Clone Graph

Given a reference of a node in a connected undirected graph.
Return a deep copy (clone) of the graph.
Each node in the graph contains a value (int) and a list (List[Node]) of its neighbors.

```
class Node {
public int val;
public List<Node> neighbors;
}
```

Input: adjList = [[2,4],[1,3],[2,4],[1,3]]
Output: [[2,4],[1,3],[2,4],[1,3]]
Explanation: There are 4 nodes in the graph.
1st node (val = 1)'s neighbors are 2nd node (val = 2) and 4th node (val = 4).
2nd node (val = 2)'s neighbors are 1st node (val = 1) and 3rd node (val = 3).
3rd node (val = 3)'s neighbors are 2nd node (val = 2) and 4th node (val = 4).
4th node (val = 4)'s neighbors are 1st node (val = 1) and 3rd node (val = 3).

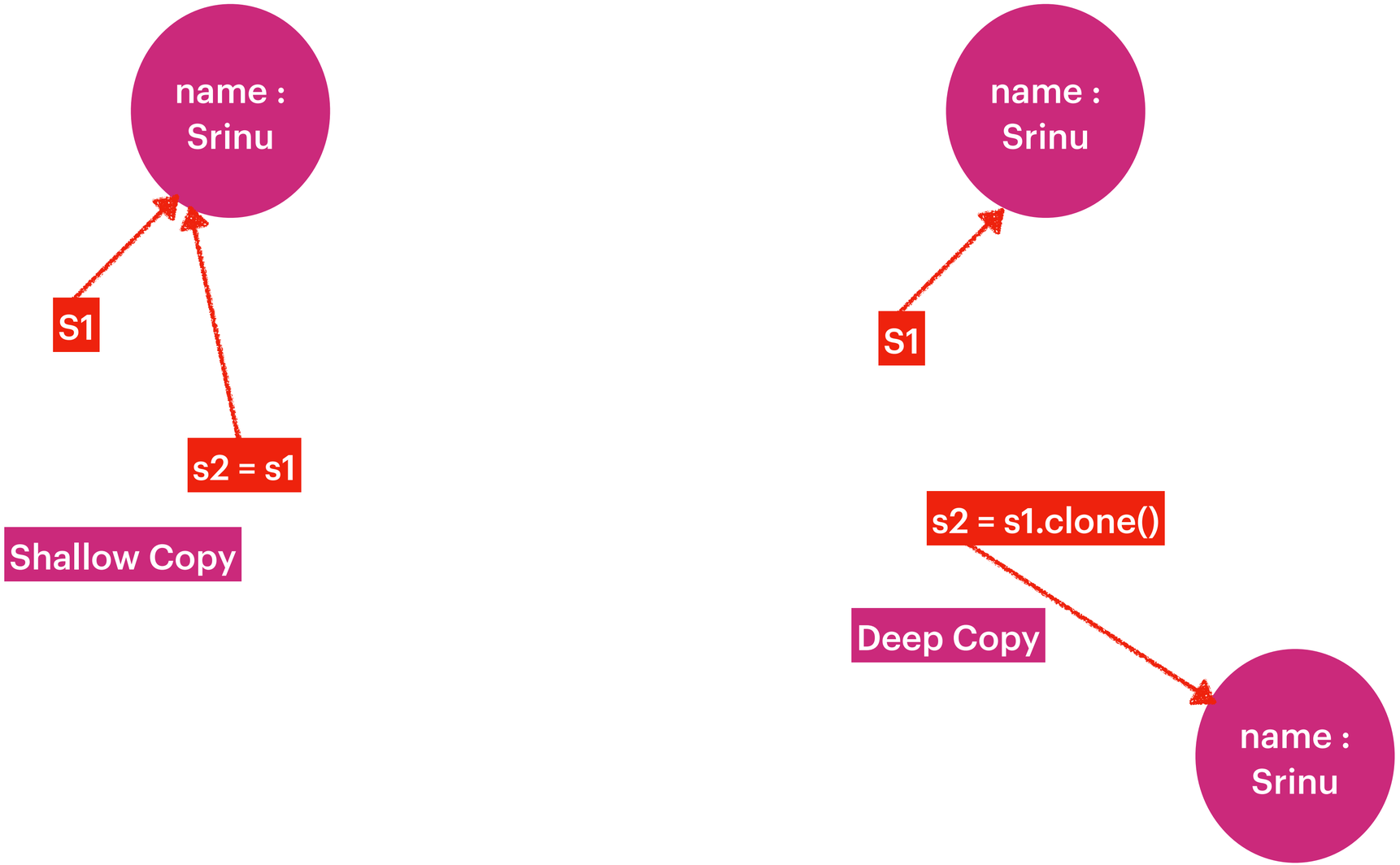
Constraints:
The number of nodes in the graph is in the range `[0, 100]`.
1 <= Node.val <= 100
`Node.val` is unique for each node.
There are no repeated edges
and no self-loops in the graph.
The Graph is connected and all nodes can
be visited starting from the given node.

Input: adjList = [[2],[1]]
Output: [[2],[1]]

```
class cloneGraph {
public Node cloneGraph(Node node) {
}

}
```

Input: adjList = [[]]
Output: [[]]
Explanation: Note that the input contains one empty list. The graph consists of only one node with val = 1 and it does not have any neighbors.


name :
Srinu
S1
s2 = s1
Shallow Copy
name :
Srinu
S1
s2 = s1.clone()
Deep Copy
name :
Srinu

Input: adjList = [[2,4],[1,3],[2,4],[1,3]]
Output: [[2,4],[1,3],[2,4],[1,3]]
Explanation: There are 4 nodes in the graph.
1st node (val = 1)'s neighbors are 2nd node (val = 2) and 4th node (val = 4).
2nd node (val = 2)'s neighbors are 1st node (val = 1) and 3rd node (val = 3).
3rd node (val = 3)'s neighbors are 2nd node (val = 2) and 4th node (val = 4).
4th node (val = 4)'s neighbors are 1st node (val = 1) and 3rd node (val = 3).