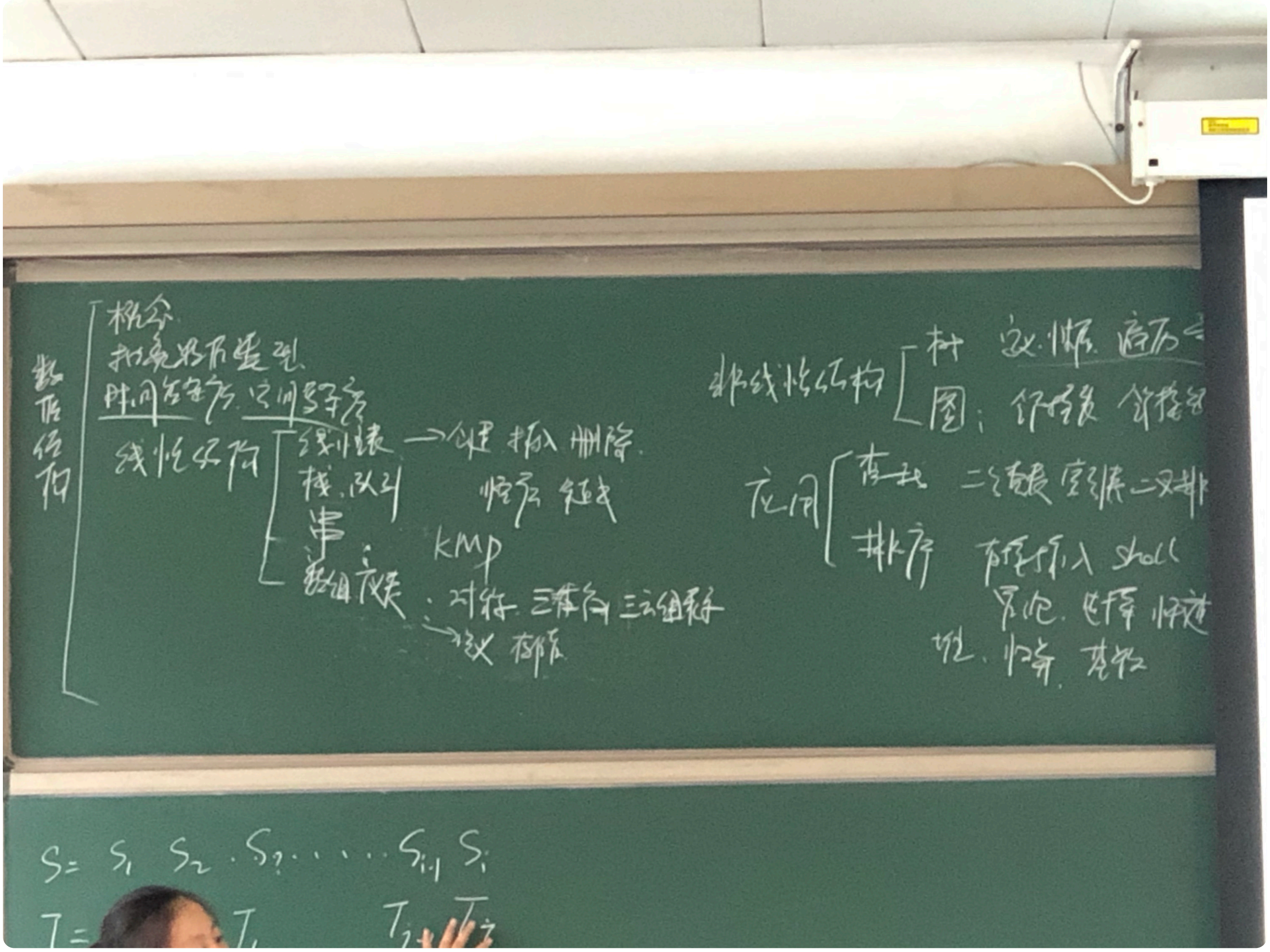

数据结构

- 概念
- 抽象数据类型
- 时间复杂度、空间复杂度
- 线性结构
  - 线性表 → 创建 插入 删除
  - 栈、队列　顺序 链式
  - 串：KMP
  - 数组广义表：对称 三角阵 三元组表
    → 定义 存储

