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

Graph a set of vertices & edges & the mapping function ψ_G which associates w/ each edge one or two vertices (Bondy & Murty def'n)

Vertice fundamental unit of a graph (i.e. nodes)

Edge unit which joins two vertices (a pair (v,w) where $v,w \in V(G)$)

Undirected Graph graph where edges can be traversed in either direction

Directed Graph graph where edges are "one-way" streets (meaning they have a head & a tail)

Path a sequence of vertices $w_1, w_2, \dots, w_N$ such that $(w_i, w_{i+1}) \in E$ for $i \in \{1, \dots, N-1\}$

Loop an edge whose endpoints are the same

Cycle a path (technically a "walk") of at least length 1 such that the endpoints are the same (i.e. start = finish)

Acyclic describes something (usually a graph) w/o cycles

Connected there exists a path from every vertex to every other vertex

Sparse $|E| << |V|$; there are much less edges than vertices

Weight a "cost" to traverse an edge

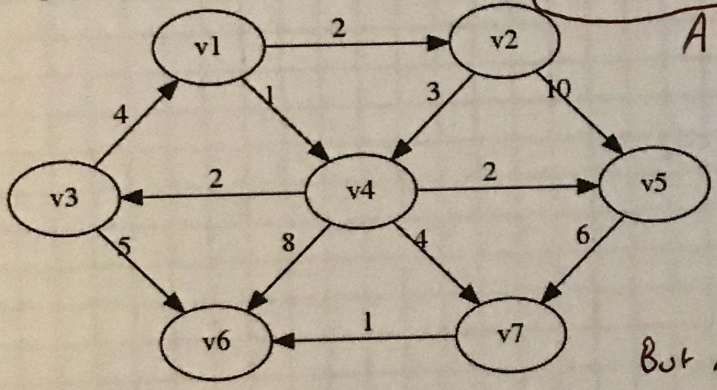
2. [4] Under what circumstances would we want to use an adjacency matrix instead of an adjacency list to store our graph?

Appropriate if the number of edges in the graph approximately equals V^2; otherwise, there are probably much less edges & we'd be wasting space since most graphs are sparse.

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

1. Street Map: nodes are destinations/locations & edges are weighted by the distance between them (via a road)
2. College Sports Ranking: nodes are teams & there exists an edge from team #1 to team #2 if team #1 beat team #2 in a game.
3. Maze: nodes are pixels or entries in a matrix & two nodes are adjacent if they're actually adjacent in the image or matrix; can solve mazes like this using A*, Dijkstra's, etc.

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

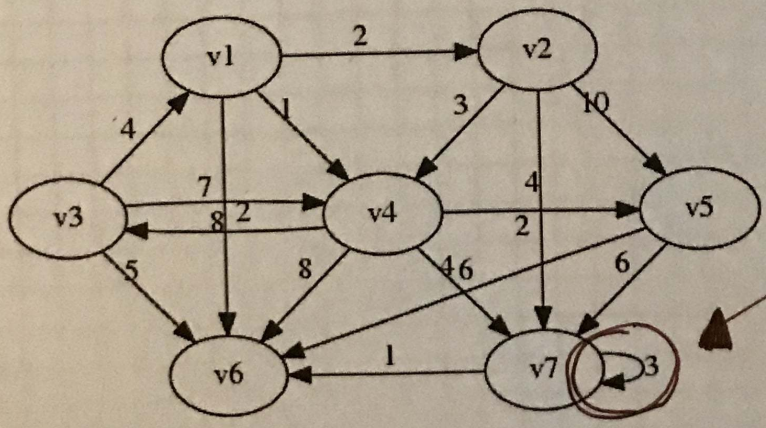


A weakly connected digraph.

Why? There is no way for nodes v5, v7, & v6 to travel to v2 (although the opposite isn't true).

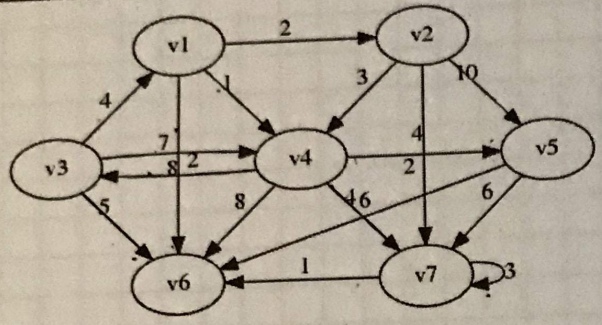
But not acyclic... $V_1 \to V_4 \to V_3 \to V_1$

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



This is the loop!

