# 数据结构与算法总结论文(五篇)

来源：网络 作者：红叶飘零 更新时间：2024-06-27

*总结是对过去一定时期的工作、学习或思想情况进行回顾、分析，并做出客观评价的书面材料，它可使零星的、肤浅的、表面的感性认知上升到全面的、系统的、本质的理性认识上来，让我们一起认真地写一份总结吧。总结书写有哪些要求呢？我们怎样才能写好一篇总结呢...*

总结是对过去一定时期的工作、学习或思想情况进行回顾、分析，并做出客观评价的书面材料，它可使零星的、肤浅的、表面的感性认知上升到全面的、系统的、本质的理性认识上来，让我们一起认真地写一份总结吧。总结书写有哪些要求呢？我们怎样才能写好一篇总结呢？这里给大家分享一些最新的总结书范文，方便大家学习。

**数据结构与算法总结论文篇一**

算法与数据结构这一门课程，就是描述了数据的逻辑结构，数据的存储结构，以及数据的运算集合在计算机中的运用和体现。数据的逻辑结构就是数据与数据之间的逻辑结构；数据的存储结构就包含了顺序存储、链式存储、索引存储和散列存储。在这学期当中，老师给我们主要讲了顺序存储和链式存储。最后数据的运算集合就是对于一批数据，数据的运算是定义在数据的逻辑结构之上的，而运算的具体实现依赖于数据的存储结构。

通过这学期的学习，让我在去年c语言的基础上对数据与数据之间的逻辑关系有了更深的理解和认识。以前在学matlab这一课程的时候，我们如果要实现两个数的加减乘除，或者一系列复杂的数据运算，就直接的调用函数就行，套用规则符号和运算格式，就能立马知道结果。在学习c语言这一课程时，我们逐渐开始了解函数的调用的原理，利用子函数中包含的运算规则，从而实现函数的功能。现今学习了算法，让我更深层次的知道了通过顺序表、指针、递归，能让数据算法的实现更加的简洁，明了，更易于理解。摒弃了数据的冗杂性。

在本书第二章中，主要介绍了顺序表的实现以及运用。顺序表中我认为最重要的是一个实型数组，和顺序表的表长，不论是在一个数据的倒置、插入、删除以及数据的排序过程中，都能将数据依次存入数组当中，利用数组下标之间的关系，就能实现数据的一系列操作了。在存储栈中，给我留下最深刻的映像就是“先进后出”，由于它特殊的存储特性，所以在括号的匹配，算术表达式中被大量应用。在存储队列之中，数据的删除和存储分别在表的两端进行操作，所以存储数据很方便。为节省队列浪费闲置空间的这一大缺点，所以引入了循环队列这一概念，很好用。

在第三章中，主要讲的是链式存储特性。它最突出的优点就是可以选择连续或者不连续的存储空间都行。所以，不管是数据在插入或者删除一个数据时，会很方便，不会像顺序表那样，要移动数组中的诸多元素。所以链表利用指针能很方便的进行删除或者插入操作。而链式在栈和队列的基础上，也有了多方面的应用，所以在这些方面有了更多的应用。

第四章字符串中，基本的数组内部元素的排序和字符串的匹配大部分代码自己还是能够理解，能够看懂，如果真的要将所学的大量运用于实践的话，那就要多花些功夫和时间了。在对称矩阵的压缩，三角矩阵的压缩，稀疏矩阵在存储中能够合理的进行，能大大提高空间的开支。

在第五章递归当中，就是在函数的定义之中出现了自己本身的调用，称之为递归。而递归设计出来的程序，具有结构清晰，可读性强，便于理解等优点。但是由于递归在执行的过程中，伴随着函数自身的多次调用，因而执行效率较低。如果要在追求执行效率的情况下，往往采用非递归方式实现问题的算法程序。

在第六章数型结构当中，这是区别于线性结构的另一大类数据结构，它具有分支性和层次性。它是数据表示，信息组织和程序设计的基础和工具。在本章中，映像深刻的是树的存储结构。有双亲表示法，孩子表示法，以及孩子兄弟表示法。在表示怎样存储数据之后，接着要从数型结构中将数据读取出来，于是，有了树的遍历，在遍历当中，又分为前序、中序和后序遍历，这三种遍历各有各的特点。

在第七章中，说到了树的扩展---二叉树。二叉树不同一般的树型结构的另一种重要的非线性结构，它是处理两种不同的数据结构，许多涉及树的算法采用二叉树表示和处理更加便捷和方便。其他的也是和一般的二叉树差不多。还多了一个树、森林和二叉树之间的转换。

第八章的围绕着图来展开，它是一种复杂的非线性结构，在人工智能、网络工程、数学、并行计算和工业设计有着广泛的应用。图最重要的由一个非空的顶点集合和一个描述顶点之间的多对多关系的边集合组成的一种数据结构。图的存储室通过邻接矩阵老存储图的信息。而图的读取是通过深度优先遍历和广度优先遍历实现。生成最小生成树有prim算法和kruskal算法，相对于这两种算法，后一种算法要更加易于理解。

在考试的时候，我以为老师只会出题作业部分。然后书中有一小部分就没看，但是题中出现了一个二叉树转换为森林的时候，我有印象，但就是没思路想法了，就没做。从中我真的理解了老师说的，考试不代表学习的结束。或者你现在看的内容在生活中学习中暂时没有太大的作用，但是到了某一特定的环境条件下，总会有作用。所以，学习是一个积累的过程，不懈怠，踏实的走下去，你才会有所收获。

**数据结构与算法总结论文篇二**

《数据结构与算法》课程学习总结报告

070401301507计本（3）班张浩

本学期开设的《数据结构与算法》课程已经告一段落，现就其知识点及其掌握情况、学习体会以及对该门课程的教学建议等方面进行学习总结。

一、《数据结构与算法》知识点

在课本的第一章便交代了该学科的相关概念，如数据、数据元素、数据类型以及数据结构的定义。其中，数据结构包括逻辑结构、存储结构和运算集合。逻辑结构分为四类：集合型、线性、树形和图形结构，数据元素的存储结构分为：顺序存储、链接存储、索引存储和散列存储四类。紧接着介绍了一些常用的数据运算。最后着重介绍算法性能分析，包括算法的时间性能分析以及算法的空间性能分析。

第二章具体地介绍了顺序表的概念、基本运算及其应用。基本运算有：初始化表、求表长、排序、元素的查找、插入及删除等。元素查找方法有：简单顺序查找、二分查找和分块查找。排序方法有：直接插入排序、希尔排序、冒泡排序、快速排序、直接选择排序及归并排序等。最后介绍了顺序串的概念，重点在于串的模式匹配。

链表中数据元素的存储不一定是连续的，还可以占用任意的、不连续的物理存储区域。与顺序表相比，链表的插入、删除不需要移动元素，给算法的效率带来较大的提高。链表这一章中介绍了链表的节点结构、静态与动态链表的概念、链表的基本运算（如求表长、插入、查找、删除等）、单链表的建立（头插法和尾插法）以及双向循环链表的定义、结构、功能和基本算法。

堆栈与队列是两种运算受限制的线性结构。其基本运算方法与顺序表和链表运算方法基本相同，不同的是堆栈须遵循“先进后出”的规则，对堆栈的操作只能在栈顶进行；而队列要遵循“先进先出”的规则，教材中列出了两种结构的相应算法，如入栈、出栈、入队、出队等。在介绍队列时，提出了循环队列的概念，以避免“假溢出”的现象。

