Graph Searching Algorithm

An undirected graph is a tree if and only if it is acyclic (contains no cycles) and connected. Therefore, to determine whether an undirected graph is a tree, you can perform the following checks:

Cycles: Use any graph traversal algorithm, such as Depth-First Search (DFS) or Breadth-First Search (BFS), to traverse the graph. If, during the traversal, you encounter a node that has already been visited and is not the parent of the current node, then a cycle exists, and the graph is not a tree.

Connectivity: Ensure that all nodes in the graph are visited during the traversal. If any node remains unvisited, the graph is not connected, and therefore not a tree.

The running time of this approach depends on the graph traversal algorithm used. Both DFS and BFS have a time complexity of O(V + E), where V is the number of vertices (nodes) and E is the number of edges in the graph. Therefore, the overall running time is O(V + E).

In summary, to determine whether an undirected graph is a tree, perform a graph traversal to check for cycles and ensure connectivity. The time complexity is linear in the number of vertices and edges.