《数据结构与算法》复习

2022年春季

屈卫兰

目录

课程成绩评定

平时成绩1

- 在线视频观看5分+课后作业5分 10%
- 线上单元测试（5次+5次） 10%
- 小班讨论（8次） 10%
- 课程实验 (8个实验) 20%

平时成绩2

- 期中测试（闭卷，集中阅卷） 10%
- 期末考试（闭卷，集中阅卷） 40%

特别提醒

平时成绩将会尽快公示，如果对每项分数有异议，本课程17周周三前申述，过期不再修改，视为自愿放弃。

目录

期末闭卷考试题型和分数

应用题 (7大题) 70%

- ✓ 线性结构
- ✓ 树结构
- ✓ 图结构
- ✓ 查找
- ✓ 排序
- ✓ 算法设计技术

算法设计题（2大题） 30%

目录

绪论

- 抽象数据类型
- 数据结构
- 算法的代价
  - 时间代价
  - 空间代价

- 渐近算法分析
- 算法的最佳情况、最差情况和平均情况
- 上限、下限
- 大O、大Ω和大Θ表示法
- 定义估算法
- 化简法则
- 递归算法的分析

概念 Concept

设计 Design

实现 Implement

应用 Application

数据结构部分-线性表、树和图

- 线性结构
- 线性表
- 线性表ADT
- 线性表的物理实现
- 线性表的应用
- 栈
- 栈的ADT
- 栈的物理实现
- 栈的应用
- 队列
- 队列的ADT
- 队列的物理实现
- 队列的应用

- 树型结构
- 树结构的性质
- 树和二叉树的ADT
- 树的物理实现
- 二叉树的遍历算法
- 二叉树遍历算法的应用
- BST树的特性
- BST的实现
- 堆树的特性
- 堆树的实现
- 哈夫曼编码树的特性
- 哈夫曼编码树的实现
- 线索化二叉树和根树

- 网状结构
- 图
- 图ADT
- 图的物理实现
- 图的遍历算法
- 拓扑排序
- 关键路径
- 最短路径算法
- 最小生成树算法

概念 Concept

设计 Design

实现 Implement

应用 Application

算法部分-查找和排序部分

- 查找
- 顺序查找
- 二分查找
- 自组织线性表
- 集合检索
- 分块检索
- **BST的查找**
- **AVL**
- **B**
- 字典树
- 散列
- 散列函数
- 冲突解决策略

- 排序
- 冒泡排序
- 选择排序
- 插入排序
- **Shell排序**
- 快速排序
- 归并排序
- 堆排序
- 基数排序
- 排序问题的下限

概念 ***Concept***

设计 ***Design***

实现 ***Implement***

应用 ***Application***

算法部分-五大经典算法

- 分治
- 分治法特征
- 时间复杂度划分
- 二分搜索
- 归并排序
- 大整数乘法
- 最近点对问题
- 动态规划
- 动态规划法特征
- 0-1背包问题
- 最长公共子序列

