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

Graph Set of vertices and a set of edges

Vertice a point on a graph, edges don't necessarily meet

Edge line depicting length (dist) joined at a vertice

Undirected Graph objects connected together, no front/back

Directed Graph all vertices traversed in both dir.

Path sequence of vertices

Loop path from a vertice to itself

Cycle directed graph w/ at least lenght 1

Acyclic no cycles

Connected path exists from every vertex to every other vertex

Sparse $|E| << |V|$

Weight sum of edge lengths in a path

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

(dense vs. sparse)

list → $O(|E|+|V|)$

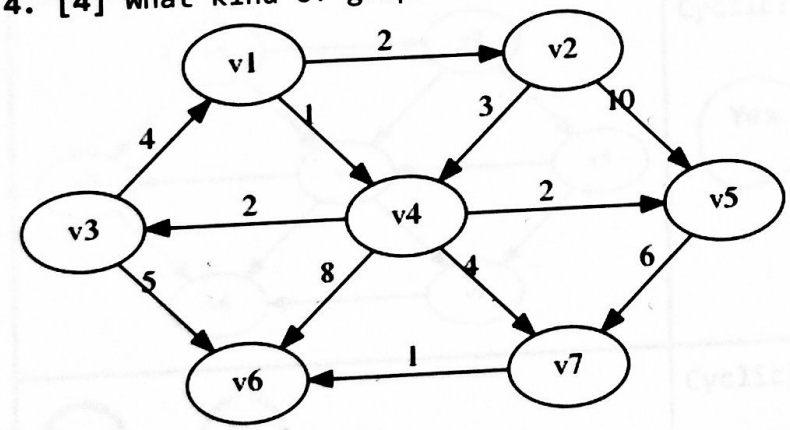
matrix → $\Theta(|V|^2)$

A[v][v] - wanting to keep track of edge weight and path weight when focusing on one vertex. → can use inf. to indicate nonexistent edges

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

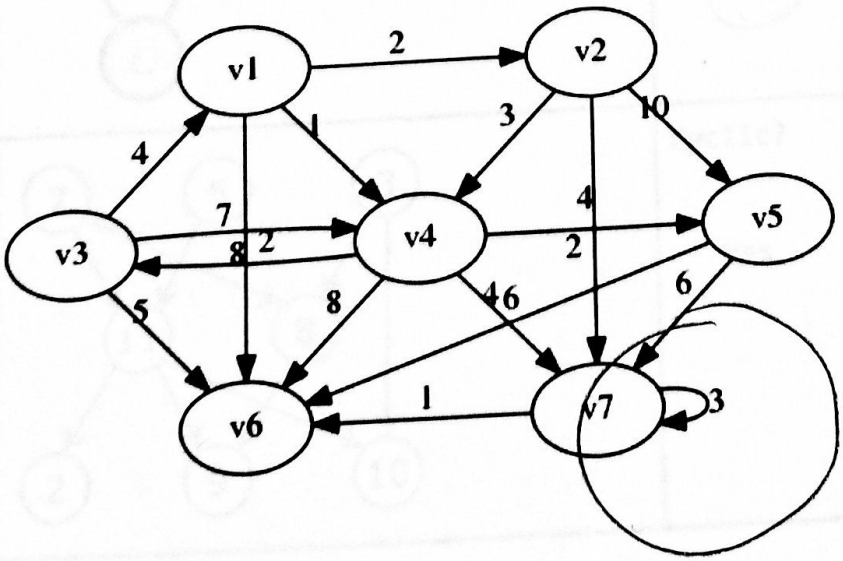
- airport connections
- traffic flow
- course prerequisites

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

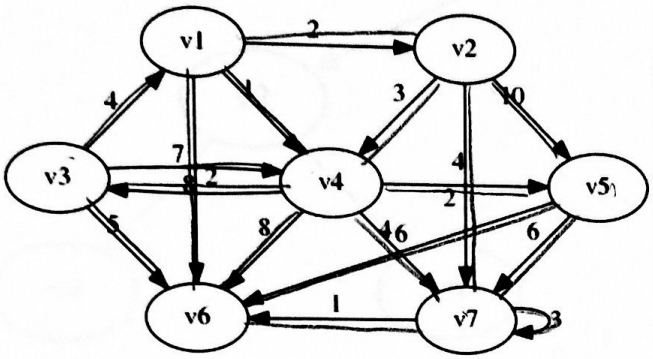


directed graph + connected

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



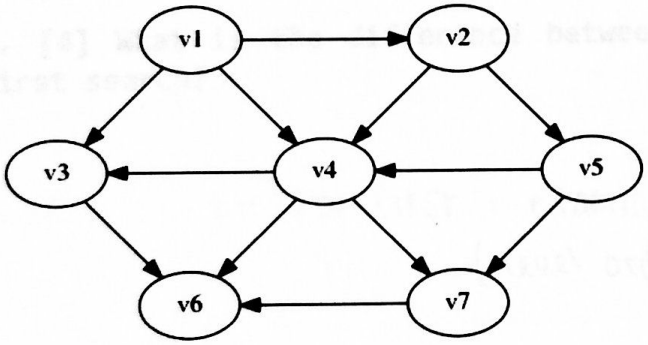
6. [4] How many vertices and edges are in this graph:



Vertices 7

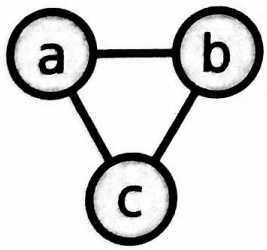
Edges 17

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



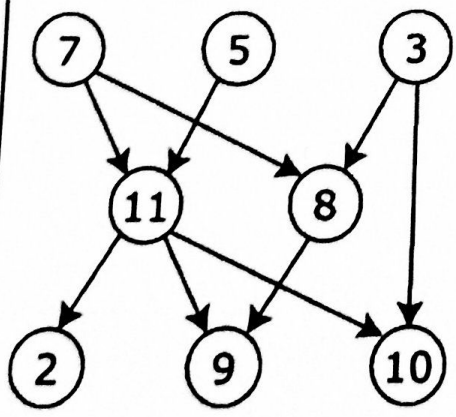
Cyclic?

Yes No



Cyclic?

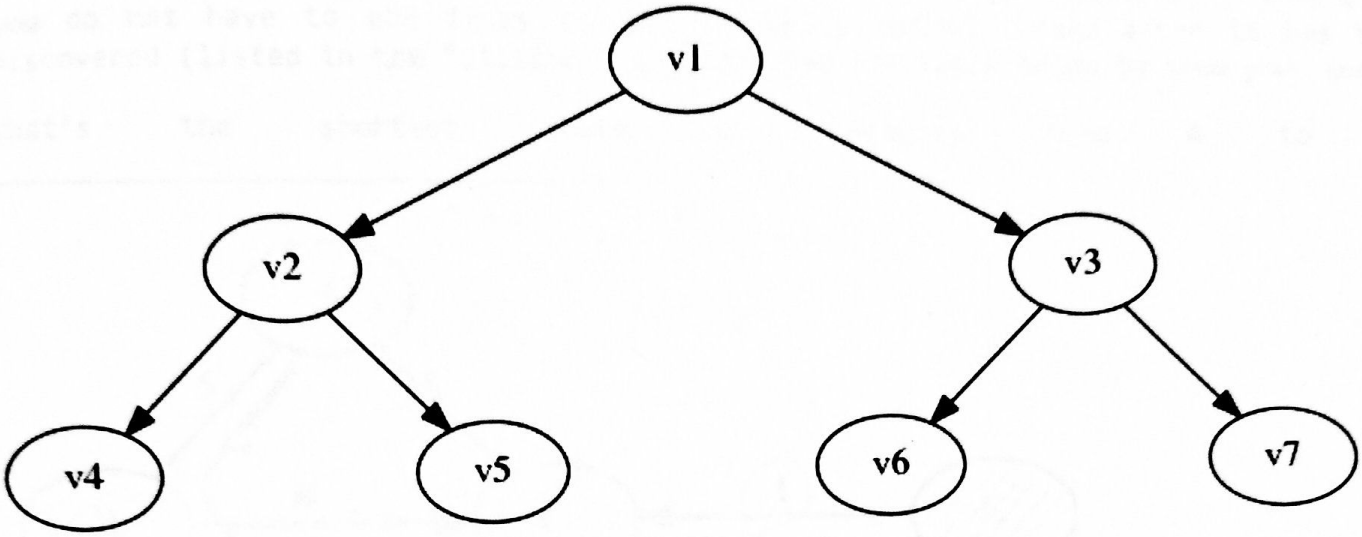
Yes No



Cyclic?

