## 教学计划编制问题

任务：针对计算机专业本科课程，根据课程之间的依赖关系（如离散数学应在数据结构之前开设）制定课程安排计划，并满足各学期课程总课时大致相同并且考试（除考查外）科目不大于3门。

要求：（1）可实现文件存储（2）要求计划安排合理。

## 散列法的实验研究

任务：散列法中，散列函数构造方法多种多样，同时对于同一散列函数解决冲突的方法也可以不同。两者是影响查询算法性能的关键因素。（1）对于几种典型的散列函数构造方法，做实验观察，不同的解决冲突方法对查询性能的影响。（2）实现一种允许并发操作的散列表

要求：（1）显示执行过程（2）演示并发执行情况。

## 括号匹配的检验

任务：假设表达式中允许有两种括号：圆括号和方括号，其嵌套的顺序随意，即（（）[ ]）或[（[ ] [ ]）]等为正确格式，[（ ]）或（（（]均为不正确的格式。检验括号是否匹配的方法可用“期待的紧迫程度”这个概念来描述。例如：考虑下列的括号序列：

[　(　[　]　[　]　)　]

1　2　3　4　5　6　7　8

当计算机接受了第1个括号以后，他期待着与其匹配的第8个括号的出现，然而等来的却是第2个括号，此时第1个括号“[”只能暂时靠边，而迫切等待与第2个括号相匹配的 第7个括号“）”的出现，类似的，因只等来了第3个括号“[”，此时，其期待的紧迫程度较第2个括号更紧迫，则第2个括号只能靠边，让位于第3个括号，显然第3个括号的期待紧迫程度高于第2个括号，而第2个括号的期待紧迫程度高于第1个括号；在接受了第4个括号之后，第3个括号的期待得到了满足，消解之后，第2个括号的期待匹配就成了最急迫的任务了，…… ，依次类推。可见这个处理过程正好和栈的特点相吻合。

要求：(1)读入圆括号和方括号的任意序列，输出“匹配”或“此串括号匹配不合法”。 (2)增加大括号的情况。(3) 动态演示栈的变化情况（4）使用顺序和链式存储两种结构实现。

## 哈夫曼编码/译码系统（树应用）

任务：利用哈夫曼编码进行通信，可以压缩通信的数据量，提高传输效率，缩短信息的传输时间，还有一定的保密性。现在要求编写程序模拟传输过程，实现在发送前将要发送的字符信息进行编码，然后进行发送，接收后将传来的数据进行译码，即将信息还原成发送前的字符信息。

要求：（1）在本例中设置发送者和接受者两个功能，发送者的功能包括：①输入待传送的字符信息；②统计字符信息中出现的字符种类数和各字符出现的次数（频率）；②根据字符的种类数和各自出现的次数建立哈夫曼树；③利用以上哈夫曼树求出各字符的哈夫曼编码；④将字符信息转换成对应的编码信息进行传送。接受者的功能包括：①接收发送者传送来的编码信息；②利用上述哈夫曼树对编码信息进行翻译，即将编码信息还原成发送前的字符信息。（2）传输使用两种以上加密方式。

## 活期储蓄帐目管理

任务：活期储蓄处理中，储户开户、销户、存入、支出活动频繁，设计程序，完成上述功能。要求：1) 能比较迅速地找到储户的帐户，以实现存款、取款记账；2) 能比较简单，迅速地实现插入和删除，以实现开户和销户的需要。

## 二叉排序树的实现

任务：用顺序和二叉链表作存储结构 ，完成学生成绩管理。

要求：1) 以回车('\n')为输入结束标志,输入数列L，生成一棵二叉排序树T；2) 对二叉排序树T作中序遍历，输出结果；3) 输入元素x,查找二叉排序树T,若存在含x的结点,则删除该结点,并作中序遍历(执行操作2)；否则输出信息“无x”；4）每个学生的成绩包括4个科目，学生的信息还包括学号，姓名，年龄，专业，班级。

## 最小生成树问题

任务：在n个城市之间建设网络，只需保证连通即可，求最经济的架设方法。

要求：（1）存储结构采用多种（2）求解算法多种。（3）显示求解过程。

## 通讯录的制作

任务：编写一个通讯录管理系统。本系统应完成一下几方面的功能：1) 输入信息enter();2) 显示信息display( );3) 查找以姓名作为关键字search( );4) 删除信息delete( );5) 存盘save ( );6) 装入load( ) ;