- 贪心

贪心法特征

部分背包问题

装载问题

哈夫曼编码

- 回溯

回溯法特征

0-1背包问题

旅行推销商问题

装载问题

- 分支限界

分支限界法特征

0-1背包问题

单源最短路径问题

装载问题

概念 ***Concept***

设计 ***Design***

实现 ***Implement***

应用 ***Application***

算法分析—能力点小结

- 理解算法复杂度符号的含义，能够专业的阐述其内涵
  - 算法复杂度、增长率、T(n)
  - O、 Ω、 Θ

- **掌握基本的算法分析方法，能够分析简单算法的性能**
  1. 求T(n), 使用化简规则求（时间）复杂度
  2. 使用定义法分析（时间）复杂度

黑色是基本能力
红色是提高能力

绪论—能力点小结

- 数据结构的含义
- 抽象数据类型的含义

黑色是基本能力

红色是提高能力

线性结构—能力点小结

- 线性表、栈和队列特点
- 线性表、栈和队列的两种物理实现方式（顺序表和链表）的特点

- 线性表、栈和队列ADT的设计和表示
- 基于线性表、栈和队列ADT设计简单问题的算法

- 基于线性表、栈和队列ADT设计较复杂问题的算法
- 基于顺序表实现线性表、栈和队列
- 基于链表实现线性表、栈和队列

黑色是基本能力
红色是提高能力

树型结构—能力点小结

- **树的特点和性质**
- 树的物理存储方式
- 二叉树的遍历

- 树的动态左子结点/右兄弟结点表示法转换
- **根据给定的二叉树写出层次、前序、中序和后序遍历序列**
- **根据二叉树的两种遍历序列，构造出对应的二叉树**
- **二叉树的遍历算法实现**
- **基于二叉树递归或遍历的思想设计简单问题的算法**
- 基于二叉树递归或遍历的思想设计较复杂问题的算法
- 二叉树的基本操作的实现
- 二叉树的构建算法

黑色是基本能力
红色是提高能力

特殊树结构—能力点小结

- BST树特征
- 堆树特征
- Huffman编码树特征
- 并查集和根树特征
- 线索化二叉树

- **BST的构建方法**
- BST的递归和迭代实现方法
- **堆的构建方法**
- 堆的实现方法
- **Huffman编码树的构建方法，以及编码**
- 利用UNION/FIND算法解决等价类问题
- 二叉树的线索化

黑色是基本能力
红色是提高能力

图结构—能力点小结

- 图的特点和性质
- 图的物理存储方式邻接矩阵或邻接表特点
- 图的遍历算法思想和性能

- 根据图构造邻接矩阵或邻接表（邻接矩阵或邻接表构造图）
- 基于邻接矩阵或邻接表实现图的基本操作
- 图的深度优先搜索序列
- 图的广度优先搜索序列
- 根据给定的图构造顶点的拓扑序列
- 根据给定的图求关键路径
- 基于数据结构设计图遍历伪代码
- 应用图的遍历思路设计图的简单问题算法

黑色是基本能力
红色是提高能力

经典图问题—能力点小结

- 最短路径问题的内涵
- 经典图的最短路径问题算法思想和特点
- 最小支撑树的内涵
- 经典图的最小支撑树问题算法思想和特点

- **用Dijkstra算法求单源最短路径**
- 用Floyd算法求每对顶点间的最短路径
- **按照Prim算法构造图的最小支撑树**
- **按照Kruskal算法构造图的最小支撑树**
- Dijkstra算法伪代码和Floyd算法伪代码
- Prim算法伪代码和Kruskal算法伪代码
- 应用图的最短路径问题求解思路设计图的相关问题算法
- 应用图的最小支撑树问题求解思路设计图的相关问题算法

黑色是基本能力
红色是提高能力

查找—能力点小结

- 查找概述
- 各类经典查找方法的思路和特点、性能

- 顺序查找的实现
- 二分查找的实现
- 自组织线性表的实现
- BST树的查找过程和性能计算（ASL）
- AVL树的构建
- 散列表的构建
- 在散列表中插入、检索和删除记录和性能计算

黑色是基本能力

红色是提高能力

排序—能力点小结

- 排序概述
- 各类排序方法的思路和特点、性能
  - 插入排序、冒泡排序和选择排序
  - Shell排序、快速排序、归并排序、堆排序
  - 分配排序和基数排序

- **给出一组关键码，写出按某种排序算法进行排序的每一趟的结果**
- **简单和经典排序方法的伪代码**
- 应用排序的思想设计（优化）相关问题算法

黑色是基本能力
红色是提高能力

算法设计—能力点小结

- 五大算法概述
- 五大算法的思想和特点、性能
  - 分治
  - 动态规划
  - 贪心
  - 回溯
  - 分支限界

- **五大算法的应用：根据分治的思想计算时间复杂度；构造动态规划的递推表；给出简单问题的贪心求解过程；构造简单问题的子集树和排列树和回溯求解结果；构造简单问题的子集树和排列树和队列和优先队列求解结果；**
- 根据给定的问题给出算法设计思想
- 基于五大算法设计问题的伪代码
- 应用五大算法思路设计简单问题算法

