

数据结构实验总结心得(汇总8篇)

总结不仅仅是总结成绩，更重要的是为了研究经验，发现做好工作的规律，也可以找出工作失误的教训。这些经验教训是非常宝贵的，对工作有很好的借鉴与指导作用，在今后工作中可以改进提高，趋利避害，避免失误。怎样写总结才更能起到其作用呢？总结应该怎么写呢？以下是小编收集整理的工作总结书范文，仅供参考，希望能够帮助到大家。

数据结构实验总结心得篇一

数据结构是计算机科学中的重要概念，涉及到如何组织和存储数据以及如何有效地操作数据。作为一门理论课程，数据结构为我们建立坚实的计算机基础，提供了解决复杂问题的方法和技巧。在学习数据结构理论课程的过程中，我深刻体会到了它的重要性和应用之广泛。接下来，我将通过以下五个方面来分享我对这门课程的理解和体会。

首先，数据结构课程的学习帮助我形成了解决问题的思维方式。在课程中，我们学习了许多经典的数据结构，如数组、链表、栈和队列，它们各自有自己的特点和适用场景。通过学习和实践，我学会了根据问题的需求选择合适的数据结构，并且掌握了数据结构之间的转换和操作方法。这种思维方式培养了我分析问题、抽象问题和解决问题的能力，使我能够更加高效地解决实际应用中的计算机问题。

其次，数据结构课程的学习拓展了我的算法设计能力。在数据结构课程中，我们不仅学习了各种数据结构的实现方式，还学习了如何设计高效的算法来操作和利用这些数据结构。例如，我们学习了排序算法和查找算法，它们在实际应用中非常常见且有广泛的应用场景。通过学习这些算法，我不仅提高了解决问题的效率，还培养了分析和优化算法的能力，使我能够对于复杂的问题快速找到解决方案。

第三，数据结构课程的实践项目增强了我对于理论知识的理解和应用能力。在课程中，我们进行了多个实践项目，如链表的实现、树的遍历等。通过实践项目，我不仅巩固了课堂上所学的知识，还深入理解了数据结构的内部实现原理和运行机制。另外，实践项目还培养了我的团队合作能力和问题解决能力，提高认识到了实际应用中需要解决的问题和挑战。

第四，数据结构课程的学习加深了我对计算机底层的理解。数据结构是计算机的核心概念之一，它们不仅贯穿于计算机科学的各个领域，还直接影响到计算机的性能和效率。通过学习数据结构课程，我不仅了解了计算机底层的数据组织和存储方式，还了解了计算机在处理和操作数据时的一些基本原理，如时间复杂度和空间复杂度。这对于我理解计算机系统的整体架构有很大的帮助，同时也为我后续的学习和研究打下了坚实的基础。

最后，数据结构课程的学习让我认识到了学习的重要性和持续学习的必要性。数据结构是计算机科学中的基础概念，但随着计算机科学的不断发展和进步，数据结构也在不断变化和更新。因此，我们需要持续学习和掌握最新的数据结构和算法，以适应计算机科学领域的发展需求。数据结构课程让我认识到了“学终生”的理念，坚定了我在计算机科学领域不断学习和探索的决心。

总之，数据结构理论课程是计算机科学中极为重要的一门课程，通过学习这门课程，我不仅深刻理解了数据结构的概念和原理，还获得了解决复杂问题的方法和技巧。学习数据结构课程不仅拓展了我的计算机知识，还培养了我解决问题的思维方式、算法设计能力和实践能力。同时，数据结构课程也让我认识到了学习的重要性和持续学习的必要性。这门课程为我打下了坚实的计算机基础，为我在计算机科学领域的未来发展奠定了基石。

数据结构实验总结心得篇二

数据结构是计算机科学中非常重要的一门课程，它涉及到如何组织和存储数据以便高效地使用和管理。作为一名计算机专业的学生，在学习数据结构理论课程的过程中，我获得了很多宝贵的知识和经验。在这篇文章中，我将分享我对这门课程的心得体会。

首先，在学习数据结构理论课程中，我深刻认识到数据结构的重要性。在计算机科学中，许多问题都可以通过合理地选择和应用适当的数据结构来解决。掌握数据结构的基本概念和操作方法，对于编写高效、可维护和可扩展的程序至关重要。例如，当我们需要对大量数据进行排序时，选择合适的排序算法和数据结构可以显著提高程序的执行效率。

其次，数据结构理论课程让我学会了如何分析和评估不同的数据结构。课程中我们学习了各种常见的数据结构，如数组、链表、栈、队列、树、图等。通过对各种数据结构的特点和潜在应用进行深入分析和研究，我能够更好地理解它们的内部机制和实现原理。这使我可以更好地选择和设计适合特定问题的数据结构，并评估其性能和资源消耗。