第六章介绍了特殊矩阵和广义表的概念与应用。其中，特殊矩阵包括对称矩阵、三角矩阵、对角矩阵和稀疏矩阵，书中分别详细介绍了它们的存储结构。稀疏矩阵的应用包括转置和加法运算等。最后介绍了广义表的相关概念及存储结构，关于它的应用，课本中举了m元多项式的表示问题。

第七章二叉树的知识是重点内容。在介绍有关概念时，提到了二叉树的性质以及两种特殊的二叉树：完全二叉树和满二叉树。接着介绍二叉树的顺序存储和链接存储以及生成算法。重点介绍二叉树的遍历算法（递归算法、先序、中序和后序遍历非递归算法）和线索二叉树。二叉树的应用：基本算法、哈弗曼树、二叉排序树和堆排序。

树与二叉树是不同的概念。教材介绍了树和森林的概念、遍历和存储结构，还有树、森林和二叉树的相互关系，树或森林怎样转化成二叉树，二叉树又如何转换为树和森林等算法。散列结构是一种查找效率很高的一种数据结构。本章的主要知识点有：散列结构的概念及其存储结构、散列函数、两种冲突处理方法、线性探测散列和链地址散列的基本算法以及散列结构的查找性能分析。

最后一章介绍了图的概念及其应用，是本书的难点。图的存储结构的知识点有：邻接矩阵、邻接表、逆邻接表、十字链表和邻接多重表。图的遍历包括图的深度优先搜索遍历和广度优先搜索遍历。其余知识点有：有向图、连通图、生成树和森林、最短路径问题和有向无环图及其应用。有向无环图重点理解aov网和拓扑排序及其算法。

二、对各知识点的掌握情况

总体来看，对教材中的知识点理解较为完善，但各个章节均出现有个别知识点较为陌生的现象。现将各个章节出现的知识点理解情况列举如下。

第一章中我对数据和数据结构的概念理解较为透彻，熟悉数据结构的逻辑结构和存储结构。而对算法的时间、空间性能分析较为模糊，尤其是空间性能分析需要加强。

第二章，顺序表的概念、生成算法理解较为清晰，并且熟悉简单顺序查找和二分查找，对分块查找较为含糊；排序问题中，由于冒泡排序在大一c语言课上已经学习过，再来学习感觉很轻松。对插入排序和选择排序理解良好，但是，在实际运用中仍然出现明显不熟练的现象。由于在归并排序学习中感觉较吃力，现在对这种排序方法仍然非常模糊，所以需要花较多的时间来补习。此外串的模式匹配也是较难理解的一个地方。

链表这一章中，除对双向循环链表这一知识点理解困难之外，其他的知识点像单链表的建立和基本算法等都较为熟悉。

接下来的有关堆栈以及队列的知识点比较少，除有关算法较为特殊以外，其余算法都是先前学过的顺序表和链表的知识，加上思想上较为重视，因此这部分内容是我对全书掌握最好的一部分。不足之处仍然表现在算法的性能分析上。

在学习第六章时感觉较为吃力的部分在于矩阵的应用上，尤其对矩阵转置算法的c语言描述不太理解。稀疏矩阵相加算法中，用三元组表实现比较容易理解，对十字链表进行矩阵相加的方法较为陌生。

第七章是全书的重点，却也有一些内容没有完全理解。在第一节基本概念中，二叉树的性质容易懂却很难记忆。对二叉树的存储结构和遍历算法这部分内容掌握较好，能够熟练运用，而对于二叉树应用中的哈弗曼树却比较陌生。

第八章内容较少，牵涉到所学的队列的有关内容，总体来说理解上没有什么困难，问题依旧出现在算法的性能分析上。

散列结构这一章理解比较完善的知识点有：基本概念和存储结构。散列函数中直接定址法和除留余数法学得比较扎实，对数字分析法等方法则感觉较为陌生。对两种冲突处理的算法思想的理解良好，问题在于用c语言描述上。

最后一章，图及其应用中，图的定义、基本运算如图的生成等起初理解有困难，但随着学习深入，对它的概念也逐步明朗起来。邻接矩阵、邻接表和逆邻接表掌握较好，而对十字链表和邻接多重表则较为陌生。感觉理解较为吃力的内容还有图的遍历（包括深度和广度优先遍历），最小生成树问题也是比较陌生的知识点。最短路径和aov网学习起来感觉比较轻松，而对于c语言描述却又不大明白。

三、学习体会

接触这门课程以前，我对该课程所学的内容有许多疑点，例如：这门课是否是在介绍一种新的计算机语言？如果不是，那么学习这门课程的用途是什么？为什么市面上各种介绍数据结构的资料采用了不同的计算机语言，如c、c++还有java？我的c语言学得不好，对学习这门课是否有影响„„

在学习伊始，老师就明确提出它不是一种计算机语言，不会介绍新的关键词，而是通过学习可以设计出良好的算法，高效地组织数据。一个程序无论采用何种语言，其基本算法思想不会改变。联系到在大一和大二上学期学习的c和c++语言，我深刻认识到了这一点。“软件开发好比写作文，计算机语言提供了许多华丽的辞藻，而数据结构则考虑如何将这些辞藻组织成一篇优秀的文章来。”在学习这门课中，要熟悉对算法思想的一些描述手段，包括文字描述、图形描述和计算机语言描述等。因此，计算机语言基础是必须的，因为它提供了一种重要的算法思想描述手段——机器可识别的描述。

这门课结束之后，我总结了学习中遇到的一些问题，最为突出的，书本上的知识与老师的讲解都比较容易理解，但是当自己采用刚学的知识点编写程序时却感到十分棘手，有时表现在想不到适合题意的算法，有时表现在算法想出来后，只能将书本上原有的程序段誊写到

自己的程序中再加以必要的连接以完成程序的编写。针对这一情况，我会严格要求自己，熟练掌握算法思想，尽量独立完成程序的编写与修改工作，只有这样，才能够提高运用知识，解决问题的能力。

四、对《数据结构与算法》课程教学的建议

1、建议在上课过程中加大随堂练习的分量，以便学生能当堂消化课堂上学习的知识，也便于及时了解学生对知识点的掌握情况，同时有助于学生保持良好的精神状态。

2、建议在课时允许的情况下，增加习题课的分量，通过课堂的习题讲解，加深对知识点的掌握，同时对各知识点的运用有一个更为直观和具体的认识。

以上便是我对《数据结构与算法》这门课的学习总结，我会抓紧时间将没有吃透的知识点补齐。今后我仍然会继续学习，克服学习中遇到的难关，在打牢基础的前提下向更深入的层面迈进！

**数据结构与算法总结论文篇三**

《数据结构与算法》课程学习总结报告

100401200510计本（4）班章兴春

本学期所学习的《数据结构与算法》课程已经告一段落，就其知识点及其掌握情况、学习体会以及对该门课程的教学建议等方面进行学习总结。以便在所学习知识有更深刻的认识。

一、《数据结构与算法》知识点：

学习数据结构之前、一直以为数据结构是一门新的语言、后来才知道学习数据结构是为了更加高效的的组织数据、设计出良好的算法，而算法则是一个程序的灵魂。经过了一学期的数据结构了，在期末之际对其进行总结。首先，学完数据结构我们应该知道数据结构讲的是什么，数据结构课程主要是研究非数值计算的研究的程序设计问题中所出现的计算机处理对象以及它们之间关系和操作的学科。