Yes No

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



Directed Acyclic graph

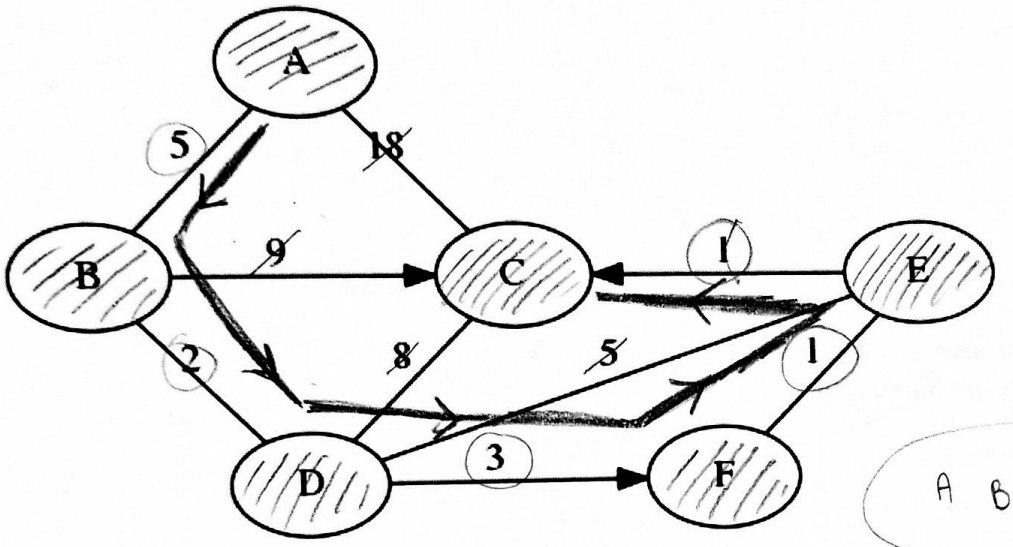
9. [4] What is the difference between a breadth-first search and a dept first search?

breadth-first : examined in layers starting at root (level order traversal)

depth-first : starts at root, goes as far down as possible on each branch before backtracking

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.

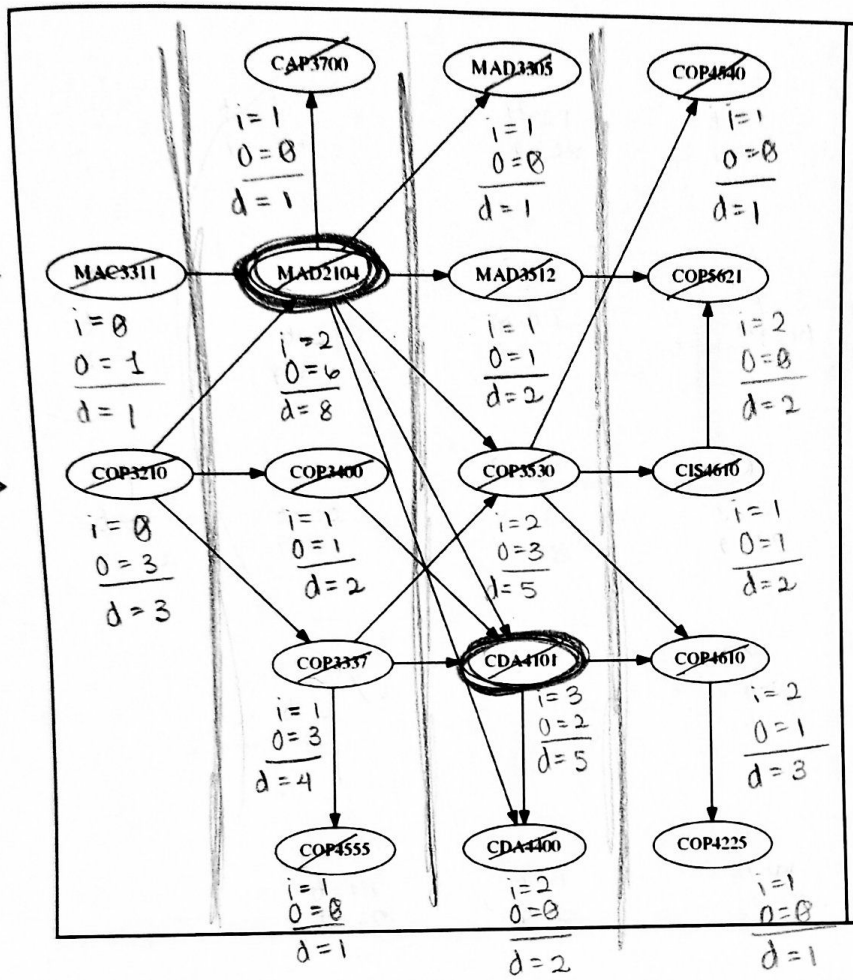
What's the shortest route (by weight) from A to C?



A B D F E C

Node: Distance	Priority Queue
0	A
5 (5)	B
7 (2)	D
10 (3)	F
11 (1)	E
12 (1)	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

d = 8

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

CDA4101

i = 3

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

MAD2104

0 = 6

Topo sort output:

MAC3311
COP3210
MAD2104
COP3400
COP3337
CAP3700
MAD3305
MAD3512
CDA4101
COP4555
COP3530
COP4540
CIS4610
COP5621
CDA4400
COP4610
COP4225