答题说明：

1. 所有题目均为开卷
2. 根据个人能力和兴趣选做
3. 答卷限时2天

# 程序设计部分

**第1题\***

将以下折半查找（顺序是从小到大）的空缺补全

**public** **int** search(**int**[] array, **int** value) {

**int** min = 0;

**int** max = array.length - 1;

**while** (max >= min) {

**int** middle = (min + max) / 2;

**if** (array[middle] == value) {

**return** middle;

} **else** **if** ( array[middle] < value ) {

\_ min = middle + 1;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

} **else** {

\_ max = middle - 1;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

}

**return** -1;

}

**第2题\***

请完善以下程序，输入为字符数组（数组内每个字符均不相同）和个数n，返回值是由这些字符组成的所有长度为n的字符串

例如输入为{'a','b','c'}和个数1，则返回值为{"a","b","c"}

例如输入为{'a','b','c','d'}和个数2，则返回值为{"ab","ac","ad","ba","bc","bd","ca","cb","cd","da","db","dc"}

**解答如下**：

**import java.util.ArrayList;**

**import java.util.Arrays;**

**import java.util.List;**

**public class Test6 {**

**public static String[] doit(char[] chars, int n){**

**if(n > chars.length){**

**String[] tt = {};**

**return tt;**

**}**

**if(n == 1){**

**String[] strs = new String[chars.length];**

**for(int i = 0; i < chars.length; i++){**

**strs[i] = String.valueOf(chars[i]);**

**}**

**return strs;**

**}**

**List<String> reLst = new ArrayList<String>();**

**for(int j = 0; j < chars.length; j++){**

**char[] tc = new char[chars.length - 1 - j];**

**for(int h = j + 1; h < chars.length; h++){**

**tc[h - 1 - j] = chars[h];**

**}**

**String[] preRes = doit(tc, n - 1);**

**for(int k = 0; k < preRes.length; k++){**

**char[] tchs = preRes[k].toCharArray();**

**for(int l =0; l < n; l++){**

**char[] chs = new char[n];**

**for(int h = 0; h < n; h++){**

**if(h == l) chs[h] = chars[j];**

**if(h < l) chs[h] = tchs[h];**

**if(h > l) chs[h] = tchs[h-1];**

**}**

**reLst.add(String.valueOf(chs));**

**}**

**}**

**}**

**return (String[])reLst.toArray(new String[0]);**

**}**

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

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

**char[] chars = {'a','b','c','d'};**

**System.out.println(Arrays.toString(doit(chars, 2)));**

**}**

**}**

**第3题\*\***

已知类定义如下

**class** Node {

**public** Double value;

**public** List<Node> children;

}

将以下递归程序改成非递归程序（两个程序的输出应保持顺序一致）

**public** **static** **void** process(Node node) {

**for** (**int** i = 0; i < node.children.size(); i++) {

Node child = node.children.get(i);

**if** (child.value < node.value) {

*process*(child);

}

}

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

**for** (**int** i = 0; i < node.children.size(); i++) {

Node child = node.children.get(i);

**if** (child.value >= node.value) {

*process*(child);

}

}

}

**解答如下**：

/\*

\* 未经测试程序

\*

\* \*/

**public** **void** doit(Node root){

java.util.Stack<Node> st = **new** java.util.Stack<>();

java.util.Map<Node,Boolean> visitedNodes = **new** java.util.HashMap<>();

st.push(root);

Node current = **null**;

**while** (!st.isEmpty()) {

current = st.peek();//获得栈顶元素，不删除

**boolean** hasLessChildrenVisted = **false**;//左子树（value 小的子树）是否访问

**for**(Node child : current.children){

**if**(child.value < current.value){

**if**(visitedNodes.get(child) != **null**){

hasLessChildrenVisted = **true**;

}

**break**;

}

}

**if**(hasLessChildrenVisted){//左子树（value 小的子树）入栈

**for** (**int** i = current.children.size(); i >= 0; i--) {

**if** (current.children.get(i).value < current.value) {

st.push(current.children.get(i));

