Homework Solutions: Trees and Sparse Graph Representations

Orysbay Ulykpan

Abstract

This document contains detailed solutions to two problems: (1) proving the equivalence of seven different characterizations of trees, and (2) reconstructing graphs from their CSC (Compressed Sparse Column) representations.

1 Problem 1: Equivalent Definitions of a Tree

Let $G = (V, E)$ be a finite, simple, undirected graph with $|V| = n$ vertices and $|E|$ edges.

We are given the following seven definitions and asked to show that they are all equivalent:

1. A tree is a connected acyclic graph.
2. A tree is one component of a forest. (A forest is an acyclic graph.)
3. A tree is a connected graph with at most $V - 1$ edges.
4. A tree is a minimally connected graph; removing any edge disconnects the graph.
5. A tree is an acyclic graph with at least $V - 1$ edges.
6. A tree is a maximally acyclic graph; adding an edge between any two vertices creates a cycle.
7. A tree is a graph that contains a unique path between each pair of vertices.

We will use the following key lemma about acyclic graphs.

Lemma: Number of Edges in an Acyclic Graph

Lemma. If G is an acyclic graph with n vertices and c connected components, then

$$|E| = n - c.$$

Proof. We prove this by induction on $n = |V|$.

Base case: $n = 1$. Then G has one vertex, no edges, and one component. Thus

$$|E| = 0, \quad n - c = 1 - 1 = 0,$$

so the formula holds.

Induction step: Assume the formula holds for all acyclic graphs with fewer than n vertices. Let G be an acyclic graph with n vertices and c components.

Every finite acyclic graph has a vertex of degree 0 or 1. Let v be such a vertex.

- **Case 1:** $\deg(v) = 0$**.**

  Then v is an isolated vertex and forms its own connected component. Remove v to obtain a new graph G':

  $$|V(G')| = n - 1, \quad |E(G')| = |E(G)|, \quad \text{components}(G') = c - 1.$$

  By the induction hypothesis applied to G',

  $$|E(G')| = (n - 1) - (c - 1) = n - c.$$

  Hence $|E(G)| = |E(G')| = n - c$.

- **Case 2:** $\deg(v) = 1$**.**

  Let e be the unique edge incident to v. Remove v and e to obtain G':

  $$|V(G')| = n - 1, \quad |E(G')| = |E(G)| - 1.$$

  Removing a leaf vertex does not change the number of components, so

  $$\text{components}(G') = c.$$

  By the induction hypothesis applied to G',

  $$|E(G')| = (n - 1) - c.$$

  Thus

  $$|E(G)| - 1 = (n - 1) - c \quad \Rightarrow \quad |E(G)| = n - c.$$

In both cases $|E| = n - c$, so the lemma holds for all acyclic graphs. □

Two important corollaries of the lemma:

- If G is connected and acyclic, then $c = 1$, so $|E| = n - 1$.
- If G is acyclic and $|E| \geq n - 1$, then $n - c = |E| \geq n - 1$, hence $c \leq 1$, so $c = 1$ and $|E| = n - 1$; thus G is connected and has exactly $n - 1$ edges.

Equivalence of the Seven Definitions

We now prove that each of the definitions (2)–(7) is equivalent to (1), so all seven are equivalent.

(1) ⇔ (2)

- **(1 ⇒ 2):** A forest is by definition an acyclic graph. Each connected component of a forest is connected and acyclic, hence a tree according to (1). Thus any tree is a component of a forest.
- **(2 ⇒ 1):** If G is a component of a forest, then it is connected (by definition of a component) and acyclic (as a subgraph of an acyclic graph). Hence G is a tree by definition (1).

(1) ⇔ (3)

- **(1 ⇒ 3):** If G is a tree, it is connected and acyclic. By the lemma (with $c = 1$),

  $$|E| = n - 1,$$

  so in particular $|E| \leq n - 1$.
- **(3 ⇒ 1):** Suppose G is connected and has $|E| \leq n - 1$. Every connected graph has a spanning tree: repeatedly remove edges that lie on cycles until no cycles remain. The resulting spanning tree T has exactly $n - 1$ edges.