要求：1) 每条信息至包含 ：姓名（NAME ）街道（STREET）城市（CITY）邮编（EIP）国家（STATE）几项；2) 作为一个完整的系统，应具有友好的界面和较强的容错能力；3) 用《数据结构》中的双向链表作数据结构、单链表结构、顺序结构、二叉平衡树分别实现。

## 长途电话区号编码/译码器

任务：设计一个利用哈夫曼算法的编码和译码系统，长途电话区号编码/译码器。

要求：1) 将权值数据(根据人口决定)存放在数据文件(文件名为data.txt，位于执行程序的当前目录中) ；2) 分别采用动态和静态存储结构；3) 初始化：键盘输入字符集大小n、n个字符和n个权值，建立哈夫曼树；4) 编码：利用建好的哈夫曼树生成哈夫曼编码；5) 输出编码；6) 译码功能；7) 显示哈夫曼树；

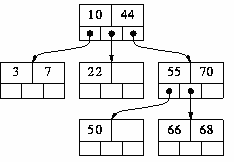
## 散列表的设计与实现

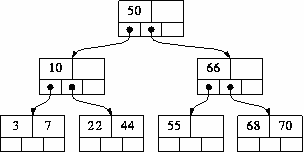
任务：设计散列表实现电话号码查找系统。

要求：1) 设每个记录有下列数据项：电话号码、用户名、地址；2) 从键盘输入各记录，分别以电话号码和用户名为关键字建立散列表；3) 采用散列函数和多种的方法解决冲突，比较冲突率和平均查找长度；4) 查找并显示给定电话号码的记录；5) 查找并显示给定用户名的记录。

## B-Trees 的实现及分析

任务：B-Trees 是一类满足特殊条件的 M 路查找树。首先说明 M 路查找树，M 路查找树是二元查找树的一般化，其结构如下图所示的 3 路查找树：M 路查找树中的任一结点至多存放M-1 个数据，并至多拥有M 棵子树；每个结点中的数据按升序排列V1 < V2 < ... Vk (k <= M-1)，每个数据 Vi 都存在一棵左子树和一棵右子树，如果左子树不空的话，该子树中所有结点的值都小于 Vi,如果右子树不空的话，该子树中所有结点的值都大于Vi。



B-Trees 是满足如下两个条件的 M 路查找树：① 所有叶结点的高度相同。② 除根之外的所有结点都至少是半满的，即该结点包含 M/2 或更多的值。下图是一个 B-树的实例。

要求:实现在 B-树上的查找，并分析其时间复杂性。实现 B-树的基本操作。实现基本操作的演示。

## 一元多项式的加法、减法、乘法的实现。

任务：设有一元多项式Am(x)和Bn(x)。Am(x)=A0+A1x1+A2x2+A3x3+… +Amxm； Bn(x)=B0+B1x1+B2x2+B3x3+… +Bnxn； 实现求M(x)= Am(x)+Bn(x)、M(x)= Am(x)-Bn(x)和M(x)= Am(x)×Bn(x)。

要求：1) 首先判定多项式是否稀疏；2) 分别采用顺序和动态存储结构实现；3) 结果M(x)中无重复阶项和无零系数项；4) 要求输出结果的升幂和降幂两种排列情况

## 利用栈求表达式的值

任务：建立试题库文件，随机产生n个题目；题目涉及加减乘除，带括弧的混合运算；

要求：随时可以退出；保留历史分数，能回顾历史，给出与历史分数比较后的评价。

## 简易文本编辑器

任务：实现文本编辑。

要求：1) 考虑编辑效率；2) 查找，替换，插入（插串，文本块的插入）、块移动（行块，列块移动），删除3) 可正确存盘、取盘；4) 正确显示总行数；5）实现关键字的高亮显示。

## 二叉树、树、森林的算法

任务：实现二叉树的中序、前序、后序的递归、非递归遍历算法，层次序的非递归遍历算法的实现，应包含建树的实现。实现树、森林的各种存储与基本操作。

要求：（1）显示二叉树、树、森林。（2）显示栈或队列的变化过程（3）显示存储结构。

## 学生搭配问题

 任务：一班有m个女生,有n个男生(m不等于n),现要开一个舞会，男女生分别编号坐在舞池的两边的椅子上。每曲开始时,依次从男生和女生中各出一人配对跳舞, 女生可以拒绝与男生跳舞（采用随机数实现），被拒绝的男生回到队列尾，从男生队列选取下一男生直到该女生不拒绝。模拟动态地显示出上述过程。

