Fundamental Algorithm Techniques

Problem 1 (Graph and Tree Definitions) Prove that the following definitions are all equivalent:

We need to show that all seven definitions describe the same thing. I wil connect them step by step.

A tree is a connected acyclic graph. 1 and 7 are equivalent
1 -> 7 unique path
If the graph is connected, theres at least one path between any two vertices. If there were two different paths between the same two vertices, they would form a cycle. But the graph has no cycles, thats why this is impossible.

7 ->1 connected and acyclic
If there is a unique path between every pair of vertices, the graph is obviously connected. If it had a cycle, then any two vertices on that cycle would have two different paths between them (going around the cycle in opposite directions). This contradicts the unique path rule.The graph must also be acyclic.

1- > 4 min connected
We know a tree with V vertices has exactly V-1 edges. If we ll remove any edge (u, v) we break the unique path between u and v. Since that was the only path, u and v become disconnected. So removing any edge disconnects the graph.

4 -> 6 max acyclic
If a graph is minimally connected it has no cycles (because if it had a cycle, you could remove an edge from the cycle and it would still be connected). So it's acyclic. Now, if you add a new edge between any two vertices u and v, you create a second connection between them. Combined with the original unique path from u to v, this forms a cycle. So, adding any edge creates a cycle, meaning it's maximally acyclic.

6->5 acyclic with at least V-1 edges
A maximally acyclic graph cannot be disconnected. If it were, you could add an edge between two components without creating a cycle, which would violate the "maximally acyclic" property. So, it must be connected. A connected, acyclic graphhas exactly V-1 edges. Therefore, it certainly has *at least* V-1 edges.

5->1connected and acyclic
The graph is given as acyclic. An acyclic graph is a forest. A forest with V vertices and

c components has E = V - c edges.We are told E ≥ V - 1. So, V - c ≥ V - 1, which means -c ≥ -1or c ≤ 1.The only possibility is c = 1, meaning the graph is connected.

1->3 connected with at most V-1 edges
A connected, acyclic graph (a tree) has exactly E = V - 1 edges. So, it automatically has at most V-1 edges.

3->4 min connected
We know that for a connected graph E ≥ V - 1.Our graph is connected and has E ≤ V - 1. The only way both can be true is if E = V - 1. Now, imagine it was *not* minimally connected. Then there is some edge we could remove and the graph would stay connected. But then you would have a connected graph with only V-2 edges, which is impossible (since a connected graph must have at least V-1 edges). So, the original graph must be minimally connected.

At last, definition 2 is just saying a tree is one connected piece of a forest. Since a forest is acyclic, a “connected component of a forest" is just a connected acyclic graph,which is exactly definition1.

Conclusion:
A cycle of implications: 1 <-> 7, and 1 -> 4 -> 6 -> 5 -> 1, and 1-> 3-> 4. This connects all the definitions together. Therefore, all seven definitions are equivalent.

Problem 2 (Sparse representation of graphs):

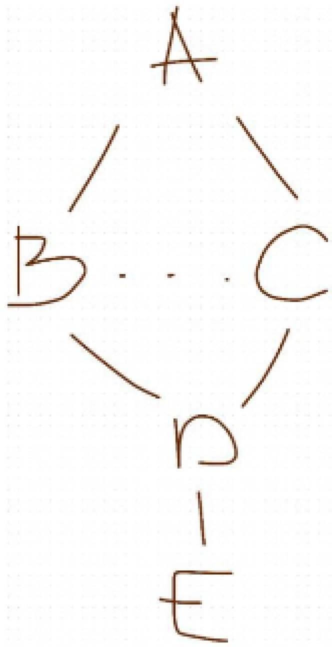
Vertices: A - >0, B - >1, C - >2, D - >3, E - >4.
CSC format is for column j, the nonzero rows are row_indices[col_pointers[j] : col_pointers[j+1]]. An entry at row r, column c means adjacency matrix entry A[r][c] = 1. For undirected graph the adjacency matrix is symmetric.

Graph 1 (undirected): columns using col_pointers ranges:

C0 (indices 0..2): rows [1,2] - >edges (1,0) and (2,0) B–A, C–A.
C1 (indices 2….5): rows [0,2,3] - > A–B, C–B, D–B.
C2 (indices 5;..8): rows [0,1,3] - > A–C, B–C, D–C.
C3 (indices 8….11): rows [1,2,4] - > B–D, C–D, E–D.
C4 (indices 11…,12): rows [3] - > D–E.

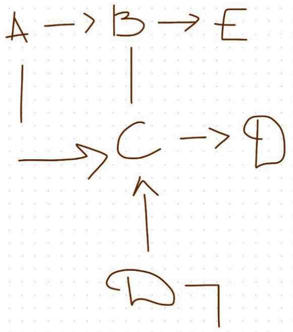
List of undirected edges (unique):
A–B, A–C, B–C, B–D, C–D, D–E.

	A	B	C	D	E
A	0	1	1	0	0
B	1	0	1	1	0
C	1	1	0	1	0
D	0	1	1	0	1
E	0	0	0	1	0



Graph 2 (directed):
C0 (0..0): no entries -> no incoming edges to A.
C1 (0..2): rows [0,3] ->edges (0,1) and (3,1 -> A ->B, D ->B.
C2 (2..4): rows [0,1] ->A -> C, B -> C.
C3 (4..5): rows [2] -> C ->D.
C4 (5..7): rows [1,3] -> B -> E, D -> E.
So directed edges:
A ->B, D ->B, A ->C, B ->C, C ->D, B ->E, D ->E.

	A	B	C	D	E	
A	0	1	1	0	0	A→B, A→C
B	0	0	1	0	1	B→C, B→E
C	0	0	0	1	0	C→D
D	0	1	0	0	1	D→B, D→E
E	0	0	0	0	0	—



Important path to see B → C → D and D → B closes a cycle

The unique directed cycle;

Directed cycle (B to C, C to D, D back to B).There are no other distinct directed cycles (the only cycle uses vertices B, C, D). So the unique simple directed cycle is:

B -> C ->D ->B.