**《数据结构》课程实践报告**

|  |  |
| --- | --- |
| 课程实践题目： | **数组和树主题实践报告** |
| 姓 名： | **刘柏田** |
| 院系： | **计算机** |
| 专 业： | **软件工程** |
| 年 级： | **大二** |
| 学 号： | **199050418** |
| 指导教师： | **任雪萍** |

年 月 日

**(任雪萍编写)**

**目 录**

1 该课程设计的目的………………………………………………………………x

2 任务分析…………………………………………………………………………x

3分析设计…………………………………………………………………………x

4调试分析…………………………………………………………………………x

5测试结果…………………………………………………………………………x

6小结………………………………………………………………………………x

7用户手册…………………………………………………………………………x

8附录………………………………………………………………………………x

9参考文献…………………………………………………………………………x

**各部分要求如下**：

1. **该课程设计的目的**

熟练使用C或C++或java编写程序，强化模块设计理念；

解决的实际问题（与题目对应）。

提高综合运用所学的理论知识和方法，独立分析和解决问题的能力（与题目相结合）。

初步掌握软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能；

了解并掌握数据结构与算法的设计方法，具备初步的独立分析和设计能力；

1. **任务分析**

无歧义的任务说明：

1. 以下是一个5\*5阶螺旋方阵。设计一个算法输出该形式n\*n（n<10）阶方阵（顺时针方向旋进、逆时针方向旋进）。

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

1. 12 11 10 9

2、 一棵完全二叉树以顺序方式存储在数组A的n个元素中。设计一个算法构造该二叉树的二叉链表存储，并判断这棵二叉树是否为完全二叉树。

系统实现的功能：（1）给定方阵的阶数n，输出螺旋矩阵（2）输入指定格式的数据 生成二叉树并判断是否为完全二叉树

测试数据：（包括合法和非法输入，及相应的合理输出，文字形式描述）

（1）

输入：

6

输出：

1 2 3 4 5 6

20 21 22 23 24 7

19 32 33 34 25 8

18 31 36 35 26 9

17 30 29 28 27 10

16 15 14 13 12 11

输入：

4

输出：

1 2 3 4

12 13 14 5

11 16 15 6

10 9 8 7

（2）

输入：8 2 null 4 9 8 null null 4

输出：false

输入：9 1 2 3 4 5 6 7 8 9

输出：true

1. **分析设计**

需要处理的数据的逻辑结构(线性结构、树形结构、图状结构和集合)：线性结构 树形结构

合适的存储结构（顺序存储、链式储存、索引顺序存储和散列存储等）： 链式存储 顺序存储

设计好的数据类型： 队列 二维数组

与功能对应的模块划分定义（定义时，写出函数头）：

* Left\_to\_right 从左到右进行输出
* Top\_to\_bottom 从上到下进行输出
* Right\_to\_left 从右到左进行输出
* Bottom\_to\_top 从下到上进行输出
* Solve 对输入的Integer数组转换成链式树结构
* isCompleteTree 对链式树结构进行判断是否是完全二叉树

模块调用关系：

按照需要的顺序分别调用四个方法

先调用 solve 对数组进行转换 再调用 isCompleteTree 对链式树结构进行判断

1. **调试分析**

碰到的调试问题：

对于高阶的矩阵（n >32）的输出有异常

完全二叉树的判断不够简便 效率不够高

相应的解决方案：

对问题进行分步处理 每一步的解决方案通用化

算法的时间复杂性：O(m\*m)

空间复杂性：O(m\*m)

教师提出的修改意见：对完全二叉树的判断应该更直接一点 条件设定的更直接一些 判断所需的步骤少一些

修改结果： 对完全二叉树的判断放在前面的代码一部分 有部分情况不需要等到后面的代码 所需要的时间减少

**测试结果**

结论：（程序是否正确?稳健？）

对极端的情况如Integer的临界值等情况没有考虑的足够周到

判断完全二叉树的解法有不稳健的地方 可能会导致程序崩溃和错误

1. **小结**

设计是否合理？(从提高程序质量的角度考虑，比如：有无冗余代码等)

没有考虑到有上限或者其中一些小限制的地方，这些内容是能够减少后续无用的处理量的

可以改进的设想（具体分析，包括改进目标和改进手段等）：

（1）对方法的编写的时候更应该考虑到通用的解法，对数据的限制不应该写的特别死

（2）判断条件的时候不需要全部都等到生成完就判断 可以在生成二叉树的时候就对有显然的非完全二叉树特点的树进行判断 减少后续的代码运行开销

**用户手册**

机器硬件及操作系统有无要求？

无 ，Java对跨平台支持度高

需要什么软件运行测试？

JetBrains Inteligence Idea

运行后，该怎么操作？（说明每一个步骤）

打开项目后运行src\Ch56\ArrayToTree.java 或者 NLevelMatrix 中的 main 函数

**附录**

程序清单如下：



**参考文献**

参考的任何资料（包括网上的资料）