}

}

**continue**;

}

System.***out***.println(current.value);

visitedNodes.put(current, Boolean.***TRUE***);

st.pop();//访问后出栈

**for** (**int** i = current.children.size(); i >= 0; i--) {//右子树（value 大的子树）入栈

**if** (current.children.get(i).value >= current.value) {

st.push(current.children.get(i));

}

}

}

}

**第4题\*\*\***

参考第1题Node类定义，Node类定义了一个树形结构，请完善以下程序

输入node满足以下条件：

1 node的value是大于0的浮点数

2 node的下级节点（以及更下级节点）的value可能是null或者大于0的浮点数

程序的作用如下：

1 将树形结构里面所有value是null的均设为大于0的浮点数

2 非叶子节点（即children数量大于0的节点）的value均等于它的children的value之和

**解答如下**：

/\*

\* 未经测试程序

\* 对于value未null的节点，找到一个值，将来设定的值必须大于此值

\* \*/

**private** Double getValuMoreThan(Node root){

**if** (root.value != **null**){

**return** root.value;

}

**if**(root.children.size() == 0){

**return** 0.0;

}

Double valueMoreThan = 0.0;

Double tmpValue = **null**;

**for**(Node child : root.children){

valueMoreThan += getValuMoreThan(child);

}

**return** valueMoreThan;

}

/\*

\* 未经测试程序

\*

\* \*/

**public** **void** doit(Node root){//默认root必须有value

**int** numOfNull = 0;

Double sum = 0.0;

**for**(Node child : root.children){

**if**(child.value == **null**){

sum += **this**.getValuMoreThan(child);

numOfNull += 1;

}**else**{

sum += child.value;//child 有value的话就加value

}

}

Double average = (root.value - sum)/numOfNull;

**for**(Node child : root.children){

**if**(child.value == **null**){

child.value = **this**.getValuMoreThan(child) + average;//重复调用this.getValuMoreThan，可以在上次调用时保存值进行优化

}

doit(child);

}

}示例



解答



# 逻辑与几何部分

**第1题\*\***

6个球，A和B轮流拿，每次可以拿走1到4个，拿最后一个的输，如果A先拿， \_B\_\_\_能赢。规律是\_无论A拿几个球（1-4），B拿的时候总可以保证剩下一个。

**第2题\*\*\***

20个球，按顺序标上序号1-20，将这些球放进一个箱子，老师从这个箱子中随机取出2个球，序号分别是a和b，将a与b的和告诉学生甲，将a与b的积告诉学生乙，然后

甲问乙：你知道a和b分别是多少吗？

乙回答：不知道

乙问甲：你知道a和b分别是多少吗？

甲回答：不知道

乙说：呵呵，我知道了

甲说：呵呵，我也知道了

请回答a和b分别是多少，并写出推理过程。

**解答如下**：

1）乙不知道答案 =》M=a\*b 解不唯一。

可得M ∈(6，8，10，12，14，15，16，18，20…) (推论1)

2）甲不知道答案 =》S=a+b解不唯一 =》 5 <= S <= 37

分两种情况： S=5，S=6时a，b有两组解 S>=7 时a，b有三组及三组以上解

1.假设 S=5 则有两组解

a=1，b=4。M=a\*b=4 (不满足推论1，舍去)

a=2，b=3。M=a\*b=6。得到唯一解a=2，b=3。即甲知道答案，“甲不知道答案“相矛盾

2.假设 S=6 则有两组解

a=1，b=5。M=a\*b=4 (不满足推论1，舍去)

a=2，b=4。M=a\*b=8。得到唯一解a=2，b=4。即甲知道答案，“甲不知道答案“相矛盾

因此：S >= 7。（推论2）

3）由题设条件：乙说“那我知道了” =>乙通过已知条件M=a\*b及推论(1)(2)可以得出唯一解。

当M=6时：有两组解 (a = 1, b =6) 和 (a = 2, b=3). 但是后者不满住推论2(S>=7), 所以 前者是唯一解；

当M=8时：有两组解 (a = 1, b =8) 和 (a = 2, b=4). 但是后者不满住推论2(S>=7), 所以 前者是唯一解；

当S>8 时：容易证明均为多组解。

所以 a 和 b有两组解：（1，6）和 （1， 8）。