[**Serialize and Deserialize Binary Tree**](https://leetcode.com/problems/serialize-and-deserialize-binary-tree/)

**package** leetcodeHard;

**import** java.util.LinkedList;

**import** java.util.Queue;

**public** **class** SerializeDeserialize {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

TreeNode root = **new** TreeNode(10);

root.left = **new** TreeNode(5);

root.left.left = **new** TreeNode(3);

root.left.right = **new** TreeNode(7);

root.right = **new** TreeNode(15);

root.right.right = **new** TreeNode(18);

String data = *serialize*(root);

System.***out***.println(data);

TreeNode node = *deserialize*(data);

System.***out***.println(node.val);

}

// Encodes a tree to a single string.

**public** **static** String serialize(TreeNode root) {

**if**(root == **null**) {

**return** "";

}

StringBuilder str = **new** StringBuilder();

*serializeHelper*(root , str);

**return** str.toString();

}

**public** **static** **void** serializeHelper(TreeNode node , StringBuilder str) {

**if**(node == **null**) {

str.append("X").append(";");

**return**;

}

str.append(node.val).append(";");

*serializeHelper*(node.left , str);

*serializeHelper*(node.right, str);

}

// Decodes your encoded data to tree.

**public** **static** TreeNode deserialize(String data) {

**if**(data == **null** || data.length() == 0)

**return** **null**;

Queue<String> queue = **new** LinkedList<>();

**for**(String token : data.split(";")) {

queue.offer(token);

}

**return** *deserializeHelper*(queue);

}

**public** **static** TreeNode deserializeHelper(Queue<String> queue) {

String temp = queue.poll();

**if**(temp.equals("X")) {

**return** **null**;

}

TreeNode node = **new** TreeNode(Integer.*parseInt*(temp));

node.left = *deserializeHelper*(queue);

node.right = *deserializeHelper*(queue);

**return** node;

}

}

**class** TreeNode {

**int** val;

TreeNode left;

TreeNode right;

TreeNode(**int** x) { val = x; }

}

**Serialize:**

Time Complexity : O(N where N is number of nodes in Tree

Space Complexity : O(N) where N is number of nodes in Tree

**Deserialize:**

Time Complexity : O(N where N is length of the given serialized tree

Space Complexity : O(N) where N is length of the given serialized tree