黑色是基本能力
红色是提高能力

目录

2021年期末考题

二、（10 分）已知操作数栈 S_1 和操作符栈 S_2 如下图所示（见下页），其中 S_1 的栈顶元素为 2，S_2 的栈顶元素为+，栈底的$为结束标记。函数 EXP（）的作用如下：

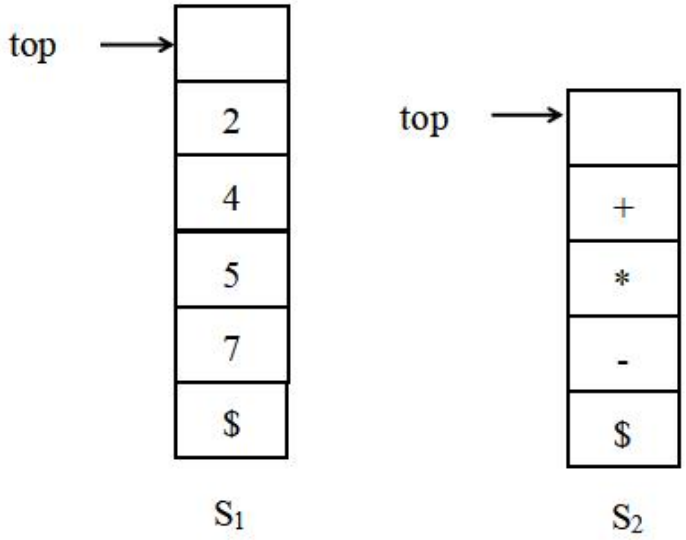
① 从栈 S_1 中依次弹栈 2 次分别存储在 $OPRD_2$ 和 $OPRD_1$（第一次弹栈存储在 $OPRD_2$，第二次弹栈存储在 $OPRD_1$）；

② 从栈 S_2 中弹栈一次存储到 OP；

③ 执行指令 $OPRD_1$ OP $OPRD_2$，把运算结果压入 S_1。

（1）请参照下图画出第一次调用 EXP（）后的两个栈 S_1 和 S_2 的示意图；

（2）反复调用 EXP（）直至栈 S_1 中只剩 1 个操作数和结束标记$，画出此时栈 S_1 的示意图。



2021年期末考题

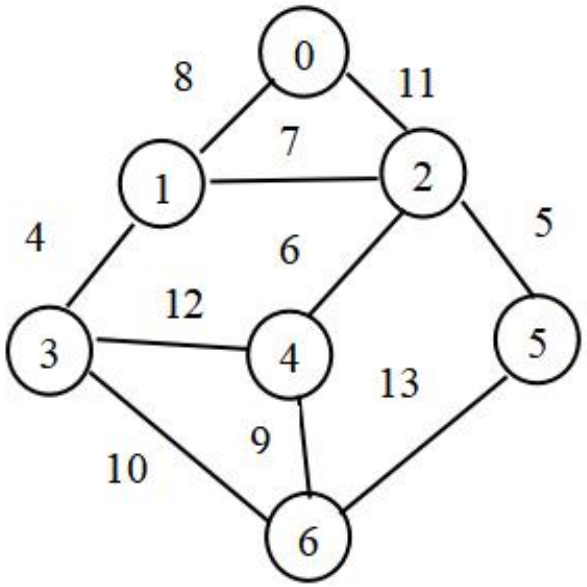
三、（10 分）

（1）设哈夫曼编码的长度不超过 4，若已经对两个字符编码为 0 和 10，则最多还可以对其它多少个字符编码，请结合画图说明理由。

（2）假设一段报文由字符集{A，B，C，D，E，F}上的字母构成，这 6 个字母在报文中出现的频率分别是 0.12，0.25，0.18，0.15，0.26，0.04，请为这几个字母设计哈夫曼编码，要求画出最终的哈夫曼树并给出每个字符的哈夫曼编码。