第一章主要介绍了相关概念，如数据、数据元素、数据类型以及数据结构的定义。其中，数据结构包括逻辑结构、存储结构和运算集合。逻辑结构分为四类：集合型、线性、树形和图形结构，数据元素的存储结构分为：顺序存储、链接存储、索引存储和散列存储四类。最后着重介绍算法性能分析，包括算法的时间性能分析以及算法的空间性能分析。

第二章具体地介绍了顺序表的定义、特点及其主要操作，如查找、插入和删除的实现。需要掌握对它们的性能估计。包括查找算法的平均查找长度，插入与删除算法中的对象平均移动次数。

链表中数据元素的存储不一定是连续的，还可以占用任意的、不连续的物理存储区域。与顺序表相比，链表的插入、删除不需要移动元素，给算法的效率带来较大的提高。链表这一章中介绍了链表的节点结构、静态与动态链表的概念、链表的基本运算（如求表长、插入、查找、删除等）、单链表的建立（头插法和尾插法）以及双向循环链表的定义、结构、功能和基本算法。

第三章介绍了堆栈与队列这两种运算受限制的线性结构。其基本运算方法与顺序表和链表运算方法基本相同，不同的是堆栈须遵循“先进后出”的规则，对堆栈的操作只能在栈顶进行；而队列要遵循“先进先出”的规则，教材中列出了两种结构的相应算法，如入栈、出栈、入队、出队等。在介绍队列时，提出了循环队列的概念，以避免“假溢出”的现象。算法上要求掌握进栈、退栈、取栈顶元素、判栈空盒置空栈等五种操作及掌握使用元素个数计数器及少用一个元素空间来区分队列空、队列满的方法。

第四章串和数组中，我们知道串是一种特殊的线性表，是由零个或多个任意字符组成的字符序列。串的储存结构分为紧缩模式和非紧缩模式。

基本运算需掌握求串长、串赋值、连接操作、求子串、串比较、串定位、串插入、串删除、串替换等。

第五章二叉树的知识是重点内容。在介绍有关概念时，提到了二叉树的性质以及两种特殊的二叉树：完全二叉树和满二叉树。接着介绍二叉树的顺序存储和链接存储以及生成算法。重点介绍二叉树的遍历算法（递归算法、先序、中序和后序遍历非递归算法）和线索二叉树。二叉树的应用：基本算法、哈弗曼树、二叉排序树和堆排序。

树与二叉树是不同的概念。教材介绍了树和森林的概念、遍历和存储结构，还有树、森林和二叉树的相互关系，树或森林怎样转化成二叉树，二叉树又如何转换为树和森林等算法。

第六章介绍了图的概念及其应用，图的存储结构的知识点有：邻接矩阵、邻接表、逆邻接表、十字链表和邻接多重表。图的遍历包括图的深度优先搜索遍历和广度优先搜索遍历。其余知识点有：有向图、连通图、生成树和森林、最短路径问题和有向无环图及其应用。有向无环图重点理解aov网和拓扑排序及其算法。

最后两章集体说明了查找和排序算法，查找教材上介绍了静态查找表和哈希查找表，静态查找表中介绍了顺序查找、折半查找以及分块查找。哈希法中，学习要点包括哈希函数的比较；解决地址冲突的线性探查法的运用，平均探查次数；解决地址冲突的二次哈希法的运用。

排序是使用最频繁的一类算法，可分为内部排序和外部排序。主要需要理解排序的基本概念，在算法上、需要掌握插入排序（包括直接插入排序算法、折半插入排序算法），交换排序（包括冒泡排序算法、快速排序递归算法）,选择排序（包括直接选择排序算法、堆排序算法）等。

二、对各知识点的掌握情况

总体来看，对教材中的知识点理解较为完善，但各个章节均出现有个别知识点较为陌生的现象。现将各个章节出现的知识点理解情况列举如下。

第一章中我对数据和数据结构的概念理解较为透彻，熟悉数据结构的逻辑结构和存储结构。而对算法的时间、空间性能分析较为模糊，尤其是空间性能分析需要加强。

第二章，顺序表的概念、生成算法理解较为清晰，并且熟悉简单顺序查找和二分查找，对分块查找较为含糊；排序问题中，由于冒泡排序在大一c语言课上已经学习过，再来学习感觉很轻松。对插入排序和选择排序理解良好，但是，在实际运用中仍然出现明显不熟练的现象。由于在归并排序学习中感觉较吃力，现在对这种排序方法仍然非常模糊，所以需要花较多的时间来补习。此外串的模式匹配也是较难理解的一个地方。

链表这一章中，除对双向循环链表这一知识点理解困难之外，其他的知识点像单链表的建立和基本算法等都较为熟悉。

接下来的有关堆栈以及队列的知识点比较少，除有关算法较为特殊以外，其余算法都是先前学过的顺序表和链表的知识，加上思想上较为重视，因此这部分内容是我对全书掌握最好的一部分。不足之处仍然表现在算法的性能分析上。

在学习第六章时感觉较为吃力的部分在于矩阵的应用上，尤其对矩阵转置算法的c语言描述不太理解。稀疏矩阵相加算法中，用三元组表实现比较容易理解，对十字链表进行矩阵相加的方法较为陌生。

第七章是全书的重点，却也有一些内容没有完全理解。在第一节基本概念中，二叉树的性质容易懂却很难记忆。对二叉树的存储结构和遍历算法这部分内容掌握较好，能够熟练运用，而对于二叉树应用中的哈弗曼树却比较陌生。

第八章内容较少，牵涉到所学的队列的有关内容，总体来说理解上没有什么困难，问题依旧出现在算法的性能分析上。

散列结构这一章理解比较完善的知识点有：基本概念和存储结构。散列函数中直接定址法和除留余数法学得比较扎实，对数字分析法等方法则感觉较为陌生。对两种冲突处理的算法思想的理解良好，问题在于用c语言描述上。

最后一章，图及其应用中，图的定义、基本运算如图的生成等起初理解有困难，但随着学习深入，对它的概念也逐步明朗起来。邻接矩阵、邻接表和逆邻接表掌握较好，而对十字链表和邻接多重表则较为陌生。感觉理解较为吃力的内容还有图的遍历（包括深度和广度优先遍历），最小生成树问题也是比较陌生的知识点。最短路径和aov网学习起来感觉比较轻松，而对于c语言描述却又不大明白。

由于平时上机练习的少，对于教材中很多算法都掌握的不是很熟悉、不过这些都是可以弥补的，我会在剩下的时间中不断练习书上给出的算法和练习，正如教材上说的，学习数据结构，仅从书本上学习是不够的，必须经过大量的程序设计实践，在实践中体会构造性思维方法，掌握数据组织与程序设计技术。

三、学习体会：

多做实验！这个就没有太多理由了，我一直觉得编程是一门熟练科学，多编程，水平肯定会提高，最重要的是能够养成一种感觉，就是对程序对算法的敏感，为什么那些牛人看一个算法一下子就看懂了？而自己要看很久才能弄懂，而且弄懂了过了一阵子又忘记了？其实这个是因为牛人们以前看的程序很多，编得也很多，所以他们有了那种感觉，所以我觉得大家应该多看程序，多写程序，培养自己的感觉。

