Problem Pattern

1. When do you think a cycle would occur in an undirected graph? Do you think revisiting a node which is already visited is a definitive indication of a cycle?
2. Do you think 1 --- 2 --- 3 has any cycle ? 2 has 1 and 3 in its adjacency list but 1 is already visited(since traversal starts from 1) . Try handling such cases by keeping in mind the parent node exists in the adjacency list as well and handling the case.
3. Do you think the entire graph is connected? Check for the cycle in all the components in the graph and not just the first component.

Problem Approach

1. Use the edge list to generate an adjacency list for the graph. Also generate a visited list marking all nodes of the graph as False (indicating not visited).
2. Start a graph traversal(DFS or BFS) from the first unvisited point. Mark the current node as visited.
3. Traverse to the connected nodes and mark them as visited.
4. Continue the traversal till there is no more node left to visit or the destination has been visited. In case you encounter an already visited node which is not the parent of the current node, return True indicating that a cycle has been detected. The traversal ends when a cycle is detected or there are no more unvisited connected vertices left.
5. When the traversal ends, if a cycle is detected return True. Otherwise, return to step 3 and check for any unvisited nodes left in the graph and a cycle in that component (i.e. continue step 3 to 5).
6. When all the nodes are visited, if no cycle exists return False.