非线性结构

- 树　定义、性质、遍历
- 图：邻接表 邻接矩

应用

- 查找　二分查找 索引表 二叉排
- 排序　直接插入 shell
  冒泡、选择 快速
  堆、归并、基数

$S = S_1\ S_2\ S_3 \cdots\cdots S_{i-1}\ S_i$

$T = T_1 \quad T_j$

第一章

1.什么是数据结构
数据结构描述现实世界实体的数学模型及其上的操作在计算机中的实现和实现
数据：信息的载体
数据元素（元素，结点，记录）：数据的基本单位，一个数据元素可以由若干数据项组成
数据项：具有独立含义的最小标识单位
数据对象：具有具体性质的数据元素的集合
4个基本结构：集合、线性结构、树形结构、网状结构
线性结构；线性表
非线性结构：树、图
数据的存储结构（物理结构）：顺序存储表示、链接存储表示、索引存储表示、散列存储表示
数据类型：一个值的集合和定义在这个值集上的一组操作的总称
2.什么是算法
为了解决某类问题而规定的一个有限长的操作序列
特性：有穷性、确定性、可行性
3.时间复杂度
运行时间=算法中每条语句执行时间之和
每条语句执行时间=该语句的执行次数（频度）✖ 语句执行一次所需时间

第二章

1.线性表概念
n个数据元素的有限序列
a1,a2..ai,ai+1..an
ai是表中数据元素，n是表长度
特点：同一线性表中元素具有相同属性。相邻数据元素之间存在序偶关系。除第一个元素外，其他每一个元素只有一个直接前驱。除最后一个元素外，其他每一个元素只有一个直接后继
2.顺序表的概念
将线性表中的元素相继存放在一个连续的存储空间中
数组
线性表的顺序存储方式
顺序访问，可以随机存取
3.顺序表的存取和存储方式
4.链表的概念
线性表的链接存储表示
（1）单链表
用一组地址任意的存储单元存放线性表中的数据元素。逻辑连续，物理地址不连续
结点构成：数据域、指针域
链式存储结构
存取方式：顺序存取
（2）静态链表
用一维数组描述线性链表
（3）循环链表
最后一个结点的指针指向头结点
存储结构：链式存储结构
（4）双向链表
指向直接前驱，数据，指向直接后继
5.链表的存取和存储方式
6.集合的交并
7.顺序表和链表的比较
（1）存储分配的方式
顺序表的存储空间是静态分配的
链表的存储空间是动态分配的
（2）存储密度=结点数据本身所占的存储量➗结点结构所占的存储总量
顺序表的存储密度=1
链表的存储密度<1
（3）存取方式
顺序表可以随机存取，也可以顺序存取
链表顺序存取
（4）插入/删除时移动元素个数
顺序表平均需要移动一半元素
链表不需要移动元素，只需要修改指针

第三章

1.栈
限定仅在表尾进行插入删除操作的线性表
后进先出
2.顺序栈
栈的顺序存储结构，利用一组地址连续的存储单元依次存放自栈底到栈顶的数据元素，top指向栈顶元素在顺序栈中的 *下一个位置*，base指向栈底的位置
空栈：top、base同时指向最底一个
进栈：b进入后，top在b上面
3.链式栈
栈的链接表示
无栈满问题，空间可扩充
插入和删除仅在栈顶处执行
链式栈的栈顶在链头
适合于多栈操作
4.栈的应用举例
（1）数值转换
N=（N div d）✖ d+N mod d
（2）行编辑程序
退格符
退行符
（3）迷宫求解
（4）表达式求值
前缀式
中缀式
后缀式
5.队列
只允许在表的一端插入，在另一端删除
允许插入的一端叫队尾rear，允许删除的一端叫队头front
出队列（队头）a1,a2..an（队尾）入队列
先进后出
6.链队列
队列的链式表示
队头指针和队尾指针
无队满问题，有队空问题
7.循环队列
队列的顺序存储表示，用一组地址连续的存储单元依次存放队头到队尾的元素
插入新元素：rear+1
删除元素；front+1
头指针始终指向队头元素，队尾指针始终指向队尾指针的后一个
（1）队头指针进1
front=（front+1）%maxsize
（2）队尾指针进1
rear=（rear+1）%maxsize
（3）队列初始化
front=rear=0
（4）队空条件
front==rear
（5）队满条件
（rear+1）%maxsize==front
队满，会空一个
8.队列的应用举例
（1）打印二项栈开始的系数
（2）优先级队列
每次从队列中取出的是具有最高优先权的元素
9.递归
若一个对象部分地包含它自己，或用它自己给自己定义，则称这个对象是递归的。若一个过程直接地或间接的调用自己，则称这个过程是递归的过程
（1）定义是递归的
求阶乘、斐波那契数列
（2）数据结构是递归的
搜索链表的最后一个结点并显示
在链表中寻找等于给定值的结点，并打印其值
（3）问题的解法是递归的
汉诺塔
（4）递归过程和递归工作栈
调用的过程中，自己调用自己，层层向下递归，退出的次序正好相反
（5）递归改成非递归
（5）n皇后问题