第三，数据结构理论课程培养了我的问题解决能力和编程思维。在课程中，我们经常面临各种复杂的问题，并尝试用适当的数据结构来解决。这要求我们具备良好的逻辑思维和抽象能力，能够将复杂问题分解为简单的子问题，并找到解决方案。通过反复练习和实践，我的问题解决能力得到了极大的提高，并能够更好地运用编程语言和数据结构来解决实际问题。

第四，数据结构理论课程加深了我对编程语言的理解和应用。在学习数据结构的过程中，我们通常使用编程语言来实现和运行代码。这使我更加熟悉和熟练掌握了编程语言，提高了我的编程水平和技能。通过编写实际代码来实现不同的数据

结构，我可以更好地理解其原理和操作，并能够将其应用到实际项目中。

最后，数据结构理论课程培养了我的团队合作和沟通能力。在课程中，我们经常进行小组项目和作业，需要与队友合作共同完成任务。通过与队友合作，我学会了有效地沟通和协调，学会了分工合作和互相支持，在团队合作中获得了愉快的学习体验和更好的成果。

综上所述，数据结构理论课程对我来说是一门非常重要和有意义的课程。通过学习这门课程，我深刻认识到了数据结构的重要性，学会了分析和评估不同的数据结构，培养了我的问题解决能力和编程思维，加深了对编程语言的理解和应用，并提高了我的团队合作和沟通能力。我相信这些知识和经验将对我未来的学习和工作有着深远的影响和指导。

数据结构实验总结心得篇三

一、课程基本概况

课程名称：数据结构

课程名称（英文） data structures

课程编号 b09042

课程总学时：60（其中，讲课48，实验12）

课程学分：3

课程分类：专业选修课

开设学期：4

适用专业：计算机网络工程本科

先修课程：集合论，图论，高级语言（结构或记录，指针）

后续课程：数据库，编译原理，操作系统等

二、课程的性质、目的和任务

数据结构是计算机专业的一门核心专业课程，是软件课程中非常重要的一门课程，在整个专业教学中占有十分重要的地位，是一门理论性非常强的课程。通过课堂教学、课外练习和上机实习，使学生了解数据对象的特性，数据组织的基本方法，并初步具备分析和解决现实世界问题在计算机中如何表示和处理的能力以及培养良好的程序设计技能，为后续课程的学习和科研工作的参与打下良好的基础。

三、主要内容、重点及深度

本门课程共60学时，其中理论教学48学时，实验教学12学时。其中，理论教学部分：

第一章

绪论

（一）目的要求

了解数据结构的意义与发展过程、数据结构在计算机科学中的作用、学习本课程的目的、任务及要求。理解数据结构的基本概念；算法设计；掌握算法的时间和空间复杂度。

（二）教学内容 本章知识点：

1. 相关的基本概念(掌握)；

2. 算法五大要素(掌握);

3. 计算语句频度和估算算法时间复杂度的方法(掌握)。

(三) 重点与难点

重点: 数据结构的定义; 算法的描述方法。

难点: 数据结构的定义; 算法与程序的区别; 时间复杂度及其计算。

第二章

线性表

(一) 目的要求

掌握线性表的逻辑结构; 线性表的存储结构及操作的实现; 理解一元多项式的表示;

(二) 教学内容 本章知识点:

1. 线性表的逻辑结构(掌握); 2. 线性表的存储结构(掌握);

3. 线性表在顺序结构和链式结构上实现基本操作的方法(掌握);

