// 0\_treeNode查看-----------------------------

        // 队列的方式

        function findNodeById(root, id) {

            const quene = [root];

            do{

                const first = quene.shift();

                if(first.id === id) {

                    return first;

                }

                if(first.children) {

                    quene.push(...first.children);

                }

            }while(quene.length)

        }

// 1\_学生按成绩分组-------------------------------------

        let students = [

            {name: '张三', score: 84},

            {name: '李四', score: 58},

            {name: '王五', score: 99},

            {name: '赵六', score: 69},

        ]

        function getGrade(score) {

            return score < 60 ? 'C' :

                   score < 80 ? 'B' : 'A';

        }

        function groupBy(students) {

            var len = students.length,

                group = {};

            for(var i = 0; i < len; i++) {

                if( !group[getGrade(students[i].score)]  ) {

                    group[getGrade(students[i].score)] = [];

                }

                group[getGrade(students[i].score)].unshift  (students[i]);

            }

            return group;

        }

// 2\_字符串Parse------------------------------------

        ini ==> {type: 'int'}

        Array<bool>   ==> {type: 'Array', typeArgs: [{type: 'bool'}]}

        Array<Array<string>>  ==> {type: 'Array', typeArgs: [{type: 'Array', typeArgs: [{type: 'string'}]}]}

        Map<string>

        function parse(input) {

            if(!input) {

                return;

            }

            // 去掉左右的尖括号

            if(/^\<[\W\w]+\>$/.test(input)) {

                input = input.replace(/^\<|\>$/g, function() {

                    return '';

                })

            }

            // 正则

            var obj = {};

            var reg = /\<[\W\w]+\>$/g;

            var typeArgs;

            var type = input.replace(reg, function(a, b, c) {

                console.log('---typeArgs---', a);

                if(a) {

                    typeArgs = a;

                    obj.typeArgs = [];

                    obj.typeArgs.push(parse(typeArgs));

                }

                return '';

            })

            obj.type = type;

            return obj;

        }

// 7\_自增ID

        var checkName = function (srcName, TreeNode) {

            var checkMap = new Map();

            function dfs(TreeNode) {

                if(!checkMap.has(TreeNode.type)) {

                    checkMap.set(TreeNode.type, []);

                }else {

                    if(!checkMap[TreeNode.type][TreeNode.name.slice(5) || 0]) {

                        checkMap[TreeNode.type][TreeNode.name.slice(5) || 0] = true;

                    }

                }

                for(var i = 0; i < TreeNode.children.length; i++) {

                    dfs(TreeNode.children[i]);

                }

            }

            dfs();

            if(checkMap.has(srcName)) {

                var temp = checkMap.srcName;

                for(var i = 0; i < temp; i++) {

                    if(!temp[i]) {

                        return `${srcName}\_${i}`;

                    }

                }

            }else {

                return srcName;

            }

        }