Determining whether an undirected graph is a tree involves looking at two primary conditions: connectedness and acyclicity.

1. \*\*Interconnectivity:\*\* A tree is a linked graph, which means that every two nodes in the graph have a path connecting them. Start from any node and conduct a depth-first search (DFS) or breadth-first search (BFS) to verify connectivity. The graph is connected if every node is visited throughout the traversal process.

2. \*\*Acyclicity:\*\* A tree lacks cycles and is hence an acyclic graph. You can retain a visited set of nodes and do a depth-first search (DFS) to verify acyclicity. A cycle occurs during the DFS if you come across a node that has previously been visited and is not the parent of the present node.  
  
A graph is considered to be a tree if it meets both the requirements of connectedness and acyclicity.

Running time:

DFS or BFS for Connectivity: You can check connectivity using either DFS or BFS. The number of vertices, or nodes, in the graph is V, and the number of edges is E. The running time is therefore O(V + E).

DFS for Acyclicity: DFS is used to verify that acyclicity has an O(V + E) running time.

O(V + E) would be the total running time for the acyclicity and connection checks. As a result, the algorithm's time complexity is linear in relation to the graph's vertex and edge counts.