4. 从时间和空间复杂度的角度比较线性表两种存储结构的不同特点及其适用场合(掌握)。

(三) 重点与难点

重点: 线性表的概念; 线性表的顺序存储结构、链式存储结构及其常用算法。难点: 链式存储结构及其常用算法; 双向循环链表。

第三章 栈和队列

（一）目的要求

掌握栈的定义，表示及实现；表达式求值；栈与递归过程；队列的定义、表示及实现。

（二）教学内容 本章知识点： 1. 栈和队列的特点(掌握)；

2. 在两种存储结构上栈的基本操作的实现(掌握)； 3. 循环队列和链队列的基本运算(熟练掌握)； 4. 递归算法执行过程中栈状态的变化过程(掌握)。

（三）重点与难点

重点：堆栈和队列的概念；递归的定义；循环队列和链队列的基本运算。难点：递归的编程实现；循环队列和链队列的基本运算。

第四章 串

（一）目的要求

了解串的逻辑结构，存储结构；掌握串操作的实现(重点难点bf和kmp算法)串的应用。

（二）教学内容 本章知识点：

1. 串的七种基本运算的定义（了解）；

算法，熟悉next函数和改进next函数的定义和计算(掌握)；

5. 串名的存储映象和在堆存储结构实现串操作的方法(理解)。

（三）重点与难点 重点：串定义和存储方法；串的操作 难

点：串操作实现方法

第五章 数组和广义表

（一）目的要求

掌握数组的存储结构；稀疏矩阵的表示及操作的实现；广义表的定义和存储结构；广义表的递归算法。

4. 广义表的定义及其存储结构，学会广义表的表头，表尾分析方法(掌握)； 5. 学习编制广义表的递归算法(掌握)。

（三）重点与难点

重点：多维数组元素存储地址的计算；稀疏矩阵的三元组表示；广义表的存储定义、操作。难点：稀疏矩阵的三元组表示；广义表的存储定义、操作。

第六章 树和二叉树

（一）目的要求

（二）教学内容 本章知识点： 1. 二叉树的结构特点(理解)；

7. 建立最优二叉树和哈夫曼编码的方法(掌握)。

（三）重点与难点 重点：二叉树的概念、性质；二叉树的遍历方式；构造二叉排序树。难点：二叉树的遍历方式；二叉排序树的构造方法；二叉树的线索化。

第七章 图

（一）目的要求

理解图的基本概念；图的存储结构；掌握图的遍历及应用{最

小生成树，最短路径等}；拓扑排序和关键路径。

（二）教学内容 本章知识点： 1. 熟悉图的各种存储结构；

4. 应用图的遍历算法求各种简单路径问题（比如，最小生成树、最短路径、拓扑排序、关键路径等）（掌握）。

（三）重点与难点

重点：图的存储结构；图的遍历 难点：图遍历的算法；

第八章

动态存储管理

（一）目的要求

了解边界标识法和伙伴系统；无用单元收集和紧缩；

（二）教学内容 本章知识点：

1. 存储器分配策略和算法（了解）；

2. 无用单元收集时的标志算法（了解）。

（三）重点与难点

存储器分配策略和算法、无用单元收集时的标志算法

第九章

查找

（一）目的要求

(二) 教学内容 本章知识点:

1. 顺序查找、折半查找和索引查找的方法、应用(掌握);
2. 二叉排序树的构造方法(掌握);
3. 二叉平衡树的建立方法(掌握);
4. b-树, b+树和键树的特点以及它们的建立过程(理解);
5. 哈希表的构造方法(掌握);
6. 按定义计算各种查找方法在等概率情况下查找成功时和失败时的平均查找长度;
7. 哈希表在查找不成功时的平均查找长度的计算方法(掌握)。

(三) 重点与难点

重点: 二叉排序树的构造方法、二叉平衡树的建立方法; 哈希表的构造、应用;

难点: 二叉排序树的构造及应用; 哈希表的构造方法; 查找的平均长度。

第十章

内部排序

(一) 目的要求

掌握插入排序、交换排序(起泡排序, 快速排序)、选择排序(简单选择, 树形选择, 堆)、归并排序、基数排序等算法。

(二) 教学内容 本章知识点:

1. 各种排序方法的特点并能灵活应用(掌握); 2. 各种方法的排序过程(掌握);
3. 各种排序方法的时间复杂度分析(掌握)。

(三) 重点与难点

重点: 各种排序方法的特点及其应用; 实现排序的各种算法。
难点: 各种排序算法的时间复杂度分析。

十一章

外部排序

(一) 目的要求

理解外部排序的基本方法; 掌握败者树和多路平衡归并的实现; 置换-选择排序; 最佳归并树。

(二) 教学内容 本章知识点:

1. 外部排序的两个过程(理解);
2. 外排过程中所需进行外存读/写次数的计算方法(掌握);
3. 败者树的建立过程(掌握);
4. 实现多路归并的算法(掌握);
5. 置换-选择排序的过程(掌握);
6. 最佳归并树的构造方法(熟悉);
7. 按最佳归并树的归并方案进行平衡归并时, 外存读/写次数的计算方法(掌握)。