复习和考试的技巧，我想大家应该都有这样的感觉，就是觉得自己什么都掌握了，但是在考试的时候就是会犯晕，有时候一出考场就知道错在哪个了，然后考完以后一对答案，发现其实考得很简单，应该都是自己会做的，这个就是与自己的复习和考试的技巧有关系了。

首先就是复习，前面已经说过其实我们学的算法也就是几十个，那么我们的任务也就是理解这几十个算法，复习也就是要加深你的理解。如何理解算法，然后理解到什么程度呢？ 是能默出整个算法吗？其实不是这样的，数据结构的考试有它的特点，考过程考试了，大家应该都发现数据结构其实不要求你把整个算法背出来，它注重考察你的理解，那么怎么考察呢？其实也就是两种方式吧，一种就是用实例，就是给你一个例子，要你用某个算法运行出结果，我想这个期末考试的时候仍然会有很多这样的题目，比如排序那块就很好出这样的题目，要复习这种题目我觉得很简单，就是每个算法都自己用例子去实践一下，以不变应万变，我期中复习的时候就是这样去做的，而且考试之前我就觉得那个并查集的题目就很有可能会考，于是就自己出了几个例子，做了一下。另外一种考察方式就是算法填空和算法改错，可能有一些同学觉得这种题目很难，其实我们首先可以确定这两种题目肯定是与书上算法有关系的，只要理解了书上的算法就可以了，有人觉得看完书以后什么都懂了，而且要默也默得出来，其实不是这样的，算法改错和填空主要是考察的细微处，虽然你觉得你默得出来，那是能够默出算法的主体部分，很多细微的地方你就会很容易忽略。我想大家考过期中考以后应该都有这种感觉吧？那要怎样解决这种问题呢？ 我觉得有两种方法，一种就是自己去编程实现，这种方法比较有意义，还能够提高编程水平，另外一种就是用实例分析算法的每句话，我认为这种方法是最有效的。

然后还有一种题目，就是最后的写算法的题目，我觉得这种题目还是很好解决的，只要是能够自己做出作业的，基本上都会很容易做出来，这也是为什么我前面觉得平时做作业应该自己独立思考的原因，同时做这种题目千万要小心，尤其是题目简单的时候，那肯定会有一些小地方要考虑清楚，一不小心就会被扣掉很多分，这样很不值。

我觉得考试的时候没有太多要讲的，只要复习好了，考试的时候细心一点就可以了，然后就是做一个题目开始就要尽量保证正确，如果觉得留在那里等后面做完了再来检查，这样错误还是很有可能检查不出来，我期中考试的时候就基本上没有检查，因为我做每个题目都是确保正确，用的时间也挺多的，然后也觉得没有检查的必要了。

三、对《数据结构与算法》课程教学的建议

1、建议在上课过程中加大随堂练习的分量，以便学生能当堂消化课堂上学习的知识，也便于及时了解学生对知识点的掌握情况，同时有助于学生保持良好的精神状态。

2、建议在课时允许的情况下，增加习题课的分量，通过课堂的习题讲解，加深对知识点的掌握，同时对各知识点的运用有一个更为直观和具体的认识。

3、要更加重视实验的重要性。

以上便是我对《数据结构与算法》这门课的学习总结，我会抓紧时间将没有吃透的知识点补齐。今后我仍然会继续学习，克服学习中遇到的难关，在打牢基础的前提下向更深入的层面迈进！

**数据结构与算法总结论文篇四**

数据结构与算法课程学习总结报告

11计本一班 许雪松 1104013018

数据结构与算法是计算机程序设计的重要理论技术基础，它不仅是计算机科学的核心课程，而且也已经成为其他理工专业的热门选修课。总的来说感触还是比较深的，刚开始上的时候还蛮简单的，越到后面感觉越难，算法也更复杂了，有时候甚至听不懂，老师上课时讲的也蛮快的，所以只能靠课下下功夫了。下面是我对本学期学习这门课的总结。

一、数据结构与算法知识点

第一章的数据结构和算法的引入，介绍了数据和数据类型、数据结构、算法描述工具、算法和算法评价四个方面的知识。

第二章具体地介绍了顺序表的概念、基本运算及其应用。基本运算有：初始化表、求表长、排序、元素的查找、插入及删除等。元素查找方法有：简单顺序查找、二分查找和分块查找。排序方法有：直接插入排序、希尔排序、冒泡排序、快速排序、直接选择排序及归并排序等。最后介绍了顺序串的概念，重点在于串的模式匹配。

第三章主要介绍的是线性逻辑结构的数据在链接存储方法下数据结构链表的相关知识。主要是单链表、循环链表的数据类型结构、数据结构、基本运算及其实现以及链表的相关应用问题，在此基础上介绍了链串的相关知识。在应用方面有多项式的相加问题、归并问题、箱子排序问题和链表在字符处理方面的应用问题等。本章未完全掌握的是循环链表的算法问题和c的描述。

第四章介绍在两种不同的存储结构下设计的堆栈，即顺序栈和链栈的相关知识，了解堆栈的相关应用，掌握应用堆栈来解决实际问题的思想及方法。本章主要内容是顺序栈和链栈的概念、数据类型、数据结构定义和基本运算算法及其性能分析。本章堆栈算法思想较为简单，所以能较好掌握。

第五章主要介绍顺序存储和链接存储方法下的两种队列、顺序（循环）队列和链队列的数据结构、基本运算及其性能分析以及应用。顺序队列(重点是循环队列）和链队列的概念、数据类型描述、数据结构和基本运算算法及其性能分析等。本章同堆栈有点类似，算法思想较为简单，所以能较好掌握；但难点重在循环队列队空、队满的判断条件问题。

第六章“特殊矩阵、广义表及其应用”将学习数组、稀疏矩阵和广义表的基本概念，几种特殊矩阵的存储结构及其基本运算，在此基础上学习特殊矩阵的计算算法与广义表应用等相关问题。本章的重点是相关数据结构的存储结构及其基本运算算法。掌握了特殊矩阵的压缩存储结构，在该存储结构下元素的定位方法，理解了稀疏矩阵的计算和广义表的存储结构。

第七章二叉树及其应用。分为二叉树的基本概念、二叉树存储结构、二叉树的遍历算法、线索二叉树、二叉树的应用（哈夫曼树、二叉排序树、堆和堆排序、基本算法）。基本算法包括二叉树的建立、遍历、线索化等算法。在此基础上，介绍二叉树的一些应用问题，包括哈夫曼编码问题、（平衡）二叉排序树问题和堆排序问题等。

第八章说的是树和森林，首先我们要知道树与二叉树是不同的概念。课本介绍了树和森林的概念、遍历和存储结构，还有树、森林和二叉树的相互关系，树或森林怎样转化成二叉树，二叉树又如何转换为树和森林等算法。

第九章“散列结构及其应用”是逻辑结构“集合型”的数据元素在散列存储方法下的数据结构及其应用知识内容。主要介绍散列函数的概念、散列结构的概念、散列存储结构的概念---散列表、散列函数和散列表中解决冲突的处理方法---开放定址法、链地址法以及散列表的基本算法及其性能分析。本章概念较为多，所以掌握不太好。