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



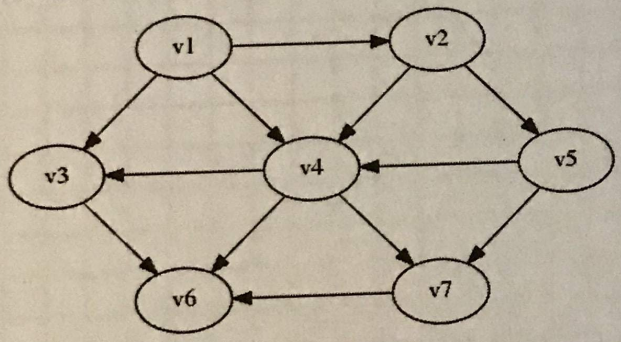
Vertices 7

Edges 17

~~total degree~~

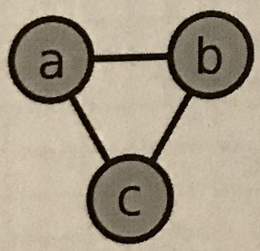
total in degree = total out degree = |E|

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



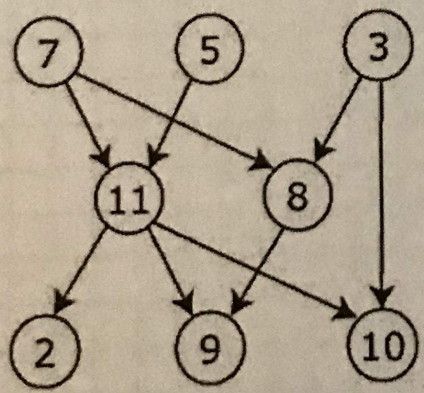
Cyclic?

Yes **No**



Cyclic?

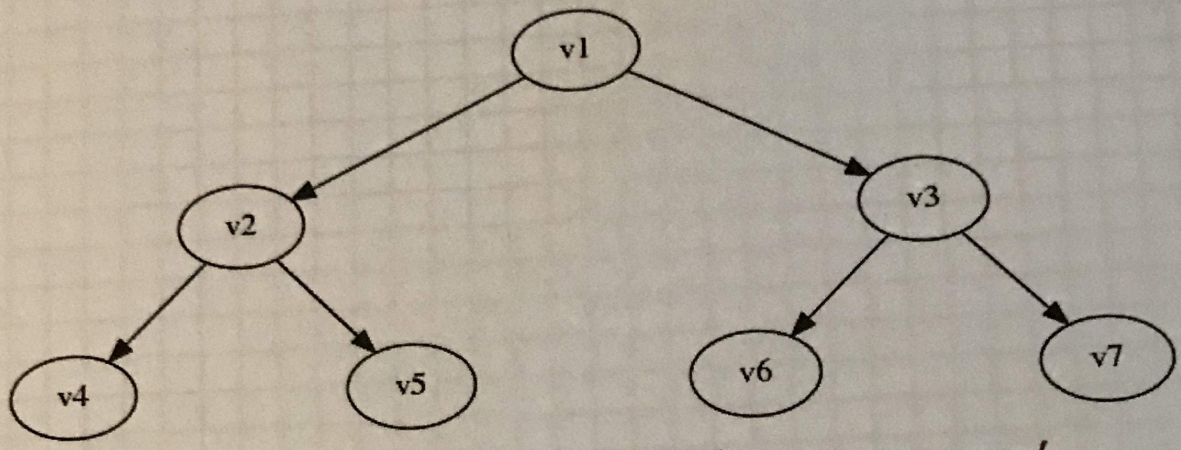
Yes No



Cyclic?

