**Tree Hard Questions**

## **297. Serialize and Deserialize Binary Tree**

* Serialization is the process of converting a data structure or object into a sequence of bits so that it can be stored in a file or memory buffer, or transmitted across a network connection link to be reconstructed later in the same or another computer environment.
* Design an algorithm to serialize and deserialize a binary tree. There is no restriction on how your serialization/deserialization algorithm should work. You just need to ensure that a binary tree can be serialized to a string and this string can be deserialized to the original tree structure.

class Codec {

public:

// Encodes a tree to a single string.

string serialize(TreeNode\* root) {

string s = "";

queue<TreeNode \*> q;

q.push(root);

while(!q.empty()) {

TreeNode \*node = q.front();

q.pop();

if(!node) s += "n";

else {

s += to\_string(node->val);

q.push(node->left);

q.push(node->right);

}

if(!q.empty()) s += ',';

}

return s;

}

// Decodes your encoded data to tree.

TreeNode\* deserialize(string data) {

char del = ',';

stringstream ss(data);

string word = "";

queue<TreeNode \*> q;

TreeNode \*root = NULL;

while(!ss.eof()) {

if (!root) {

getline(ss, word, del);

if(word != "n") {

root = new TreeNode(stoi(word));

q.push(root);

q.push(root);

}

}

else{

getline(ss, word, del);

TreeNode \*node1 = q.front();

q.pop();

if(word != "n") {

TreeNode \*n = new TreeNode(stoi(word));

node1->left = n;

q.push(n);

q.push(n);

}

getline(ss, word, del);

TreeNode \*node2 = q.front();

q.pop();

if(word != "n") {

TreeNode \*n = new TreeNode(stoi(word));

node2->right = n;

q.push(n);

q.push(n);

}

}

}

return root;

}

};

**New Concept Learned**

How to Split String in C++:

stringstream ss(str);

string word = “”;

char del = ‘|’;

while(!ss.eof()) {

getline(ss, word, del);

}