第十章图及其应用。分为图的概念、图的存储结构及其基本算法、图的遍历及算法、有向图的连通性和最小生成树、图的最小生成树、非连通图的生成森林算法、最短路径、有向无环图及其应用。

二、对各知识点的掌握情况

我对各知识点的掌握情况总结如下：

对于第一章对数据结构的概念理解颇深，大概是每次都要谈论到吧。对算法的时间性能，空间性能基本了解。这些在后面的章节都会有运用。第二章本章重点和难点在查找和排序问题的算法思想上，6种排序方法的性能比较。本章未掌握的为希尔排序、快速排序、归并排序的时间复杂度分析。第三章，对链表掌握还好，对其数据结构进行了分析，有循环链表，掌握的不是很好，对其中一些用法不熟练。第四章堆栈，本章堆栈算法思想较为简单，所以能较好掌握，但表达式计算问题未掌握好的。第五章的循环队列队空、队满的判断条件问题掌握的不是很好。第六章的重点是相关数据结构的存储结构及其基本运算算法。掌握了特殊矩阵的压缩存储结构，在该存储结构下元素的定位方法，理解了稀疏矩阵的计算和广义表的存储结构。第七章对二叉树掌握较好，其概念，存储，遍历有很好的掌握。就是对二叉排序树有点生疏，它的生成算法不是很会。第八章树树与二叉树之间的转换，森林与二叉树的转换算法思想基本掌握。第九章散列的一些知识，没有深入学习，大概了解了散列存储结构散列表，散列函数，冲突的处理方法。第十章了解了图的逆邻接表的存储结构，关键路径求解算法未能掌握好，不能灵活运用图的不同数据结构和遍历算法解决复杂的应用问题。

三、学习体会

刚刚接触这门课时，看到课本中全是算法，当时就晕了，因为我的c语言学的不好，我担心会影响这门课的学习，后来上课时老师说学习这门课的基础是c语言，所以我当时就决定一定要好好补补，争取不被拖后腿，在学习这门课的期间，也遇到了不少问。但是通过学习数据结构与算法，让我对程序有了新的认识，也有了更深的理解。同时，也让我认识到，不管学习什么，概念是基础，所有的知识框架都是建立在基础概念之上的，所以，第一遍看课本要将概念熟记于心，然后构建知识框架。并且，对算法的学习是学习数据结构的关键。在第二遍看课本的过程中，要注重对算法的掌握。对于一个算法，读一遍可能能读懂，但不可能完全领会其中的思想。掌握一个算法，并不是说将算法背过，而是掌握算法的思想。我们需要的是耐心。每看一遍就会有这一遍的收获。读懂算法之后，自己再默写算法，写到不会的地方，看看课本想想自己为什么没有想到。对算法的应用上，学习算法的目的是利用算法解决实际问题。会写课本上已有的算法之后，可以借其思想进行扩展，逐步提高编程能力。

四、对课程教学的建议

1、课程课时较紧，课堂上的练习时间较少，讲解的东西越多，头脑有时就很混乱。

2、感觉上课时的气氛不是很好，虽然大部分人都在听，可是效果不是很好。所以希望老师能在授课中间能穿插一些活跃课堂氛围的话题，可以是大家都非常关心的一些内容，这样既让大家能在思考之余有一个放松，也能够提高学生的学习积极性和学习效率。

3、学习的积极性很重要，有时候我们花了很长时间去写实验报告，也很认真的去理解去掌握，可是最后实验报告可能就只得了一个c，抄的人反而得a，这样的话很容易打击学生的积极性，在后面的实验报告中没动力再去认真写。所以希望老师能在这方面有所调整。

4、虽然讲课的时间很紧，但是还是希望老师能在讲述知识点的时候能运用实际的调试程序来给我们讲解，这样的话能让我们对这些内容有更深刻的印象和理解。

**数据结构与算法总结论文篇五**

金陵科技学院实验报告

学 生 实 验 报 告 册

课程名称：

学生学号：

所属院部：计算机工程学院

（理工类）

算法与数据结构 专业班级： 计算机统招(1)班

1413101006 学生姓名： 邢亦波

指导教师： 徐永华 15 ——20 16 学年 第 2 学期

金陵科技学院教务处制

金陵科技学院实验报告

实验报告书写要求

实验报告原则上要求学生手写，要求书写工整。若因课程特点需打印的，要遵照以下字体、字号、间距等的具体要求。纸张一律采用a4的纸张。

实验报告书写说明

实验报告中一至四项内容为必填项，包括实验目的和要求；实验仪器和设备；实验内容与过程；实验结果与分析。各院部可根据学科特点和实验具体要求增加项目。

填写注意事项

（1）细致观察，及时、准确、如实记录。（2）准确说明，层次清晰。

（3）尽量采用专用术语来说明事物。

（4）外文、符号、公式要准确，应使用统一规定的名词和符号。（5）应独立完成实验报告的书写，严禁抄袭、复印，一经发现，以零分论处。

实验报告批改说明

实验报告的批改要及时、认真、仔细，一律用红色笔批改。实验报告的批改成绩采用百分制，具体评分标准由各院部自行制定。

实验报告装订要求

实验批改完毕后，任课老师将每门课程的每个实验项目的实验报告以自然班为单位、按学号升序排列，装订成册，并附上一份该门课程的实验大纲。

金陵科技学院实验报告

实验项目名称： 顺序表 实验学时： 2 同组学生姓名： 无 实验地点： 实验日期： 04.05 实验成绩： 批改教师： 徐永华 批改时间：

金陵科技学院实验报告

实验1 顺序表

一、实验目的和要求

掌握顺序表的定位、插入、删除等操作。

二、实验仪器和设备

turbo c 2.0

三、实验内容与过程（含程序清单及流程图）

1、必做题

（1）编写程序建立一个顺序表，并逐个输出顺序表中所有数据元素的值。编写主函数测试结果。

（2）编写顺序表定位操作子函数，在顺序表中查找是否存在数据元素x。如果存在，返回顺序表中和x值相等的第1个数据元素的序号（序号从0开始编号）；如果不存在，返回－1。编写主函数测试结果。（3）在递增有序的顺序表中插入一个新结点x，保持顺序表的有序性。

解题思路：首先查找插入的位置，再移位，最后进行插入操作；从第一个元素开始找到第一个大于该新结点值x的元素位置i即为插入位置；然后将从表尾开始依次将元素后移一个位置直至元素i；最后将新结点x插入到i位置。

（4）删除顺序表中所有等于x的数据元素。

2、选做题

（5）已知两个顺序表a和b按元素值递增有序排列，要求写一算法实现将a和b归并成一个按元素值递减有序排列的顺序表（允许表中含有值相同的元素）。

金陵科技学院实验报告

程序清单： 1.（1）

#include

#define maxsize 100 #define datatype int typedef struct shun { datatype a[maxsize];int length;}shun;shun s;void init(shun \*s){ } void setup(shun \*s){

} void display(shun \*s){

} main()int i;if(s->length==0)printf(“没有数据n”);else for(i=0;i

length;i++){ } printf(“%-5d”,s->a[i]);int i,j;printf(“需要几个数n”);scanf(“%d”,&s->length);while(s->length>=maxsize){ printf(“需要几个数n”);scanf(“%d”,&s->length);} for(i=0;i

length;i++){ } scanf(“%d”,&s->a[i]);s->length=0;printf(“溢出n”);

