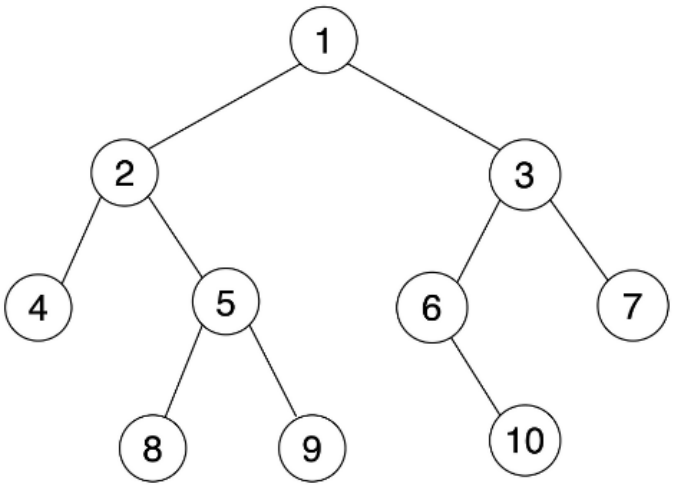
1. **Given a matrix of dimension m*n where each cell in the matrix can have values 0, 1 or 2 which has the following meaning:**
   0: Empty cell
   1: Cells have healthy person
   2: Cells have infected person
   Determine what is the minimum time required so that every person is infected. An infected person at index [i, j] can infect other healthy person at indexes [i-1,j], [i+1,j], [i,j-1], [i,j+1] (up, down, left and right). If it is impossible to infect every person then simply return -1.

   Test Cases:
   1. [[2,1,1], [1,1,0], [0,1,1]] output = 4
   2. [[2,1,1], [0,1,1], [1,0,1]] output = 1
   3. [[0,2]] output = 0

2. **Sort Linked List**
   i/p: -2 -> 10 -> 7 -> 5 -> -1 -> 2

   Constraint: TC= O(NlogN), SC = O(1)

3. **Create following Binary Tree from scratch. And print it using levelOrderTraversal.**



4. **Perform Inorder Traversal on above BTree.**
   Constraint: SC = O(1)
   Output = 4 2 8 5 9 1 6 10 3 7.

5. **Find LCA of node 8 and node 7 in above BTree.**
   Output = 1

6. **Convert the above BTree into LL. (Flatten the BTree).**