要求:1) 输出每曲配对情况；2) 计算出任何一个男生(编号为X)和任意女生(编号为Y),在第K曲配对跳舞的情况，至少求出K的两个值；3) 进一步实现记录男生被拒绝次数，拒绝次数越多，优先级越低。优先级高的男生优先出队列。提示：可使用多级队列实现。

## 敢死队问题

任务：有M个敢死队员要炸掉敌人的碉堡，谁都不想去，排长决定用轮回数数的办法来决定哪个战士去执行任务。如果前一个战士没完成任务，则要再派一个战士上去。现给每个战士编一个号，大家围坐成一圈，随便从某一个战士开始计数，当数到5时，对应的战士就去执行任务，且此战士不再参加下一轮计数。如果此战士没完成任务，再从下一个战士开始数数，被数到第5时，此战士接着去执行任务。以此类推，直到任务完成为止。排长是不愿意去的，假设排长为1号，设计程序，求出从第几号战士开始计数才能让排长最后一个留下来而不去执行任务。

要求：至少采用两种不同的数据结构的方法实现。

## 猴子吃桃子问题

任务：有一群猴子摘了一堆桃子，他们每天都吃当前桃子的一半且再多吃一个，到了第10天就只余下一个桃子。用多种方法实现求出原来这群猴子共摘了多少个桃子。

 要求：1) 采用数组数据结构实现上述求解；2) 采用链数据结构实现上述求解；3) 采用递归实现上述求解；4) 动态演示过程。

## 数制转换问题

任务：任意给定一个M进制的数x ，请实现如下要求：1) 求出此数x的10进制值（用MD表示）2) 实现对x向任意的一个非M进制的数的转换。

要求：至少用两种或两种以上的方法实现上述要求（用栈解决，用数组解决，其它方法解决）。

## 学生成绩管理系统

任务：现有学生成绩信息文件1（1.txt），内容如下

姓名    学号   语文  数学   英语

张明明  01     67    78      82

李成友  02     78    91      88

张辉灿  03     68    82      56

王露    04     56    45      77

陈东明  05     67    38      47

….      ..      ..     ..       …

学生成绩信息文件2（2.txt）,内容如下:

姓名    学号   语文  数学   英语

陈果    31     57    68      82

李华明  32     88    90      68

张明东  33     48    42      56

李明国  34     50    45      87

陈道亮  35     47    58      77

….      ..      ..     ..       …

试编写一管理系统。

要求:1) 实现对两个文件数据进行合并,生成新文件3.txt；2) 抽取出三科成绩中有补考的学生并保存在一个新文件4.txt；3) 对合并后的文件3.txt中的数据按总分降序排序(至少采用两种排序方法实现)；4) 输入一个学生姓名后,能查找到此学生的信息并输出结果(至少采用两种查找方法实现，其中包括平衡二叉树)；5) 要求使用结构体,链或数组等实现上述要求。

## 图的遍历和生成树求解实现

任务：图的遍历和生成树求解实现。

要求：1) 创建一个图；2) 图的DFS,BFS的递归和非递归算法的实现；3) 最小生成树（两个算法）的实现，求连通分量的实现；4) 要求分别用邻接矩阵、邻接表、十字链表多种结构存储实现。

## 线索二叉树的应用

任务：实现线索树建立、插入、删除、恢复线索的实现。

要求：（1）给出各种遍历次序（2）实现各种遍历次序的线索化。（3）给出线索化后遍历过程。

## 稀疏矩阵应用

任务：实现三元组、十字链表、数组下的稀疏矩阵的加、减、转置、乘的实现。

要求：（1）显示三元组，十字链表（2）采用文件输入和输出（3）比较存储效率和运算效率。

## 交通咨询系统中的最短路径

任务：建立交通图的存储结构、解决单源最短路径问题、实现两个地点最短路径、最佳路线问题。

要求：（1）单源最短路径问题（2）从某一地点到另一地点的最短路径。（3）选一条最佳路线，使游客可以不重复地游览各地点，最后回到起点。

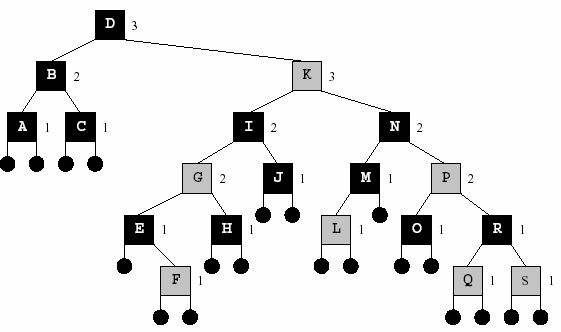
## 长整数四则运算

任务：设计一个实现任意长的整数进行加法、减法运算的演示程序。

要求：利用双向循环链表实现长整数的存储，每个结点含一个整形变量。任何整形变量的范围是 -(2^15 - 1)~ (2^15 - 1)。输入和输出形式：按中国对于长整数的表示习惯，每四位一组，组间用逗号隔开。