第四章

1.字符串
n个字符的有限序列
串名字、串值、串中字符、串的长度
2.提取子串
pos=2，len=3
3.提取第i个字符
4.串连接
5.串的模式匹配
在串中寻找子串（第一个字符）在串中的位置
词汇在模式匹配中，子串称为模式，串称为目标

第五章

1.数组
（1）一维数组
（2）二维数组
行优先存放
（3）三维数组
各个维的元素个数m1，m2，m3
下标为i1，i2，i3的数组元素的存储地址（按页、行、列存放），每个元素占L个存储单元
LOC（i1,i2,i3）=a+(i1✖m2✖m3+i2✖m3+i3）✖L
2.对称矩阵的压缩存储
上、下三角矩阵，按行存放于一个一位数组B中，称之为对称矩阵A的压缩存储方式
B中有n✖（n+1）➗2个元素
3.三对角矩阵的压缩存储
三对角矩阵有3n-2个非零元素
A[i][j]在B中的位置2i+j
4.稀疏矩阵
用三列表格表示元素
转置矩阵
5.广义表
n个表元素组成的有限序列LS=（a0，a1...an-1）
LS是表名，ai是表元素：可以是子表、数据元素（原子）
n是表的长度，n=0的广义表是空表
m>0，表的第一个元素是表头，其余都是表尾

第六章

1.数的基本概念
（1）数的定义
由n个结点的有限集合。n=0，空树。只有一个根结点。除根结点以外的其他结点划分为m个互不相交的有限集，其中每个集合本身又是一棵树，称为子树。结点。
2.树的基本术语
（1）结点
（2）结点的度
（3）子女
（4）祖先
（5）树的深度
（6）叶结点
（7）双亲
（8）子孙
（9）树的度
（10）分支结点
（11）兄弟
（12）结点层次
（13）森林
3.二叉树
至多只有2个子树，5种形态
（1）第i层上至多有2^（i-1）个结点
（2）深度为k的二叉树至多有2^k-1个结点
（3）叶子结点有n0个，度为2的结点有n2个，则n0=n2+1
（4）满二叉树
深度为k且2^k-1个结点
（5）***完全二叉树***
从右向左连续缺失若干个结点
（6）n个结点的完全二叉树的深度为1+log2（n）
（7）n个结点的完全二叉树自顶向下，每一层从左向右编号，有以下关系
1️⃣i=0，则i无双亲。i>0，则i的双亲为（i-1）/2
2️⃣若2✖i+1<n，则i的左子女为2i+1，若2i+2<n，则i的右子女为2i+2
3️⃣若结点编号i为偶数，且i！=0，则左兄弟结点i-1
4️⃣若结点编号i为奇数，且i！=n-1，则右兄弟结点为i+1
5️⃣结点i所在层次为log2（i+1）
4.二叉树的存储结构
（1）顺序表示
从上到下，从左到右
（2）链表表示
含有n个结点的二叉链表有n+1个空链域
5.二叉树的遍历
前序，中序，后序
6.二叉树遍历应用
（1）按前序建立二叉树（递归算法）
@表示空结点
（2）计算二叉树结点个数
（3）求二叉树中叶子结点的个数
（4）求二叉树高度
7.树到森林的转化
树的存储结构
（1）双亲表示：以一组连续空间存储树的结点，同时在结点中附设一个指针，存放双亲结点在链表中的位置
（2）左子女-右兄弟表示法
（3）森林与二叉树的转换
8.树的遍历
深度优先遍历：先根次序遍历，后根次序遍历
9.哈夫曼树
（1）路径长度
外部路径长度EPL+内部路径长度IPL
（2）带权路径长度wPL
叶子结点的路径长度✖结点权值
（3）哈夫曼树：WPL最小的二叉树，权值大的结点离根最近
（4）哈夫曼树的构造
最小的2个
（5）最佳判定树
（6）哈夫曼编码