（三）重点与难点

重点：外部排序过程和实现方法；多路并归算法及其实现；
难点：最佳并归树的构造方法及其应用。

实践教学部分：上机实验分4个专题，每个专题可提供2~4个难度不等的题目供选。

实验一

停车场管理系统

（一）实验内容 以栈模拟车场，以队列模拟车场外的便道，按照从终端读入的输入数据序列进行模拟管理。栈以顺序结构实现，队列以链表结构实现。

（二）实验过程 编程实现实验内容。

（三）实验教学基本要求

通过实例，使学生掌握栈和队列两种特殊的线性结构，掌握栈和队列的特点。实验后学生提交实验报告。

（四）实验设备和材料 计算机。

（五）实验学时 4学时

实验二

教学计划编制问题

（一）实验内容

假设任何专业都有固定的学习年限，每学年含两学期，每学期的时间长度和学分上限值均相等。每个专业开设的课程都

是确定的，而且课程在开设时间的安排必须满足先修关系。每门课程有哪些先修课程是确定的，可以有任意多门，也可以没有。每门课恰好占一个学期。编制一个教学计划程序。

(二) 实验过程编程实现实验内容。

(三) 实验教学基本要求

通过实例，使学生熟悉图的各种存储结构的特性，掌握如何应用图结构解决具体问题。实验后学生提交实验报告。

(四) 实验设备和材料 计算机。

(五) 实验学时 2学时

实验三

最小生成树问题

(一) 实验内容

利用克鲁斯卡尔算法求最小生成树。以文本形式输出树中各条边以及他们的权值。

(二) 实验过程 编程实现实验内容

(三) 实验教学基本要求

通过实例，使学生熟悉图的各种存储结构的特性，掌握如何应用图结构解决具体问题。实验后学生提交实验报告。

(四) 实验设备和材料 计算机。

(五) 实验学时 2学时

实验四

哈希表设计

（一）实验内容

假设人名为中国人的汉语拼音形式。待填入哈希表的人名共有30个，取平均查找长度的上限为2。哈希函数用除留余数法构造，用伪随机探测再散列法处理冲突。

（二）实验过程 编程实现实验内容

（三）实验教学基本要求 掌握索引技术的使用。

（四）实验设备和材料 计算机

（五）实验学时 4学时

五、课程教学的基本要求和主要环节

本课程可采用课堂讲授、课堂讨论、习题课等进行课堂教学；条件允许可采用cai□电子教案、幻灯片、参观等进行辅助教学；每章布置3~6道习题以巩固教学；在课程后半程，安排3~4个上机实验，让学生应用数据结构的理论、方法，分组设计几个较大的软件，使理论与实际相结合。

考试采用闭卷方式。总成绩由平时成绩和考试成绩组成。平时成绩占30%，考试成绩占70%。

六、本课程与其它课程的联系与分工

先修课包括：集合论，图论，高级语言（结构或记录，指针）；

后续课包括：数据库，编译原理，操作系统等。

七、建议教材与参考教材

《数据结构□□c语言版》

严蔚敏等

清华大学出版社

1997 《数据结构题集》

严蔚敏等

清华大学出版社

1999

《数据结构习题与解析》

李春葆

清华大学出版社

2004

八、负责人

撰稿人：刘景汇、李玉香

审稿人：

系（院）领导：

数据结构实验总结心得篇四

忙碌了一个学期的老师们想必都在好好享受着假期，无论是

选择休闲娱乐，还是继续充电，或者是为下一学期的教育教学工作做准备。作为老师，我是幸运的！假期我有了一个这样的外出学习的机会。那么在这里我想说谢谢小教部给我们林区教师提供一个这样的学习机会，希望这项活动能永葆青春，让每位林区的教师都能受益。下面我就我个人学习谈一点心得体会：

随着教育的不断深入，传统教学的改革已势在必行。变“要我学”为“我要学”就要求教师要给学生充分的时间和空间，给学生提供丰富的学习资料，让他们通过讨论、交流等多种形式的学习，使学生自主学习可以自学的内容。

此次学习我聆听了十位教育专家的讲座，分别从师德修养、教育叙事研究、小学语文阅读法、新课程下的教学反思、教学研究导论、数学课题教学的实效性、语文教学中的朗读指导、数学课堂教学评价指南、有效教学促进学生思考、信息时代下教师专业发展这十个方面对我们进行了培训。

一、提升师德修养

师生关系的新注解，对话与理解、爱与平等。课程实际上对教师提出了教育专业工作者的要求，教师在教学活动中所遇到的“老”问题必须以新的理念为指导，师生关系的正确处理是教学工作的一个关键问题。新课程要求师生关系从对抗走向对话，从压迫走向平等，教师角色从传统的课堂主宰转变为教学的促进者，学生成长的引路人，学生潜能的唤醒者。只有在新型师生关系下，学生的素质、教师的智能、新课程的优势才可得到充分的发挥。

创新是事物发展的动力和源泉，只有不断创新，一个国家才能不断走向前进，一个人才能不断得到完善和提高。没有创新意识，最后的结果必然导致自身价值的实现将被社会进步与发展的洪流所淹没，我们不能总是站在先人创造的历史山峰上沾沾自喜。在知识爆炸，科技迅猛发展的今天，要求我们