2021年期末考题

四、（10 分）已知几个城市之间的交通代价图——带权无向图 G，如下所示。



（1）写出从顶点 0 出发调用 BFS 的广度优先搜索树；

（2）画出用 Kruskal 算法构造最小生成树将这几个城市连通的过程。

2021年期末考题

五、（10 分）已知数据序列 20，15，32，25，21，26

（1）画出依次插入该序列后得到的二叉查找树（BST）（**不要求过程**）；

（2）分别画出依次插入该序列的平衡二叉树（**要求有过程**）。

六、（15 分）给定 n 个村庄之间的交通图（注：该图为无向图），其中村庄被抽象为图中的顶点，村庄 i 与村庄 j 之间的道路被抽象为图中的边，该边上的权值 w_{ij} 表示这条道路的长度。现打算在这 n 个村庄中选定一个村庄建一所小学。该小学应该建立在最中心的村庄里，这样使得距离小学最远的村庄到小学的路径长度 D 最短，(即如果选择小学建立在另外一个非最中心的村庄，则距离小学最远的村庄到小学的路径长度 D’ 将大于 D)。请设计一个算法求出该小学应建在哪个村庄才能使距离小学最远的村庄到小学的路径长度最短。要求：

（1）给出算法的基本设计思想；

（2）根据设计思想，采用伪码描述算法，关键之处给出注释；

（3）分析算法的时间复杂度。

2021年期末考题

七、（10 分）给定关键字集合{13，3，17，2，14，6，10，21}，假定一个散列表有 11 个槽，散列函数为 H(key)=key MOD 11。请回答下列问题：

（1）如果发生冲突后采用探查序列为：（$H(key)+i^2$）MOD 11（i=1,2,3...），请画出依次插入上述所有关键字后的散列表。

（2）求在等概率情况下，上述散列表检索成功的平均检索长度。

八、（10 分）给定一个整数序列：19，8，25，97，30，3，7，20，需对该序列按**升序**排序。请回答：

（1）采用堆排序对其进行排序，请画出序列初始化后的堆结构和第一次删除堆顶最大值元素调整后的堆结构。

（2）在我们已学排序算法中，当遇到待排序列中的记录“基本有序或者 n 值较小”时，一般采用什么排序算法？

2021年期末考题

九、（15 分）城市园林部门需要在道路沿线安装自动喷淋系统，设一条笔直道路沿线的树木种植位置为 x_1，x_2，……x_n（例如：输入的树木种植位置为 2，3，4，5，-2，6……），假定每个灌溉喷头的有效喷洒直径为 k 米，请设计一个**贪心算法**，计算至少要安装多少个喷头才能实现给这 n 棵树木自动喷淋。

注：树木之间的间距可能大于 k。

（1）给出算法的基本设计思想；

（2）根据设计思想，采用伪码描述算法，关键之处给出注释；

（3）分析算法的时间复杂度。

本题是一个区间完全覆盖问题。贪心策略是：每个喷头覆盖尽可能多的点。

由于每棵树都必须在喷头覆盖范围之内，这里将所有的树木种植位置按从小到大排序，相当于树木位于一条x轴上。接着，从左往右，每次以没有被覆盖的树的坐标为最左端，按喷洒直径计算喷头个数，这样覆盖的范围最大。

```
int greedy（vector<int> x, int k)
{
    int sum=1, n=x.size();
    sort(x.begin(), x.end());    //对树的种植位置从小到大排序
    for（int i=1，temp=x[0]; i<n; i++）
        if (x[i]-temp>k)
        {
            sum++;
            temp=x[i];
        }
    return sum;
}
```

算法分析：时间复杂度为O(nlogn)（算法复杂度取决于为排序的复杂度，如果用的是简单排序，则复杂度是n^2)

目录