每题字数在100以内

1. 数据结构的主要问题是什么？（5）哪些情形导致数据结构问题的困难？（5）请按照你理解的线索整理本学期高级数据结构课程的主要内容和主要技术方法（10）

答：1）数据结构的主要问题是数据在计算机系统中的表示和组织问题。

2）数据规模、访问规模

3）Why Advanced Data Structures?

1. Data Structure and Computer Science, Principles of Data Structures—How?

Why and How O(log n) access time?

1. Dynamic data structures and Analysis
2. Randomized Data Structure
3. Augmented Data Structure
4. Data Structures in Distributed Environments

2.shell排序、冒泡排序、以及插入排序有什么本质的不同？（2）shell排序取得优势的关键是？（8）P230 shell

答：1）shell排序是不稳定的，冒泡排序（交换）和插入排序是稳定的；shell排序是的算法时间复杂度是平均O(nlogn)、最差O（(n ^2），冒泡排序和插入排序的算法时间复杂度是平均O(n ^2)。

2）Shell排序是基于插入排序的一种改进。Shell取得优势的关键是，在希尔排序开始时增量较大，分组较多，每组的记录数目少，各组内直接插入排序较快，后来增量di逐渐缩小，分组数逐渐减少，而各组的记录数目逐渐增多，但由于已经按di-1作为距离排过序，使序列较接近于有序状态，所以新的一趟排序过程也较快。

不需要大量辅助空间。

（当序列初态基本有序时插入排序所需的比较和移动次数均较少。当n值较小时，n和n^2的差别也较小，即直接插入排序的最好时间复杂度O(n）和最坏时间复杂度0(n^2）差别不大。)

3.什么情况下选用平衡二叉树、红黑树、伸展树（splay tree）或B+树（5）

答：1）AVL：二叉查找树的主要查找、插入、删除等操作的运行时间，与树的高度成正相关。在动态情况下，为了将查找的时间复杂度保证在O(logN)范围内，使用AVL。绝对平衡维护成本过高，AVL是适度平衡。平均和最坏的查找时间都是O(logN)，同时插入和删除元素也只需要O(logN)。P278

2）红黑树：当需要一个平衡二叉查找树的任何动态操作，无论插入还是删除，对树形拓扑结构的影响都能控制在常数的范围之内，使用红黑树。旋转操作非常局部化，而且次数较少（插入最多两次旋转，删除最多三次旋转），红黑树能达到O(logn)高度，高度要求更加灵活的，但是不像AVL树那样严格要求左右子树高度差必需相差不超过1。

1. *每个结点要么是红的，要么是黑的。*
2. *根结点是黑的。*
3. *每个叶结点，即空结点（NULL）是黑的。*
4. *如果一个结点是红的，那么它的子结点都是黑的。*
5. *对每个结点，从该结点到其子孙结点的所有路径上包含相同数目的黑结点。*

3）伸展树： 刚被访问的数据极可能被很快的再次访问，这要求二叉查找树可以进行自我调节，将活跃元素靠近根节点，使不活跃元素向叶结点移动。使用伸展树通过splay可以实现上述目标。P293

4）B+树： 需要范围查找时使用B+树，B树只利于单个关键字查找，而数据库等应用需要范围查找。B+树可以实现范围查找，B+树的元素只存放在叶结点。中间结点的关键字只起引导查找的作用。P307

4.chord的存储结构和跳表有什么联系（5）chord的次线性查询复杂度和路由表的次线性储存复杂度是怎么实现的？（5）

答：1）

跳表的构造过程是：1、

一个跳表，应该具有以下特征：有几个层（level）组成；跳表的第一层包含所有的元素；每一层都是一个有序的链表；如果元素x出现在第i层，则所有比i小的层都包含x；第i层的元素通过一个down指针指向下一层拥有相同值的元素；在每一层中，-1和1两个元素都出现(分别表示INT\_MIN和INT\_MAX)；Top指针指向最高层的第一个元素。

2）负载平衡：所有的节点以同等的概率分担系统负荷，从而可以避免某些节点负载过大的情况。节点之间完全平等并完成同样的工作，使得Chord具有很高的鲁棒性，當node數增加時, 搜尋的次數以對數方式成長, 不需其他參數調整，當有node加入或離開時, Chord 會自動調整內部的table, 保證key能被找到，可以使用任何名稱來當作Chord的key。