要能够根据新形式新任务的需要，积极推动各项工作的与时俱进，克服不思进取，墨守成规的观念和行为，努力发掘解决问题的好方式、好方法。当然创新也不是盲目和随意的，创新也需要以牢固扎实的理论知识和长期的工作实践作为前提和保障，以此为基础进而寻求更为高效、合理、多样的方法来改进我们的工作，提高工作成效，才能进一步实现好、维护好、发展好人民的根本利益。“捧着一颗心来，不带半根草去。”陶行知老先生曾用这句话来高度概括了教师无私奉献的精神。

通过些次学习使我更清楚的认识到了：作为新时代的人民教师，更应该用自己一颗挚热的爱心去了解学生，公平地对待他们，尊重他们，关心、爱护他们。用自己的爱心去开启每一个学生心灵的窗户。让教师的爱心成为孩子学习、生活成功的原动力，激发他们的求知欲和创造欲。

在不断的付出自己对学生的爱心的同时，不断地提高自身的师德修养。成为每名学生的良师益友，使自己的教学环境自然的形成民主、合谐、平等、愉快的教学氛围，成为培养孩子们创造性的思维的主阵地。

二、提升教学理念

新课改的春风吹拂教育界好多年，我心中的教学理念也在不断的提升中，通过这次学习我有了一个更清楚更系统的认识。“振兴民族的希望在教育，振兴教育的希望在教师”，没有教师的支持，任何教育改革都无法进行。此次基础教育课程改革所力图实现“知识与技能、过程与方法以及情感态度价值观”三位一体的课程功能。而此次学习我对这一功能有了简单的体会：

教学目的的新构想，教学的目的应是为了挖掘每一个孩子的天分，培养其独特的个性，教学活动应成为促进学生全面发展的基石，教师则是构筑基石的材料。多元化的发展、多元

化的素质，。新课程所强调的课程内容的选择性正是为了适应学生发展的不同需求而提出的。我们应该用积极乐观的眼光和态度来欣赏和预见学生的天性，应将每一个学生都视为一片有待开发的资源和财富，并进行有效的挖掘和利用。我们首先要改变传统的教学目的理念，认识到教学活动的意义在于培养人的发展所必备的素质，诸如塑造学生健全的人格，健全的意志品质，陶冶丰富的情感，培养研究性学习的能力，树立正确的价值观念等。使学生学会做人、学会求知、学会生活、学会创造、得到全面和谐的发展，成为新一轮课程改革所力求达成的目标。刚走出大学校门，我就很荣幸地成为合肥格力公司中的一员，怀着自己美好的希望和从零开始的心态，开始了自己人生的新征程。

三、提升教学方法

教学活动的新诠释，树立以生命为本的教学过程观、以研究为主体的教学方式观、以促进发展为目的的教学评价。叶澜教授曾经指出：“今日教学改革所要改变的不只是传统的教学理论，还要改变千百万教师的教学观念，改变他们每天都在进行着的、习以为常的教学行为。”作为学生生命成长的重要支柱，教师必须转变观念，认识到教学活动并不是围绕书本传授“符号”的过程，做教材的“奴隶”，而在于以课程为中介，通过师生之同的积极互动，将课程“动态化”，通过体验和发现去开发新的课程，能动地、个性地解读教材，去创建新的课程意义，使课程活起来，使教学活起来，力求达到生命的一次又一次的提升。

以上就是我此次学习的一点心得体会，可能某些方面的认识还很肤浅，但我相信，用我对一颗对教育的执着追求的奉献之心，在以后的工作中会不断的提升自己，完善自己，时刻看到自己、认识自己、丰富自己。

将本文的word文档下载到电脑，方便收藏和打印

推荐度：

点击下载文档

搜索文档

数据结构实验总结心得篇五

计算机数据结构是指在计算机科学中用于组织和存储数据的方法。在计算机科学和编程中，数据结构是实现算法的基础，它能够提高程序的效率和可读性。在学习计算机科学的过程中，计算机数据结构是一个必要的基础课程。在本文中，将会分享笔者学习计算机数据结构的心得体会，总结计算机数据结构的重要性，以及如何更好的学习这门课程。

第二段：学习心得

在课程学习的过程中，我想说的第一件事就是计算机数据结构是充满挑战的。在学习过程中，我们需要学习各种数据结构如链表、堆栈、队列、树、图等等。这些数据结构看起来相似，但在实际应用中不能随意混用，每个数据结构都有其独特的使用方法。在学习这门课程时，我更加深刻地认识到计算机数据结构对计算机科学中算法实现的重要性。