金陵科技学院实验报告

{

} init(&s);setup(&s);display(&s);

1.（2）

#include

#define maxsize 100 #define datatype int typedef struct shun { datatype a[maxsize];int length;}shun;shun s;void init(shun \*s){ } void setup(shun \*s){

} int find(shun \*s,int x){ int i;for(i=0;i

length;i++){ int i,j;printf(“需要几个数n”);scanf(“%d”,&s->length);while(s->length>=maxsize){ printf(“需要几个数n”);scanf(“%d”,&s->length);} for(i=0;i

length;i++){ } scanf(“%d”,&s->a[i]);s->length=0;printf(“溢出n”);

金陵科技学院实验报告

if(s->a[i]==x)return i;} return 0;} void display(shun \*s){

} main(){

} int x;init(&s);setup(&s);display(&s);printf(“输入xn”);scanf(“%d”,&x);if(find(&s,x))printf(“找到位置是%dn”, find(&s,x));printf(“-1n”);else int i;if(s->length==0)printf(“没有数据n”);else for(i=0;i

length;i++){ } printf(“%-5d”,s->a[i]);

1.(3)#include

#define maxsize 100 #define datatype int typedef struct shun { datatype a[maxsize];int length;}shun;shun s;void init(shun \*s){ s->length=0;

金陵科技学院实验报告

} void setup(shun \*s){

} void insert(shun \*s,int x){ int i,j;if((s->length+1)>=maxsize){ printf(“溢出n”);exit(0);int i,j;printf(“需要几个数n”);scanf(“%d”,&s->length);while(s->length>=maxsize){ printf(“需要几个数n”);scanf(“%d”,&s->length);} for(i=0;i

length;i++){ } scanf(“%d”,&s->a[i]);printf(“溢出n”);} for(i=0;i

length;i++){ if(s->a[i]>=x)break;} for(j=s->length-1;j>=i;j--){ s->a[j+1]=s->a[j];} s->a[i]=x;s->length++;} void display(shun \*s){

int i;if(s->length==0)printf(“没有数据n”);else for(i=0;i

length;i++)

金陵科技学院实验报告

} main(){

} int x;init(&s);setup(&s);printf(“输入xn”);scanf(“%d”,&x);insert(&s,x);display(&s);{ } printf(“%-5d”,s->a[i]);

1.（4）#include

#define maxsize 100 #define datatype int typedef struct shun { datatype a[maxsize];int length;}shun;shun s;void init(shun \*s){ } void setup(shun \*s){

int i,j;printf(“需要几个数n”);scanf(“%d”,&s->length);while(s->length>=maxsize){ printf(“需要几个数n”);scanf(“%d”,&s->length);s->length=0;printf(“溢出n”);

金陵科技学院实验报告

} void delet(shun \*s,int x){ int i,j;for(i=0;i

length;i++){

} void display(shun \*s){

} main(){

} int x;init(&s);setup(&s);printf(“输入xn”);scanf(“%d”,&x);delet(&s,x);display(&s);int i;if(s->length==0)printf(“没有数据n”);else for(i=0;i

length;i++){ } printf(“%-5d”,s->a[i]);if(s->a[i]==x){

for(j=i;j

length;j++){ s->a[j]=s->a[j+1];} s->length--;i--;} for(i=0;i

length;i++){ } scanf(“%d”,&s->a[i]);} }

金陵科技学院实验报告

2.#include

#define maxsize 100 #define datatype int typedef struct shun { datatype a[maxsize];int length;}shun;shun a,b,c;void init(shun \*s){ } void setup(shun \*s){

} } int i,j,t;printf(“需要几个数n”);scanf(“%d”,&s->length);while(s->length>=maxsize){ printf(“需要几个数n”);scanf(“%d”,&s->length);} for(i=0;i

length;i++){ } for(i=0;i

length;i++){

for(j=i+1;j

length;j++){

} if(s->a[i]

a[j]){

} t=s->a[i];s->a[j]=t;s->a[i]=s->a[j];scanf(“%d”,&s->a[i]);s->length=0;printf(“溢出n”);

金陵科技学院实验报告

void cat(shun \*a,shun \*b){ int i,j=0,t;if((a->length+b->length)>=maxsize){

} for(i=0;ilength;i++){ } for(j=0;j

length;j++){

} =a->length+b->length;

} void display(shun \*s){

} int i;if(s->length==0)printf(“没有数据n”);else for(i=0;i

length;i++){ } printf(“%-5d”,s->a[i]);} for(i=0;i

for(j=i+1;j

} if(c.a[i]

} t=c.a[i];c.a[j]=t;c.a[i]=c.a[j];c.a[i]=b->a[j];i++;c.a[i]=a->a[i];printf(“溢出n”);exit(0);

金陵科技学院实验报告

main(){

} init(&a);printf(“a初始化n”);setup(&a);init(&b);setup(&b);cat(&a,&b);display(&c);printf(“b初始化n”);

金陵科技学院实验报告

四、实验结果与分析（程序运行结果及其分析）1.（1）

1.（2）

金陵科技学院实验报告

1．（3）

1.（4）

金陵科技学院实验报告

2.金陵科技学院实验报告

五、实验体会（遇到问题及解决办法，编程后的心得体会）

我觉得编程不能只要完成其主要功能就行了，还要考虑到边界值，考虑是否会出错等等。有时候一种方法编不通，不如换种方法编。我觉得编程挺考虑耐心的，恩，就这么多感悟了。

金陵科技学院实验报告

实验项目名称： 单链表 实验学时： 2 同组学生姓名： 实验地点： 实验日期： 实验成绩： 批改教师： 批改时间：

金陵科技学院实验报告

实验2 单链表

一、实验目的和要求

1、实验目的

掌握单链表的定位、插入、删除等操作。

2、实验要求

（1）注意链表的空间是动态分配的，某结点不用之后要及时进行物理删除，以便释放其内存空间。

（2）链表不能实现直接定位，一定注意指针的保存，防止丢失。

二、实验仪器和设备

turbo c 2.0

三、实验内容与过程（含程序清单及流程图）

1、必做题

（1）编写程序建立一个单链表，并逐个输出单链表中所有数据元素。（2）在递增有序的单链表中插入一个新结点x，保持单链表的有序性。

解题思路：首先查找插入的位置然后进行插入操作；从第一个结点开始找到第一个大于该新结点值的结点即为插入位置；然后在找到的此结点之前插入新结点；注意保留插入位置之前结点的指针才能完成插入操作。

（3）编写实现带头结点单链表就地逆置的子函数，并编写主函数测试结果。

2、选做题

已知指针la和lb分别指向两个无头结点单链表的首元结点。要求编一算法实现，从表la中删除自第i个元素起共len个元素后，将它们插入到表lb中第j个元素之前。程序清单：

1.（1）

#include

#define datatype char typedef struct lnklist { datatype a;struct lnklist \*next;}list;list \*s;list \*setup(list \*s)

金陵科技学院实验报告

{

} void display(list \*head){ list \*rear;if(head==null){

} } main(){

list \*head;head=setup(s);display(head);free(s);rear=head;printf(“%c”,rear->a);while(rear->next!=null){