分布式链表Chord基于安全的一致性散列函数来分配节点ID和对象ID。在一个有N个节点的网路中，每个Chord节点保存O(logN)个其他节点的信息，查询数据对象需要的覆盖网络的路由跳数也是O(logN)，当节点离开或者加入网络时，为了维持网络结构，保持自适应性所需要的的消息数在O(log2N)。作为分布式的散列表，Chord具有几乎最优的路由效率，确定性的对象查询，负载均衡，高可扩展性以及良好的容错性与自适应性。

* Can二维散列表

支持基本的哈希表操作的键值对（k，v）：插入，查找，删除

每个节点是哈希表中（k，v）对子集，存储块（区）的哈希表。

每个节点存储邻区信息状态。

请求（插入，查找，删除）的一个关键是路由的中间节点使用的路由算法

不需要集中控制（完全分布式的）

每个小节点的状态是独立的系统中的节点数（可扩展）

节点可以绕过故障（容错）

Routing: 沿直线路径从源到目的地的坐标、每个节点维护一个表的IP地址和每个地方的邻居虚拟坐标区、用贪心路由邻居接近目的地

Join can：定位引导节点->引导节点提供随机节点IP地址->节点加入二维空间->存储邻区信息->邻区更新信息

* Chord
* Baton基于平衡树的结构

优势

与其他系统相比，更新路由表成本更低

支持精确匹配查询和范围查询效率

灵活高效的负载平衡方案

可扩展性（不受网络规模或ID空间前手）

5.根据你的理解，存储介质的特性对数据结构的设计有什么影响？包括内存存储、磁盘存储和分布存储的需要解决的问题、技术方法（10）

答：1）

2） 内存存储考虑空间问题、内排、AVL/RBTree/SplayTree

磁盘存储考虑交换问题、外排mergesort、BTree/B+Tree/RTree

分布存储考虑交换、传输成本问题：Can分布式散列表，Chord分布式链表，Baton分布式二/多叉树

6.如何评价数据结构的好坏？有哪些度量方法？（5）

答：1）数据的逻辑结构。是指对数据及其关系的抽象逻辑描述，对立与计算机，与机器实现无关。根据定义的关系不同，数据的逻辑结构分为四种： 集合结构。数据元素之间未定义任何关的松散集合。线性结构。数据元素之间定义了次序关系的集合（全序集合），描述的是1对1关系。 树形结构。数据元素之间定义了层次关系的集合（偏序集合），描述的是1对多关系。图状结构。数据元素之间定义了网状关系的集合，描述的是多对多关系。

数据的存储结构（亦成物理结构）是指数据结构在计算机存储器中的具体实现。 存储结构与孤立的数据元素表示形式不同，数据结构中的数据元素不但要表示其本身的实际内容，还要表示清楚数据元素之间的逻辑结构。常见的存储结构有：顺序存储结构：特点是借助于数据元素的相对存储位置来表示数据元素之间的逻辑结构；链式存储结构：特点是借助于指示数据元素地址的指针表示数据元素之间的逻辑结构。散列存储结构：顺序+算列。索引存储结构：顺序+索引。

2）

8.分摊分析的基本思想是什么？（3）请简述二项式堆插入分摊代价的分析思路？（7）

答：1）总体成本->每一次的成本（分摊复杂度）P18

对数据结构连续地实施足够多次操作，所需总体成本分摊到单词的操作。分摊复杂度和平均复杂度的结果没有必然联系，它不牵涉到概率

2）P177

9.快的打车软件需要解决的主要数据存储和访问的问题有哪些？简述你的解决方案（10）

答：1）

2）

10.简述算法和数据结构的关系（5）简单描述你所实现项目中所采用的数据结构及其实现效果（5）

Chapter 1

答：1）数据结构由一个数据对象以及该对象中的所有数据元素之间的关系组成。

数据结构与算法的联系： 程序=算法+数据结构。数据结构是算法实现的基础，数据结构的设计和选择需要为算法服务。算法的操作对象是数据结构。算法的设计和选择要同时结合数据结构，不可能独立于数据结构的。

数据结构关注的是数据的逻辑结构、存储结构以及基本操作，而算法更多的是关注如何在数据结构的基础上解决实际问题。算法是编程思想，数据结构则是这些思想的逻辑基础

2）