第三段：重要性

计算机数据结构在计算机科学的应用中非常重要，它能够增强计算机程序的执行效率和可读性。通过使用合适的数据结

构，可以使代码更易于理解和维护。例如，使用链表可以快速添加和删除元素，而使用数组则能快速访问元素。因此，学习计算机数据结构可以让我们更好地理解数据的存储方式和查找方法，从而更好地编写程序。

第四段：优化方法

在学习计算机数据结构时，我发现实践是一种很好的学习方法。除了理论课程，我们还应该运用所学的知识来解决实际问题，这样有助于我们更好地理解数据结构的使用和实际应用。此外，合理地使用数据结构的注释也很重要，注释可以让代码更加清晰易懂，也可以帮助我们在处理问题时更快捷地找到所需的数据结构。

第五段：总结

在本文中，我们分享了关于计算机数据结构的心得体会。学习计算机数据结构不仅仅局限于课本上的理论知识，它也需要我们运用所学的知识来解决实际问题。计算机数据结构可以帮助我们更好地理解数据的储存方式和查找方法，从而更好地编写程序。我相信，在使用实践学习的方法下，大家都能更好地理解和应用计算机数据结构。

数据结构实验总结心得篇六

数据结构读书笔记

第一章是绪论部分，因为大家都是刚刚接触这门课，所以还有很多不是很多的了解，随着计算机的迅速发展，计算机的应用领域已经不再只是科学计算领域，而更多的应用于控制管理以及数据处理等非数值计算的处理工作，与此对应，计算机加工处理的对象由纯粹的数值发展到字符，表格和图像等各种具有一定结构的数据，这就给程序设计带来了一些新的问题，为了编写出一个好的程序，必须分析待处理的对象

的特性以及各处理对象之间存在的关系。所以在这种环境下，数据结构这门课就诞生了。

第二章. 线性表的相关基本概念，如：前驱、后继、表长、空表、首元结点，头结点，头指针等概念。

2. 线性表的结构特点，主要是指：除第一及最后一个元素外，每个结点都只有一个前趋和只有一个后继。

方式。此外，近年来在不少学校中还多次出现要求用递归算法实现单链表输出（可能是顺序也可能是倒序）的问题。

5. 线性表的顺序存储及链式存储情况下，其不同的优缺点比较，即其

各自适用的场合。单链表中设置头指针、循环链表中设置尾指针而不设置头指针以及索引存储结构的各自好处。

第三章本章主要重点是1. 栈、队列的定义及其相关数据结构的的概念，包括：顺序栈，链栈，共享栈循环队列，链队等。栈与队列存取数据（请注意包括：存和取两部分）的特点。

2. 递归算法。栈与递归的关系，以及借助栈将递归转向于非递归的经典算法

4. 循环队列中判队空、队满条件，循环队列中入队与出队算法。

第四章1. 串的基本概念，串与线性表的关系（串是其元素均为字符型数据的特殊线

性表），空串与空格串的区别，串相等的条件

2. 串的基本操作，以及这些基本函数的使用，包括：取子串，串连接，串替换，求串长等等。运用串的基本操作去完成特

定的算法是很多学校在基本操作上的考查重点。

3. 顺序串与链串及块链串的区别和联系，实现方式。

算法思想□kmp中next数组以及nextval数组的求法。明确传统模式匹配算法的不足，明确next数组需要改进之外。其中，理解算法是核心，会求数组是得分点。

查方式是：求next和nextval数组值，根据求得的next或nextval数组值给出运用kmp算法进行匹配的匹配过程。

1. 多维数组中某数组元素的position求解。一般是给出数组元素的首元素地址和每个元素占用的地址空间并组给出多维数组的维数，然后要求你求出该数组中的某个元素所在的位置。

2. 明确按行存储和按列存储的区别和联系

3. 将特殊矩阵中的元素按相应的换算方式存入数组中。这些矩阵包括：对称矩阵，三角矩阵，具有某种特点的稀疏矩阵等。熟悉稀疏矩阵的三种不同存储方式：三元组，带辅助行向量的二元组，十字链表存储。掌握将稀疏矩阵的三元组或二元组向十字链表进行转换的算法。

4. 广义表的概念，特别应该明确表头与表尾的定义。这一点，是理解整个广义表一节算法的基础。

1的性质□n个结点的完全二叉树的深度，顺序存储二叉树时孩子结点与父结点之间的换算关系（左为： $2*i$ □右为： $2*i+1$ □□

2. 二叉树的三种遍历算法

种递归算法改造而来（比如：求叶子个数），所以，掌握了三种遍历的非递归算法后，对付诸如：“利用非递归算法求二叉树叶子个数”