} rear=rear->next;printf(“%c”,rear->a);printf(“没有数据n”);else list \*head=null;list \*rear=null;

char c;printf(“请输入c直到$n”);c=getchar();while(c!=\'$\'){

} if(rear!=null)rear->next=null;return head;s=malloc(sizeof(list));s->a=c;if(head==null){ } else rear->next=s;rear=s;c=getchar();head=s;

金陵科技学院实验报告

} free(head);1.(2)#include

#define datatype char typedef struct lnklist { datatype a;struct lnklist \*next;}list;list \*s;list \*setup(list \*s){

} void paixu(list \*head){

list \*rear;list \*p;datatype min;if(head==null){ list \*head=null;list \*rear=null;

char c;printf(“请输入c直到$n”);c=getchar();while(c!=\'$\'){

} if(rear!=null)rear->next=null;return head;s=malloc(sizeof(list));s->a=c;if(head==null){ } else rear->next=s;rear=s;c=getchar();head=s;

金陵科技学院实验报告

} void insert(list \*head,datatype x){

list \*p;list \*q;list \*r;if(head==null)printf(“空表n”);p=head;q=head;r=malloc(sizeof(list));r->a=x;while(p->next!=null){

} while(q->next!=p)q=q->next;r->next=p;q->next=r;if(p->next==null)if(x

a){ } p=p->next;break;} p=head;

while(p->next!=null){

do {

rear=rear->next;if(min>rear->a){

} min=rear->a;rear->a=p->a;p->a=min;

min=p->a;rear=p;printf(“空表！n”);exit(0);} while(rear->next!=null);p=p->next;}

金陵科技学院实验报告

} void display(list \*head){ list \*rear;if(head==null){

} } main(){

} datatype x,c;list \*head;head=setup(s);paixu(head);printf(“请输入xn”);c=getchar();x=getchar();insert(head,x);display(head);free(s);free(head);rear=head;printf(“%c”,rear->a);while(rear->next!=null){

} rear=rear->next;printf(“%c”,rear->a);printf(“没有数据n”);else p->next=r;1.(3)#include

#define datatype char typedef struct lnklist { datatype a;struct lnklist \*next;}list;list \*s;

金陵科技学院实验报告

list \*setup(list \*s){

} list \*nizhi(list \*head){

list \*h;list \*rear;int i=0;char b[100];h=malloc(sizeof(list));h->next=head;rear=head;do {

b[i]=rear->a;rear=rear->next;i++;list \*head=null;list \*rear=null;

char c;printf(“请输入c直到$n”);c=getchar();while(c!=\'$\'){

} if(rear!=null)rear->next=null;return head;s=malloc(sizeof(list));s->a=c;if(head==null){ } else rear->next=s;rear=s;c=getchar();head=s;}while(rear->next!=null);b[i]=rear->a;rear=head;for(;i>=0;i--){

} rear->a=b[i];rear=rear->next;

金陵科技学院实验报告

} void display(list \*head){ list \*rear;if(head==null){

} } main(){

} list \*head;head=setup(s);head=nizhi(head);display(head);free(s);free(head);rear=head;printf(“%c”,rear->a);while(rear->next!=null){

} rear=rear->next;printf(“%c”,rear->a);printf(“没有数据n”);else return head;2.#include

#define datatype char typedef struct lnklist { datatype a;struct lnklist \*next;}list;list \*s1;list \*s2;list \*setup(list \*s){

list \*head=null;list \*rear=null;

char c;printf(“请输入c直到$n”);c=getchar();while(c!=\'$\')

金陵科技学院实验报告

} void dein(list \*la,list \*lb,int i,int len,int j){

int k;list \*rear;list \*t;list \*h;list \*r;list \*q;h=null;rear=la;q=la;for(k=1;k!=i;k++){

} while(q->next!=rear){

t=malloc(sizeof(list));t->a=rear->a;if(h==null)h=t;q=q->next;for(k=1;knext;if(rear->next==null&&k!=i){

} printf(“没找到i的位置n”);exit(0);

{

} if(rear!=null)rear->next=null;return head;s=malloc(sizeof(list));s->a=c;if(head==null){ } else rear->next=s;rear=s;c=getchar();head=s;

金陵科技学院实验报告

} void display(list \*head){ list \*rear;if(head==null){

rear=head;printf(“%c”,rear->a);while(rear->next!=null){

} printf(“n”);rear=rear->next;printf(“%c”,rear->a);printf(“没有数据n”);else

} q->next=rear;if(r!=null)r->next=null;rear=lb;for(k=1;k!=j;k++){

} q=lb;while(q->next!=rear)q=q->next;r->next=rear;q->next=h;rear=rear->next;if(rear->next==null&&k!=j){

} printf(“没找到j的位置n”);exit(0);else r->next=t;r=t;rear=rear->next;if(rear->next==null&&k

} printf(“len太长n”);exit(0);

金陵科技学院实验报告

} } main(){ char c;

} list \*la;list \*lb;int i,len,j;printf(“建立单链表lan”);la=setup(s1);printf(“建立单链表lbn”);lb=setup(s2);printf(“请输入要删的位置in”);scanf(“%d”,&i);printf(“请输入要删减的数据长度lenn”);scanf(“%d”,&len);printf(“请输入要插入的位置jn”);scanf(“%d”,&j);dein(la,lb,i,len,j);printf(“显示lan”);display(la);printf(“显示lbn”);display(lb);free(la);free(lb);c=getchar();

金陵科技学院实验报告

四、实验结果与分析（程序运行结果及其分析）1.（1）

金陵科技学院实验报告

1.（2）

1.（3）

金陵科技学院实验报告

2.金陵科技学院实验报告

五、实验体会（遇到问题及解决办法，编程后的心得体会）

单链表以前没怎么编过，所以现在编有点陌生，要编译好几次才能运行。我觉得还是不能光看书，还要多编几道题比较有手感。

金陵科技学院实验报告

实验项目名称： 堆栈和队列 实验学时： 2 同组学生姓名： 实验地点： 实验日期： 实验成绩： 批改教师： 批改时间：

金陵科技学院实验报告

实验3 堆栈和队列

一、实验目的和要求

（1）掌握应用栈解决问题的方法。（2）掌握利用栈进行表达式求和的算法。

（3）掌握队列的存储结构及基本操作实现，并能在相应的应用问题中正确选用它们。

二、实验仪器和设备

turbo c 2.0

三、实验内容与过程（含程序清单及流程图）

1、必做题

（1）判断一个算术表达式中开括号和闭括号是否配对。（2）测试“汉诺塔”问题。

（3）假设称正读和反读都相同的字符序列为”回文”，试写一个算法判别读入的一个以’@’为结束符的字符序列是否是“回文”。

2、选做题

在顺序存储结构上实现输出受限的双端循环队列的入列和出列算法。设每个元素表示一个待处理的作业，元素值表示作业的预计时间。入队列采取简化的短作业优先原则，若一个新提交的作业的预计执行时间小于队头和队尾作业的平均时间，则插入在队头，否则插入在队尾。程序清单：

1.（1）

#include

#include

char a[100];int panduan(char \*a){

int i,k,count1=0,count2=0;for(i=0;a[i]!=\'\';i++){ {

count1++;for(k=i+1;a[k]!=\'\';k++){ if(a[k]==\')\')if(a[i]==\'(\')

金陵科技学院实验报告

} main(){

} printf(“请输入算式n”);gets(a);if(panduan(a)==1){ } else printf(“算式（）不配对n”);printf(“算式（）配对n”);

break;} if(a[k]==\'\')return 0;} if(a[i]==\')\')} if(count1!=count2)return 0;return 1;count2++;1.（2）