  Since T is a subgraph of G, we must have $|E(G)| \geq |E(T)| = n - 1$. Combining this with $|E(G)| \leq n - 1$, we get $|E(G)| = n - 1$.

  If G had any additional edge besides those in the spanning tree, then $|E(G)|$ would be at least n, and adding that edge to the tree would create a cycle. Hence G itself must be acyclic. Thus G is connected and acyclic, i.e., a tree by (1).

(1) ⇔ (4)

- **(1 ⇒ 4):** Let G be a tree, i.e., connected and acyclic. Take any edge $e = uv \in E$. In a connected graph, there is a path from u to v; one such path is just the edge e. If there were another path from u to v not using e, then together with e this would form a cycle, contradicting acyclicity.

  Therefore e lies on the unique path between u and v. Removing e destroys this path, so u and v become disconnected. Thus removing any edge disconnects G, i.e., G is minimally connected.

- **(4 ⇒ 1):** Suppose G is minimally connected, i.e., it is connected, and removing any edge disconnects it. If G contained a cycle, pick an edge e on this cycle. Removing e leaves all vertices in the cycle still connected via the remaining edges of the cycle, and all other connections remain unchanged. So the graph would remain connected after removing e, contradicting minimal connectivity. Therefore G has no cycles and is acyclic. Since G is connected and acyclic, it is a tree by (1).

(1) ⇔ (5)

- **(1 ⇒ 5):** If G is a tree, it is connected and acyclic. By the lemma, $|E| = n - 1$, so in particular G is acyclic and has at least $n - 1$ edges.

- **(5 ⇒ 1):** Assume G is acyclic and $|E| \geq n - 1$. By the lemma,

  $$|E| = n - c,$$

  where c is the number of connected components. Since $|E| \geq n - 1$, we have

  $$n - c \geq n - 1 \quad \Rightarrow \quad c \leq 1.$$

  Hence $c = 1$, so G is connected, and then $|E| = n - 1$. Therefore G is connected and acyclic, so it is a tree by (1).

(1) ⇔ (6)

- **(1 ⇒ 6):** Let G be a tree. Take any two distinct vertices u and v. Since G is connected, there is a path P between u and v. If we add an edge uv to G, then the path P together with the new edge uv forms a cycle. Thus adding any edge between two vertices creates a cycle, so G is maximally acyclic.

- **(6 ⇒ 1):** Suppose G is maximally acyclic, i.e., it is acyclic, and adding any new edge between two vertices creates a cycle. If G were disconnected, there would be vertices u and v in different components. Adding an edge uv would not create a cycle, because there was no path between u and v before. This contradicts maximal acyclicity. Hence G must be connected and acyclic, so it is a tree by (1).

(1) ⇔ (7)

- **(1 ⇒ 7):** Let G be a tree (connected and acyclic). For any two vertices u and v, connectedness implies there is at least one path between them. Assume there are two distinct simple paths between u and v. Then the union of these two paths contains a cycle (you can go from u to v using the first path and return from v to u using the second, forming a cycle), contradicting acyclicity. Therefore there is a unique path between each pair of vertices.

- **(7 ⇒ 1):** Suppose between each pair of vertices there is a unique path. Then G is connected (since there is a path between any pair). If G contained a cycle, pick two distinct vertices on this cycle. Then along the cycle we could travel between them in two different ways (clockwise and counterclockwise), producing two distinct paths, contradicting uniqueness. Hence G is acyclic and connected, i.e., a tree by (1).

Conclusion

We have shown that each of the definitions (2)–(7) is equivalent to (1). Therefore,

$$(1) \Leftrightarrow (2) \Leftrightarrow (3) \Leftrightarrow (4) \Leftrightarrow (5) \Leftrightarrow (6) \Leftrightarrow (7),$$

so all seven definitions are equivalent characterizations of a tree.