第七章

1.图的概念
由顶点集合及顶点间的关系集合组成
有向图：弧头、弧尾
2.图的基本性质
（1）有向图和无向图
<x,y>,(x,y)
（2）有向完全图、无向完全图
n（n-1）条边，n（n-1）/2条边
（3）临接顶点
一条边的两个顶点
（4）子图
（5）权
边上有数
（6）顶点的度TD（v）
顶点的所有边数。有向图中，入度+出度
（7）顶点的入度
有向边的终点ID（v）
（8）顶点的出度
有向边的起点OD（v）
（9）路径
从一个顶点经过几个顶点到达终点，顶点序列为路径
（10）路径长度
非带权图：边的条数
带权图：各条边的权值之和
（11）简单路径
路径上的顶点各不相同
（12）简单回路
第一个顶点是最后一个顶点
（13）连通图、连通分量
一个顶点到另一个顶点有路径，则这两个顶点是连通的。任意两个顶点连通，就是连通图。非连通图的极大连通子图叫做连通分量
（14）强连通图、强连通分量
有向图，任意两个顶点都能互相到达，则是强连通图。非强连通图的极大强连通子图叫做强连通分量
（15）生成树
假设一个连通图有n个顶点和e条边，其中n-1条边和n个顶点构成一个极小连通子图，称该极小连通子图为此连通图的生成树
在极小连通子图增加一条边，则一定有环
在极小连通子图去掉一条边，则成为非连通图
n个顶点，n-1条边的图不一定是生成树
非连通图，各个连通分量的生成树的集合为此非连通图的生成森林
3.图的存储结构
（1）邻接矩阵
自己到自己是0
无向图的邻接矩阵是对称的，有向图的邻接矩阵可能不对称
（2）网络的邻接矩阵
自己到自己是0
不到达是∞
到达是权值
（3）邻接表
链式存储结构
无向图的邻接表
有向图的邻接表和逆邻接表
带权图的邻接表
4.图的遍历
从图中某一顶点出发遍历图中所有的顶点，且每个顶点仅被访问一次
（1）深度优先搜索DFS
深度优先生成树
（2）广度优先搜索BFS
广度优先生成树，分层
5.图的连通性问题
（1）连通分量
（2）重连通分量
1️⃣关节点：删去v和它的所有边，将图分割成多个连通分量
2️⃣没有关节点：重连通图
3️⃣重连通图上，任何一对顶点之间至少存在2条路径，删去某个顶点和它的边，不破坏连通性
4️⃣一个连通图如果不是重连通图，它就包含几个重连通分量
5️⃣连通图，重连通分量可以用深度优先生成树找到
6️⃣在图中各顶点旁表明的深度优先数，给出进行深度优先搜索时各顶点的访问的次序
7️⃣深度优先生成树的根是关节点的充分必要条件是它至少有2个子女
8️⃣其他顶点u时关节点的充要条件时它至少有一个子女w，从w出发，不能通过w、w的子孙及一条回边所组成的路径到达u的祖先
（3）最小生成树
不同的遍历得到不同的生成树
最小生成树：n-1个边，n个顶点。不能产生回路。权值最小

普雷姆算法
从某一顶点出发，选择与它关联的最小权值的边，将其加入到生成树中
初始数组，存权变-1
6.AOV图
顶点表示活动，有向边表示先后顺序
AOV网络不能出现有向回路
（1）拓扑排序
在AOV网络中选一个没有直接前驱的顶点（无入度），并输出
删去该顶点和它的边
AOV网络和邻接表
序号 入度 自己信息 出度
7.AOE网络
无有向环的带权有向图
边表示活动，边的权值是活动持续时间，顶点表示事件
路径最长的路径：关键路径
只有各条路径上的所有活动都完成了，整个工程才算完成，完成整个工程所需的时间取决于从源点到汇点的最长路径
关键路径上的活动都是关键活动
（1）事件的最早开始时间Ve（i）
源点到该顶点的最长路径长度
（2）事件的最迟允许时间Vl（i）
保证汇点完成的情况下
（3）活动的最早开始时间e[k]
最长路径长度
（4）活动的最迟允许开始时间l[k]
在不会引起时间延误的前提下，活动允许的最迟开始时间
（5）时间余量l[k]-e[k]
相等就是关键路径
8.最短路径
如果从图中某一顶点到达另一顶点的路径可能不止一条，如何找到一条路径使得沿此路径上各边上的权值总和达到最小