实现提示：（1）每个结点中可以存放的最大整数为32767，才能保证两数相加不会溢出，但若这样存放，即相当于按32768进制存放，在十进制与32768进制数之间的转换十分不方便，故可以在每个结点中仅存十进制的4位，即不超过9999的非负整数，整个链表表示为万进制。（2）可以利用头结点数据域的符号代表长整数的符号。用其绝对值表示元素结点数目。相加过程中不要破坏两个操作数链表。两操作数的头指针存于指针数组中是简化程序结构的一种方法。不能给长整数位数规定上限。

## Red-Black Tree 的实现和分析

任务：一棵具有红黑特征的树是一棵特殊的二元查找树，它在每个结点上增加一种额外属性——颜色（只能是红或黑）红黑树（Red-Black Tree）是一棵具有如下特征的二元查找树：1． 每一个结点或者是红色或者是黑色。2． 每个叶结点都是黑色。3． 如果一个结点是红色，那么它的两个孩子结点都是黑色。4．从任何一个结点到所有以该结点为子树的叶结点的简单路径上拥有相同的黑色结点。如下图就是一个红黑树的实例：一般叶结点是不存放数据的，它作为哨兵用来表示已经到达叶结点。从一个结点 x 到 x 子树中叶结点的路径（不包括 x）上黑色结点的个数称为 x 的黑色高度（black-height），标记为 bh(x)。认识 Red-Black Tree 结构，并能依据其优点解决实际问题。

要求：设计并实现 Red-Black Tree 的 ADT，该 ADT 包括 Tree 的组织存储以及其上的基本操作：包括初始化，查找，插入和删除等。并分析基本操作的时间复杂性。实现 Red-Black Tree ADT 的基本操作演示。演示实例为依次插入 A L G O R I T H M 并依同样的次序删除。

## 文字研究助手问题

任务：文字研究人员需要统计某篇英文小说中某些特定单词的出现次数和位置，试写出一个实现这一目标的文字统计系统。这称为“文学研究助手”。

算法输入：文本文件和词集。

算法输出：单词出现的次数，出现位置所在行的行号（同一行出现两次的只输出一个行号）。

要求：（1）文本串非空且以文件形式存放。（2）单词定义：用字母组成的字符序列，中间不含空格，不区分大小写。（3）待统计的单词不跨行出现，它或者从行首开始，或者前置一个空格。（4）数据结构采用二维链表，单词结点链接成一个链表，每个单词的行号组成一个链表，单词结点作为行号链表的头结点。

## 关键路径算法

任务：AOE网(即边表示活动的网络)，在某些工程估算方面非常有用。它可以使人们了解：（1）研究某个工程至少需要多少时间？（2）哪些活动是影响工程进度的关键? 在AOE网络中，从源点到汇点的有向路径可能不止一条，但只有各条路径上所有活动都完成了，这个工程才算完成。因此，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长度，即在这条路径上所有活动的持续时间之和，这条路径就叫做关键路径（critical path）。

要求：1 以某一工程为蓝本，采用图的结构表示实际的工程计划时间。2 调查并分析和预测这个工程计划每个阶段的时间。 3 用调查的结果建立AOE网，并用图的形式表示。4 用CreateGraphic ()函数建立图的邻接表存储结构，能够输入图的顶点和边的信息，并存储到相应存储结构中。5 用SearchMaxPath()函数求出最大路径，并打印出关键路径。6 编写代码并调试、测试通过。

## 表达式求值

任务：当用户输入一个合法的算术表达式后，能够返回正确的结果。能够计算的运算符包括：加、减、乘、除、括号；能够计算的操作数要求在实数范围内；对于异常表达式能给出错误提示。

要求：1首先置操作数栈为空栈，表达式起始符＃为运算符栈的栈底元素；2依次扫描表达式中每个字符，若是操作数则进OPND栈；若是运算符，则和OPTR栈的栈顶运算符比较优先权后作相应操作，直至整个表达式求值完毕。3考虑到对n位和小数点的运算。

