**Serialize and Deserialize Binary Tree**

string serializeTree(TreeNode<int>\* root) {

if (!root) {

return "#";

}

string result = to\_string(root->data) + ",";

string left = serializeTree(root->left);

string right = serializeTree(root->right);

return result + left + right;

}

TreeNode<int>\* deserializeTreeHelper(string& serialized, int& index) {

if (index >= serialized.length() || serialized[index] == '#') {

index++;

return nullptr;

}

int num = 0;

while (index < serialized.length() && serialized[index] >= '0' && serialized[index] <= '9') {

num = (num \* 10) + (serialized[index] - '0');

index++;

}

TreeNode<int>\* root = new TreeNode<int>(num);

index++;

root->left = deserializeTreeHelper(serialized, index);

root->right = deserializeTreeHelper(serialized, index);

return root;

}

TreeNode<int>\* deserializeTree(string& serialized) {

int index = 0;

return deserializeTreeHelper(serialized, index);

}