3. 可在三种遍历算法的基础上改造完成的其它二叉树算法：

归遍历算法，那么解决以上问题就是小菜一碟了。

4. 线索二叉树：

线索二叉树的引出，是为避免如二叉树遍历时的递归求解。对于线索二叉树，应该掌握：线索化的实质，三种线索化的算法，线索化后二叉树的遍历算法，基本线索二叉树的其它算法问题（如：查找某一类线索二叉树中指定结点的前驱或后继结点就是一类常考题）。

5. 最优二叉树（哈夫曼树）：

最优二叉树是为了解决特定问题引出的特殊二叉树结构，它的前提是给二叉树的每条边赋予了权值，这样形成的二叉树按权相加之和是最小的。

6. 树与森林：

二叉树是一种特殊的树，这种特殊不仅仅在于其分支最多为2以及其它特征，一个最重要的特殊之处是在于：二叉树是有序的！即：二叉树的左右孩子是不可交换的，如果交换了就成了另外一棵二叉树，这样交换之后的二叉树与原二叉树我们认为是不相同的两棵二叉树。但是，对于普通的双分支树而言，不具有这种性质。

树与森林的遍历，不像二叉树那样丰富，他们只有两种遍历算法：先根与后根（对于森林而言称作：先序与后序遍历）。在难度比较大的考试中，也有基于此种算法的基础上再进行扩展要求你利用这两种算法设计其它算法的，但一般院校很少有这种考法，最多只是要求你根据先根或后根写出他们的遍历序列。此二者的先根与后根遍历与二叉树中的遍历算法是有对应关系的：先根遍历对应二叉树的先序遍历，而后根遍

历对应二叉树的中序遍历。

第七章 图

图这一章的特点是:概念繁多,与离散数学中图的概念联系紧密,算法复杂与图两章的知识这一章的重点是:图的定义和特点,无向图,有向图,入度,出度,完全图,生成子图,路径长度,回路,连通图,(强)连通分量等概念。

2. 图的几种存储形式:

图的存储形式包括:邻接矩阵,(逆)邻接表,十字链表及邻接多重表。

3. 图的两种遍历算法:深度遍历和广度遍历

深度遍历和广度遍历是图的两种基本的遍历算法,这两个算法对图一章的重要性等同于“先序、中序、后序遍历”对于二叉树一章的重要性。在考查时,图一章的算法设计题常常是基于这两种基本的遍历算法而设计的,比如:“求最长的最短路径问题”和“判断两顶点间是否存在长为 k 的简单路径问题”,就分别用到了广度遍历和深度遍历算法。

4. 生成树、最小生成树的概念以及最小生成树的构造rim算法和kruskal

7. 最短路径问题:

最短路径问题分为两种:一是求从某一点出发到其余各点的最短路径;二是求图中每一对顶点之间的最短路径。

主要还是体现在平时做实验的时候,因为做其他课的实验最起码会知道一些基本的做法,但是遇到数据结构就会发现真的很难,有时真的什么都不会,看也看不懂,感觉很吃力,

就感觉数据结构这个模块不简单，有点复杂，总体感觉学不好，但是上次期中考试时候，发现也不是传说中的那么难，有些题真的很死，可以用固定的方法去写，但是那种题不多，大部分的题还是要自己去构造一种思想，并用代码实现它！感觉这样的题不好做，有点难度！其实，刚开始讲的时候，因为课下没有预习，上课节奏也没有跟上，所以就失去信心了。

是感觉缺少很多，所以自己还是要其他方面多多努力，接下来的复习还是要靠自己理解，如果理解好了以后对做题的帮助确实不小，所以说，理解透彻了就好办了！现在大多是自学，相比以前的学习方法，感觉有点不一样，但是我相信如果努力还会有很大的提高的。

数据结构实验总结心得篇七

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

第一章是这门学科的基础章节，从整体方面介绍了“数据结构和算法”，同时引入相关的学术概念和术语，如数据、数据元素、数据类型以及数据结构的定义。重点是数据结构的逻辑结构、存储结构和运算集合的含义及其相互联系。数据结构和两大逻辑结构的4四种常用存储方法；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。难点是算法复杂度的分析方法和性能的分析。

第二章详细地分析了顺序表。介绍了顺序表的相关概念及其有关运算。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，在各种算法思想的先分析后，要弄清各种算法的时间复杂度与空间性能的优点和缺点，在什么特定的场合适合哪种算法思想。最后介绍了顺序串的概念，顺序串是顺序表的一个特例；区别在于组成顺序串的数据元素是一组字符，其重点在于串的模式匹配。