#include

int i;void move(int n,char a,char c){ printf(“第%d步：将%d号盘子%c-%cn”,i++,n,a,c);} void hanno(int n,char a,char b,char c){

} main(){ if(n==1){

} hanno(n-1,a,c,b);move(n,a,c);hanno(n-1,b,a,c);move(1,a,c);else

金陵科技学院实验报告

} int n;char a,b,c;printf(“请输入要移动的盘子数n”);scanf(“%d”,&n);a=\'a\';b=\'b\';c=\'c\';hanno(n,a,b,c);1.（3）

#include

#include

char s[100];int huiwen(char s[]){

} main(){

while(1){ printf(“请输入字符直到@n”);gets(s);if(huiwen(s))

} printf(“是回文n”);printf(“不是回文n”);else int i,j=0;char b[100];for(i=0;s[i]!=\'@\';i++);for(i=i-1;i>=0;i--){

} j=0;for(i=0;s[i]!=\'@\';i++){ } return 1;return 0;b[j]=s[i];j++;if(s[i]!=b[j])j++;

金陵科技学院实验报告

}

2.#include

#define maxsize 100 typedef struct duilie {

int a[maxsize];int head;int rear;}dui;dui \*s;void init(dui \*s){

} void setup(dui \*s,int x){

if(xa[s->head]+s->a[s->rear])/2)){

} else { s->rear=(s->rear++)%maxsize;s->head=(s->head--)%maxsize;s->a[s->head]=x;s->head=maxsize-1;s->rear=maxsize-1;s->a[s->head]=0;s->a[s->rear]=0;

金陵科技学院实验报告

} } s->a[s->rear]=x;void display(dui \*s){

printf(“s队为：”);while(s->head==s->rear){ printf(“%-3d”,s->a[s->head]);

} main(){

} int x;while(1){ printf(“请输入x直到0n”);scanf(“%d”,&x);setup(s,x);if(x==0)} if(s->head!=(s->rear+1)%maxsize)printf(“队满n”);display(s);break;} s->head=(s->head++)%maxsize;

金陵科技学院实验报告

四、实验结果与分析（程序运行结果及其分析）1.（1）

金陵科技学院实验报告

1.（2）

1.（3）

金陵科技学院实验报告

五、实验体会（遇到问题及解决办法，编程后的心得体会）

金陵科技学院实验报告

实验项目名称： 串 实验学时： 2 同组学生姓名： 实验地点： 实验日期： 实验成绩： 批改教师： 批改时间：

金陵科技学院实验报告

实验4 串

一、实验目的和要求

掌握串的存储及应用。

二、实验仪器和设备

turbo c 2.0

三、实验内容与过程（含程序清单及流程图）

1、必做题

（1）编写输出字符串s中值等于字符ch的第一个字符的函数，并用主函数测试结果。

（2）编写输出字符串s中值等于字符ch的所有字符的函数，并用主函数测试结果。

解题思路：可以将第一题程序改进成一个子函数，在本题中循环调用。（3）设字符串采用单字符的链式存储结构，编程删除串s从位置i开始长度为k的子串。

2、选做题

假设以链结构表示串，编写算法实现将串s插入到串t中某个字符之后，若串t中不存在这个字符，则将串s联接在串t的末尾。

提示：为提高程序的通用性，插入位置字符应设计为从键盘输入。程序清单：

1.（1）

#include

#include

void fun(char s[],char ch){

int i;for(i=0;s[i]!=\'\';i++){

} printf(“没找到n”);if(ch==s[i]){

} printf(“找到字符%c在位置%dn”,s[i],i+1);exit(0);

金陵科技学院实验报告

} main(){

} char s[100],ch;printf(“请输入字符串sn”);gets(s);printf(“请输入要查找的字符chn”);scanf(“%c”,&ch);fun(s,ch);1.（2）

#include

#include

char s[100];void fun(char s[],char ch){ int i;if(strcmp(s,“\")==0){ printf(”字符串s为空n“);exit(0);} for(i=0;s[i]!=\'\';i++){

} main(){ char ch;printf(”请输入字符串sn“);gets(s);printf(”请输入要查找的chn“);scanf(”%c“,&ch);fun(s,ch);} if(ch==s[i])printf(” %c“,s[i]);} 1.（3）

金陵科技学院实验报告

#include

#include

typedef struct chuanlian { char c;struct chuanlian \*next;}chuan;chuan \*s;chuan \*setup(chuan \*s){

} void delet(chuan \*chu,int i,int k){

int j;chuan \*p;chuan \*t;if(chu==null){

} p=chu;for(j=1;j

c=ch;if(head==null){ } else

} if(rear!=null)rear->next=null;return head;rear->next=s;rear=s;head=s;s=malloc(sizeof(chuan));ch=getchar();

金陵科技学院实验报告

}

void display(chuan \*chu){ chuan \*p;

} main(){

int i,k;chuan \*head;head=setup(s);printf(”请输入要删除字符的位置in“);scanf(”%d“,&i);p=chu;if(chu==null){

} printf(”%c“,p->c);while(p->next!=null){

} p=p->next;printf(”%c“,p->c);printf(”空串n“);exit(0);{

} t=p->next;for(j=1;j

} p->next=t;if(p->next==null&&j

} t=t->next;printf(”串长度太小，无法删除%d个元素n“,k);exit(0);if(p->next==null&&j

} p=p->next;printf(”无法找到第%d位置n“,i);exit(0);

金陵科技学院实验报告

printf(”请输入要删除字符的个数kn“);scanf(”%d“,&k);delet(head,i,k);display(head);free(head);free(s);} 2.#include

#include

typedef struct chuanlian { char c;struct chuanlian \*next;}chuan;chuan \*s,\*t;chuan \*setup(chuan \*chu){

chuan \*head=null;chuan \*rear=null;char ch;printf(”请输入字符ch直到$n“);ch=getchar();while(ch!=\'$\'){ chu=malloc(sizeof(chuan));

chu->c=ch;if(head==null){ head=chu;

金陵科技学院实验报告

} } else rear->next=chu;rear=chu;ch=getchar();} if(rear!=null)rear->next=null;return head;

void insert(chuan \*s1,chuan \*s2,char x){

chuan \*p;chuan \*q;p=s1;if(s1==null){

} {

} while(p->next!=null){ if(p->c==x)break;printf(”s是空串n“);exit(0);if(s2==null)printf(”t是空串n“);exit(0);

金陵科技学院实验报告

} } p=p->next;if(p->next==null)p->next=s2;else {

} q=s2;while(q->next!=null)q=q->next;q->next=p->next;p->next=s2;void display(chuan \*chu){ chuan \*p;

} p=chu;if(chu==null){

} printf(”%c“,p->c);while(p->next!=null){

} p=p->next;printf(”%c“,p->c);printf(”空串n“);exit(0);

金陵科技学院实验报告

main(){

char x,c;printf(”建立单链串tn“);t=setup(t);c=getchar();printf(”建立单链串sn“);s=setup(s);c=getchar();printf(”请输入要在什么字符后插入n\");x=getchar();

}

insert(t,s,x);display(t);

本文档由站牛网zhann.net收集整理，更多优质范文文档请移步zhann.net站内查找