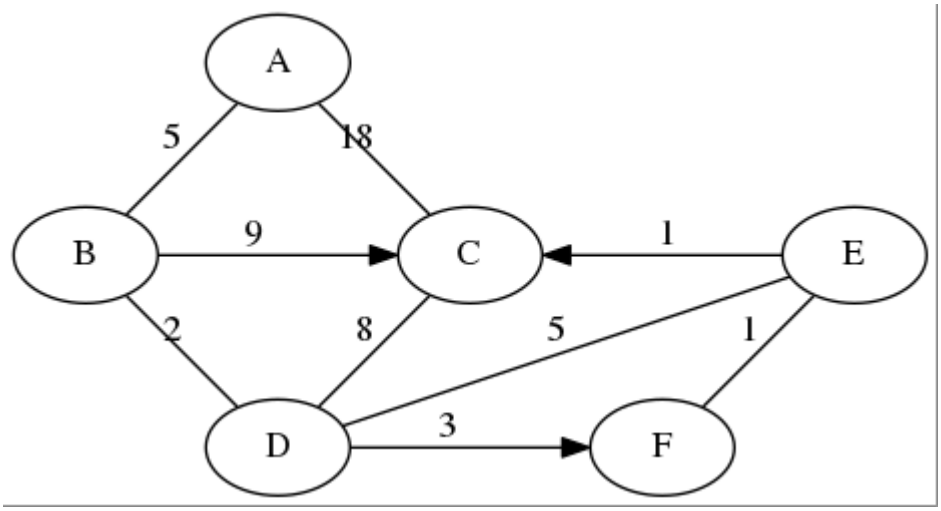
Directed, Acyclick connected graph

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

BFS uses queue data and DFS uses stack data. DFS performs quicker than BFS.

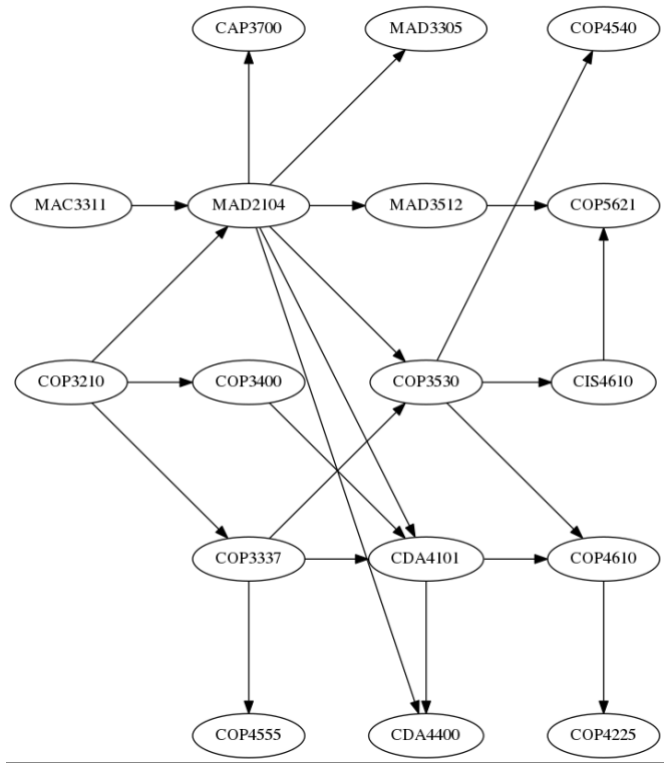
10. [10] Dijkstra's Algorithm. Use Dijkstra's Algorithm to determine the shortest path starting at **A**. Note that edges without heads are bi-directional. To save time, you do not have to add items to the "priority queue" column after it has been discovered (listed in the "distance" column). Use the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

What's the shortest route (by weight) from A to C? 12



Node: Distance	Priority Queue
5	A:B
5+2=7	B:D
7+3=10	D:F
10+1=11	F:E
11+1=12	E:C

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



What's the vertice with the largest degree and its value?

MAD2104 : 8

What's the vertice with the highest indegree and its value?

CDA4101 : 3

What's the vertice with the highest outdegree and its value?

MAD2104 : 6

Topo sort output:

MAC3311, COP3210, MAD2104, COP3400, COP3337, CAP3700, MAD3512, MAD3305, CDA4101, COP4555, COP3530, COP9540, CIS4610, COP5621, CDA4400, COP4610, COP4225