第三章介绍链表。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，且在存储空间上有动态申请的优点。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。弄清其个运算的算法思想及其时间复杂度和空间性能。最后介绍了链表之中存储结构在实际中的相关应用。

第四章，堆栈是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；堆栈在文字处理，匹配问题和算术表达式的求值问题方面的应用。

第五章，队列是一种够类似堆栈的线性结构。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“先进先出”的规则，对堆栈的操作只能在栈顶进行；其运算有入队、出队等操作。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。其中三元组和十字链表这两种结构尤为重要；对着两种结构的建立了应用要掌握。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，其中关于二叉排序树和哈弗曼树的构建是重点。

第八章介绍了树。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。

第九章，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，对某些具体的问题和应

用仍有一些模糊与措手。各个章节出现的知识点理解和掌握情况明确一下。

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。算法的时间、空间性能分析是重点，同样也是难点，尤其是空间性能分析需要加强。在某些强大与复杂的算法面前的处理有些棘手。

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。删除方面的问题比较容易些。排序问题中，由于冒泡排序在大一c语言课上已经学习过，再来学习感觉相对轻松些。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。

第三章链表中，除对双向循环链表这一知识点理解困难之外，在对链表进行插入删除和排序相关操作上同顺序表的操作基本相当。其他的知识点像单链表的建立和基本算法等都较为熟悉。

第四章和第五章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。在一些实际问题的应用与处理方面，对其进行存储结构的选择还是需要认真考虑的。在算法的时间复杂度和空间性能的分析仍有些困难。

第六章的学习感觉较为困难的部分在于矩阵的应用上。在矩阵的存储结构中，使用三元组表发相对较为简单，而使用十字链表就有些困难了。但在某些问题的处理上又必须或从节省空间考虑采用十字链表来处理，想矩阵的加法运算。广义表的定义还是比较容易理解的，其存储结构也不难掌握，关

于应用也只局限于在多项式的表示上。

第七章是全书的重点。在这一章中概念和定义都很多，有些很昏人但都很重要，要区分开来。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。关于二叉排序树和的哈弗曼树却相对有些压力，其生成和对其关键字的插入和删除时重点。

第八章关于树的分析，首先要明确树和二叉树的区别，以及书中的相关定义和概念。关于二叉树、树和森林之间的转换和遍历方法是重点，但不算是难。接着就是数的存储结构的选择及转化为二叉树的算法，这部分有些吃力。再就介绍了特殊的树-b树，关于对b树的操作，插入关键字是中带领和难点。

第九章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。

在学习开始，王教授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！

数据结构实验总结心得篇八

计算机数据结构是计算机科学中非常重要的一部分知识，它点亮了如今互联网信息时代的每一个角落。在我学习计算机数据结构之前，我对这个课程印象模糊。但在学习过程中，我深刻认识到，数据结构不仅是一种数据组织和存储方式，而且对代码实现和算法优化也有很大的影响。在本文中，我将分享我的计算机数据结构心得体会。

第二段：知识体系和分类

计算机数据结构是指数据组织和存储的方式。根据不同的特

点和需求，数据结构可以分成线性结构、树形结构、图形结构等不同种类。常用的线性结构包含数组、链表、队列、栈等，而树形结构中常见的有二叉树、霍夫曼树等，图形结构中常用的则是邻接表、邻接矩阵等。学习计算机数据结构时，我们需要掌握各种数据结构之间的联系，并能在实际问题中选择合适的数据结构。

第三段：实现方法

在学习计算机数据结构时，除了理论知识，学习实现方法也很重要。数据结构的实现方法包括顺序存储和链式存储，不同的实现方法对代码和算法的优化都有显著的影响。顺序存储通常用来保存连续的大块数据，比如数组；而链式存储则通过指针连接各个节点，适用于大型数据的存储。学习实现方法不仅让我们能深入理解数据结构，也能提高我们的思考能力和程序设计能力。

第四段：算法优化

数据结构和算法是解决计算机问题的两个关键要素。在编写算法时，合理选择数据结构能够提高算法效率。例如，在搜索算法中，二分查找和顺序查找所使用的数据结构是数组和链表，但二分查找的效率远高于顺序查找。更进一步，对于同一数据结构，我们还可以优化算法，例如使用归并排序而不是快速排序，从而提高程序的性能。

第五段：总结

计算机数据结构是计算机科学中重要的基础知识，我们需要认真学习和掌握，才能更好地应用于实际问题中。学习数据结构既需要理解基本概念和分类，也需要掌握实现方法和算法优化。如果我们能在学习中不断总结经验和提高能力，相信我们能够在未来的工作和学习中处于更好的位置。