Clone Graph

For this problem the task is to create a deep copy of an undirected graph using DFS or BFS.

Key Idea:

- Use a hash map (or unordered_map in C++) to map original nodes to their clones.
- Traverse the graph (DFS or BFS).
- For each node:
    1. If it's already cloned, return the clone.
    2. Otherwise, create a new node, store it in the map and recursively clone its neighbors.

This prevents infinite loops in cycles and ensures each node is cloned exactly once.

Algorithm (DFS):

1. Base case: if input node is nullptr, return nullptr.
2. Create a hash map unordered_map<Node*, Node*> cloned.
3. Define a recursive function:
    - If node is already in cloned, return it.
    - Otherwise, create a copy of the node.
    - Recursively clone all neighbors and add them to the new node's neighbor list.
4. Return the clone of the starting node.

Complexity:

- Time: $O(V+E)$ - each node and edge is visited once during DFS.
- Space: $O(V)$ - due to hash map and recursion stack (worst case).