第九章

1.查找
查询某个元素是否在查找表中，检索某个元素的各种属性，在查找表中插入一个元素，在查找表中删去某个元素
2.查找表的分类
静态查找表：仅查询和检索
动态查找表：插入或删除
3.关键字
元素的某个数据项的值，用以标识一个元素。主关键字：识别唯一一个记录。次关键字：识别若干记录
4.查找方法评价
（1）查找速度
（2）占用存储空间多少
（3）算法本身复杂程度
（4）平均查找长度ASL
5.顺序表的查找
顺序存储结构、链式
从表的一端逐个比较
平均查找长度=（n+1）/2
6.有序表的查找
顺序存储结构
折半查找
每次将待查区间缩小一半
采用顺序存储结构的有序表
平均查找长度=log2（n+1）-1
折半查找效率比顺序查找高，但只能有序表
7.索引顺序表（分块查找）
顺序存储结构、链式
分组，记录每组的最大值
块间有序，块内无序
平均查找长度：在顺序查找、折半查找之间
8.几种查找表的特性
查找性能最好：o(log n)，有序表
插入和删除的性能最好：o(1),链表
9.二叉排序树
左子树<根结点<右子树
10.平衡二叉树
每个节点的左右子树深度之差绝对值不大于1
左左->右
右右->左
左右->左右
右左->右左
11.哈希表
（1）线性探测再散列
（2）二次探测再散列
（3）随机探测再散列
（4）链地址法
12.哈希表的平均查找长度是a的函数，不是n的，要选一个适合的装填因子

第十章

1.排序
将一个元素的任意序列，重新排列成一个按关键字有序的序列
2.数据表
待排序数据对象的有限集合
3.主关键字
用一个属性可以区分对象
4.排序的稳定性
相同，排在前面的在排序后也在前面
5.内排序和外排序
内排序：在排序期间，数据对象全部存放在内存的排序
外排序：对象个数太多，不能同时存放在内存，必须根据排序过程的要求，不断在内外存之间移动的排序
6.排序的时间开销
衡量算法好坏的最重要标志。用执行中的数据比较次数与数据移动次数来衡量
7.内排序分类
（1）不同原则
插入排序，交换排序，选择排序，归并排序，基数排序
（2）所需工作量
简单排序 o（n^2）
先进排序方法 o（n logn）
基数排序 o（d.n）
8.排序过程的基本操作
比较2个关键字的大小，从一个位置移动到另一个位置
9.待排序记录序列的存储方式
静态链表：链表排序
10.插入排序
找到插入位置插入，原来位置上的对象向后移动
最好情况下，总比较次数n-1，时间复杂度o（n^2）
最坏情况下，总比较次数n（n-1）/2，总移动次数（n+4）（n-1）/2，时间复杂度o（n^2）
直接插入排序是稳定的排序
11.折半插入排序
有序排列，再插
比直接插入排序快，总比较次数n✖log2n
次数在直接插入排序的2个极端次数的中间
如果初始情况是接近有序的，直接插入排序的比较次数更少，2个方式的移动次数一样
折半插入是稳定的，时间复杂度o（n^2）
12.希尔排序
先将待排序的分割成若干子序列进行直接插入排序，待整个序列中的记录有序时，再对全体记录进行一次直接插入排序
不稳定排序方法
13.冒泡排序
每一趟把最大的放到最下面
最好的情况：只做一次，做n-1次比较，不移动
最坏的情况：n-1趟起泡，第i趟做n-i次排序码比较，执行n-i次对象交换
比较次数KCN=n（n-1）/2
移动次数RMN=3n（n-1）/2
起泡排序是一个稳定的排序方法，时间复杂度o（n^2）
14.快速排序
两个指针low high，low在第一个，high在最后一个，设关键字pivotkey，从high向前找到第一个小于pivotkey的，与pivotkey交换；然后从low向后找到第一个大于pivotkey的，与pivotkey交换。循环往复直到low=high