2 Problem 2: Sparse Representation of Graphs (CSC)

We are given two graphs on vertices

$$\{A, B, C, D, E\}$$

which are indexed as follows:

$$A \to 0, \quad B \to 1, \quad C \to 2, \quad D \to 3, \quad E \to 4.$$

Both graphs are stored in CSC (Compressed Sparse Column) format.

Reminder: CSC Format

A matrix (here, the adjacency matrix) is stored column-wise:

- `col_pointers` is an array of length (number of columns + 1). For each column j, the non-zero entries in that column are stored in `row_indices` between indices

$$k = \texttt{col\_pointers}[j], \ldots, \texttt{col\_pointers}[j+1] - 1.$$

- For each such k, `row_indices[k]` gives the row index i of a non-zero entry at position (i, j).
- In our case, all `values` are 1, so each such position represents an adjacency of 1.

We interpret the adjacency matrix as:

$$A[i, j] = 1 \quad \Longleftrightarrow \quad \text{there is an edge } i \to j.$$

For an undirected graph, the matrix should be symmetric.

2.1 Graph 1 (Undirected)

The CSC representation is:

```
col_pointers = [0, 2, 5, 8, 11, 12]
row_indices  = [1, 2, 0, 2, 3, 0, 1, 3, 1, 2, 4, 3]
values       = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

There are 5 columns (0 to 4), corresponding to vertices A to E.

Column-by-Column Reconstruction

For each column j:

- **Column 0 (vertex A):**

$$\text{start} = \texttt{col\_pointers}[0] = 0, \quad \text{end} = \texttt{col\_pointers}[1] = 2.$$

So we look at $k = 0, 1$:

$$\texttt{row\_indices}[0] = 1 \Rightarrow A[1, 0] = 1 \quad (B \to A),$$

$$\texttt{row\_indices}[1] = 2 \Rightarrow A[2, 0] = 1 \quad (C \to A).$$

- **Column 1 (vertex B):**

$$\text{start} = 2, \quad \text{end} = 5 \quad \Rightarrow \quad k = 2, 3, 4.$$

$$\texttt{row\_indices}[2] = 0 \Rightarrow A[0,1] = 1 \ (A \to B),$$
$$\texttt{row\_indices}[3] = 2 \Rightarrow A[2,1] = 1 \ (C \to B),$$
$$\texttt{row\_indices}[4] = 3 \Rightarrow A[3,1] = 1 \ (D \to B).$$

- **Column 2 (vertex C):**

$$\text{start} = 5, \quad \text{end} = 8 \quad \Rightarrow \quad k = 5, 6, 7.$$

$$\texttt{row\_indices}[5] = 0 \Rightarrow A[0,2] = 1 \ (A \to C),$$
$$\texttt{row\_indices}[6] = 1 \Rightarrow A[1,2] = 1 \ (B \to C),$$
$$\texttt{row\_indices}[7] = 3 \Rightarrow A[3,2] = 1 \ (D \to C).$$

- **Column 3 (vertex D):**

$$\text{start} = 8, \quad \text{end} = 11 \quad \Rightarrow \quad k = 8, 9, 10.$$

$$\texttt{row\_indices}[8] = 1 \Rightarrow A[1,3] = 1 \ (B \to D),$$
$$\texttt{row\_indices}[9] = 2 \Rightarrow A[2,3] = 1 \ (C \to D),$$
$$\texttt{row\_indices}[10] = 4 \Rightarrow A[4,3] = 1 \ (E \to D).$$

- **Column 4 (vertex E):**

$$\text{start} = 11, \quad \text{end} = 12 \quad \Rightarrow \quad k = 11.$$

$$\texttt{row\_indices}[11] = 3 \Rightarrow A[3,4] = 1 \ (D \to E).$$

Since the graph is undirected, each undirected edge appears in both directions in the adjacency matrix. Collecting unique undirected edges, we have:

$$A\text{–}B, \quad A\text{–}C, \quad B\text{–}C, \quad B\text{–}D, \quad C\text{–}D, \quad D\text{–}E.$$

(a) Adjacency Matrix of Graph 1

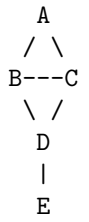
Use the vertex order $[A, B, C, D, E] = [0, 1, 2, 3, 4]$. The adjacency matrix is:

$$A_1 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

which is symmetric, as expected for an undirected graph.

(b) Diagram of Graph 1

A natural layout for the graph is:



Explanation:

- Vertices A, B, C form a triangle.
- Vertex D is connected to both B and C.
- Vertex E is a leaf attached to D.

2.2 Graph 2 (Directed)

The CSC representation is:

```
col_pointers = [0, 0, 2, 4, 5, 7]
row_indices  = [0, 3, 0, 1, 2, 1, 3]
values       = [1, 1, 1, 1, 1, 1, 1]
```

Again, we have columns 0 to 4 corresponding to vertices A to E.

Column-by-Column Reconstruction

- **Column 0 (vertex A):**

$$\text{start} = \texttt{col\_pointers}[0] = 0, \quad \text{end} = \texttt{col\_pointers}[1] = 0.$$

  No entries $\Rightarrow$ no edges into A.

- **Column 1 (vertex B):**

$$\text{start} = 0, \quad \text{end} = 2 \quad \Rightarrow \quad k = 0, 1.$$
$$\texttt{row\_indices}[0] = 0 \Rightarrow A[0,1] = 1 \ (A \to B),$$
$$\texttt{row\_indices}[1] = 3 \Rightarrow A[3,1] = 1 \ (D \to B).$$

- **Column 2 (vertex C):**

$$\text{start} = 2, \quad \text{end} = 4 \quad \Rightarrow \quad k = 2, 3.$$
$$\texttt{row\_indices}[2] = 0 \Rightarrow A[0,2] = 1 \ (A \to C),$$
$$\texttt{row\_indices}[3] = 1 \Rightarrow A[1,2] = 1 \ (B \to C).$$

- **Column 3 (vertex D):**

$$\text{start} = 4, \quad \text{end} = 5 \quad \Rightarrow \quad k = 4.$$
$$\texttt{row\_indices}[4] = 2 \Rightarrow A[2,3] = 1 \ (C \to D).$$

- **Column 4 (vertex E):**

$$\text{start} = 5, \quad \text{end} = 7 \quad \Rightarrow \quad k = 5, 6.$$
$$\texttt{row\_indices}[5] = 1 \Rightarrow A[1,4] = 1 \ (B \to E),$$
$$\texttt{row\_indices}[6] = 3 \Rightarrow A[3,4] = 1 \ (D \to E).$$

Thus the directed edges in the graph are:

$$A \to B, \quad A \to C, \quad B \to C, \quad C \to D, \quad D \to B, \quad B \to E, \quad D \to E.$$

(a) Adjacency Matrix of Graph 2

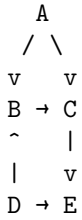
Using the vertex order $[A, B, C, D, E] = [0, 1, 2, 3, 4]$, the adjacency matrix is:

$$A_2 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Row i corresponds to outgoing edges from vertex i.

(b) Diagram of Graph 2

A reasonable layout is:



More explicitly:

- A has edges to B and C: $A \to B$, $A \to C$.
- B has edges to C and E: $B \to C$, $B \to E$.
- C has an edge to D: $C \to D$.
- D has edges to B and E: $D \to B$, $D \to E$.
- E has no outgoing edges (a sink).

(c) Unique Cycle in the Directed Graph

We inspect the subgraph on vertices $\{B, C, D\}$:

$$B \to C, \quad C \to D, \quad D \to B.$$

These three edges form a directed cycle:

$$B \to C \to D \to B.$$

There are no other cycles:

- A has no incoming edges, so no cycle can include A.
- E has no outgoing edges, so no cycle can pass through E and return.
- Thus any cycle must lie entirely among $\{B, C, D\}$, and $B \to C \to D \to B$ is the unique directed cycle.