1. **Binary Tree Maximum Path Sum**

**Given a binary tree, find the maximum path sum**

**The path may start and end at leaf node in the tree**

o

1/ 2\

o o

* Minimum path will be 1

Struct Node {

Vector <Edges \*> edges;

};

Struct Edge {

Int weight;

Node \*next;

}

Int MinimumPath(Node \*root) {

Int minium\_result = INT\_MAX;

MinimumPath(root, 0, minimum\_result);

Return minimum\_result;

}

Void MinimumPath(Node \*root, int cost, int &minimum\_result) {

If (0 == root->edges.size()) {

If (cost < minimum\_result) {

Minimum\_reuslt = cost;

}

Return ;

}

For (int I = 0; I < root->edges.size(); ++ I) {

MinimumPath(root->edges[i], cost + root->edges[i].weight, minimum\_result);

}

}

1. **Follow the interviewer’s instruction to validate the python code** 
   1. **The consideration on the problems is based on the requirements of interviewers**
   2. **Using stack to solve the problems**

I can’t clearly remember the detail. I tried my best to finish it.

bool input(vector <line>& inputs) {

For (int I = 0; I < inputs.size(); ++ i) {

if (stack.empty()) {

stack.push(line);

} else {

If (IsConstrolStatements(stack.top()) {

Line.indent\_len < stack.top.indent\_len;

Return false;

} else {

If (line.indent\_len == stack.top.indent\_len) {

Stack.push(line);

} else {

While (!stack.empty() && line.indent\_len != stack.top.indent\_len) {

Stack.pop();

If (stack.top().line != line.indent\_line) return false;

}

}

}

}

}

Return true;

}a