Yes **No**

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



A directed acyclic graph w/ one component.
(A forest if two or more components)

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

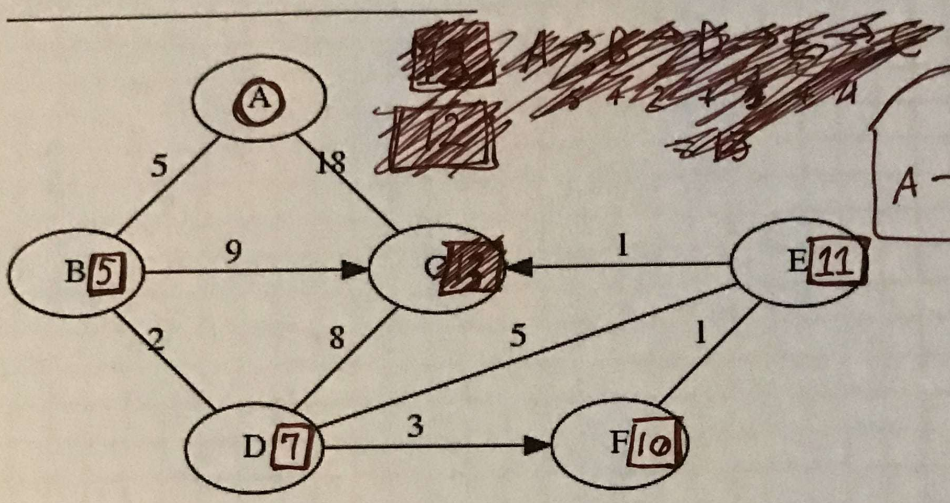
BFS explores all of the neighbors of a given node before moving on; it does this by using a queue.

DFS goes as far as it can go before checking other neighbors; in other words, we go deep (or far) from a starting node first (rather than near).

The key difference in implementation is a stack (DFS) vs. a queue (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?

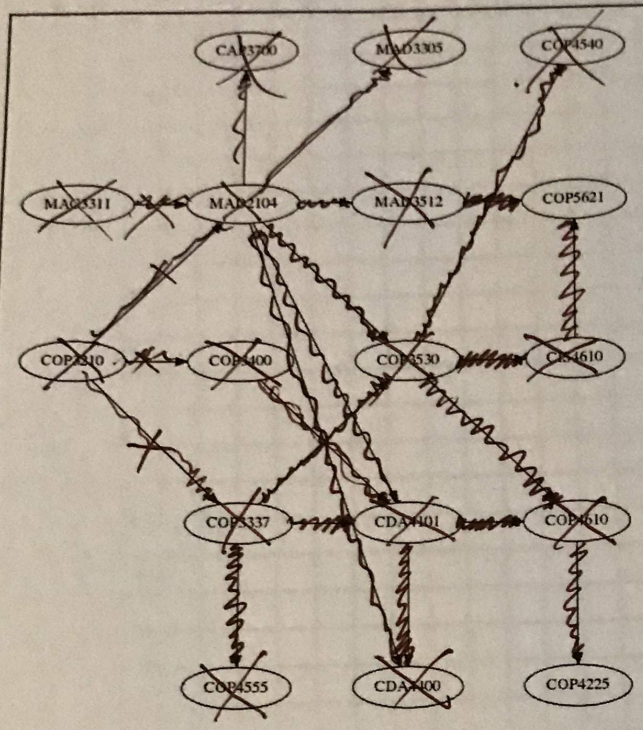


12

A → B → D → F → E → C

Node: Distance	Priority Queue
	A : 0
A : 0	B : 5 C : 18
B : 5	D : 7 C : 14 C : 18
D : 7	F : 10 E : 12 C : 14 C : 15 C : 18
F : 10	E : 11 E : 12 C : 14 C : 15 C : 18
E : 11	E : 12 C : 12 C : 14 C : 15 C : 18
[illegible]	C : 12 C : 14 C : 15 C : 18
C : [illegible] 12	C : 14 C : 15 C : 18
	dequeue the rest since C is known now

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 = 6(out) + 2 (in)

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

CDA4101

3 (in)

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

MAD2104

6 (out)

Topo sort output:

MAC3311, COP3210, ...

MAD2104, COP3400, COP3337, ...

CAP3700, MAD3305, MAD3512, COP3530, CDA4101, COP4555, ...

COP4540, C154610, COP4610, CDA4400, ...

COP5621, COP4225

Read from left to right & then go to next line at "..."