## 平衡二叉树的判定

任务：给定一个二叉树的先序遍历或后序遍历结果，判定其是否为平衡二叉树。

要求：（1）画出二叉树（2）给出二叉树各结点的平衡因子（3）如果不为平衡二叉树，按照层次遍历次序重建一棵平衡二叉树。（4）演示平衡二叉树的查找过程。

## 停车场管理系统

任务：设停车场内只有一个可停放n辆汽车的狭长通道，且只有一个大门可供汽车进出。汽车在停车场内按车辆到达时间的先后顺序，依次由北向南排列（大门在最南端，最先到达的第一辆车停放在车场的最北端），若车场内已停满n辆汽车，则后来的汽车只能在门外的便道上等候，一旦有车开走，则排在便道上的第一辆车即可开入；当停车场内某辆车要离开时，在它之后开入的车辆必须先退出车场为它让路，待该辆车开出大门外，其它车辆再按原次序进入车场，每辆停放在车场的车在它离开停车场时必须按它停留的时间长短交纳费用。汽车可有不同种类，则它们的占地面积不同，收费标准也不同，如1辆客车和1.5辆小汽车的占地面积相同，1辆十轮卡车占地面积相当于3辆小汽车的占地面积。汽车可以直接从便道上开走，此时排在它前面的汽车要先开走让路，然后再依次排到队尾。试为停车场编制按上述要求进行管理的模拟程序。以栈模拟停车场，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。

要求：每一组输入数据包括三个数据项：汽车“到达”或“离去”信息、汽车牌照号码及到达或离去的时刻，对每一组输入数据进行操作后的输出数据为：若是车辆到达，则输出汽车在停车场内或便道上的停车位置；若是车离去，则输出汽车在停车场内停留的时间和应交纳的费用（在便道上停留的时间不收费）。栈以顺序结构实现，队列以链表实现。设置一个选项：停放在便道上的汽车也收费，收费标准比停放在停车场的车低一半，完成上述内容。

实现提示：需另设一个栈，临时停放为给要离去的汽车让路而从停车场退出来的汽车，也用顺序存储结构实现。输入数据按到达或离去的时刻有序。栈中每个元素表示一辆汽车，包含两个数据项：汽车的牌照号码和进入停车场的时刻。

## 航空客运订票系统。

任务：每条航线所涉及的信息有：终点站名、航班号、飞机号、飞机周日（星期几）、乘员定额、余票量、订定票的客户名单（包括姓名、订票量、舱位等级1，2或3）以及等候替补的客户名单（包括姓名、所需数量）。

要求：查询航线：根据客户提出的终点站名输出如下信息：航班号、飞机号、星期几飞行，最近一天航班的日期和余票额；承办订票业务：根据客户提出的要求（航班号、订票数额）查询该航班票额情况，若有余票，则为客户办理订票手续，输出座位号；若已满员或余票少余订票额，则需重新询问客户要求。若需要，可登记排队候补；承办退票业务：根据客户提出的情况（日期、航班号），为客户办理退票手续，然后查询该航班是否有人排队候补，首先询问排在第一的客户，若所退票额能满足他的要求，则为他办理订票手续，否则依次询问其它排队候补的客户。

实现提示：两个客户名单可分别由线性表和队列实现。为查找方便，已订票客户的线性表应按客户姓名有序，并且，为了插入和删除方便，应以链表作为存储结构。由于预约人数无法预计，队列也应以链表作为存储结构。

## 伙伴存储管理系统演示

任务：伙伴存储管理系统是操作系统中动态存储管理的方法。试写一个演示系统，演示分配和回收存储块前后的存储空间状态变化。

要求: 可以取m=5，即Spacesize=25。程序应不断地从终端读取整数n。每个整数是一个请求。如果n>0，则表示用户申请大小为n的空间；如果n<0，则表示归还第n次申请的块；如果n=0，则表示结束运行。每读入一个数，就处理相应的请求，并显示处理之后的系统状态，系统状态由占用表和空闲表构成。占用表显示各块的起始地址和长度，空闲表显示空闲块的始址和长度。

测试数据：

l，-1，3，4，4，4，-3，-2，2，2，2，2，-4，-6，-7，-8，-9，-5，40，0。

实现提示：数据结构如下:

typedef struct BlkHeader{

BlkHeader \*llink，\*rlink；

int tag；

int kvalue；

int blkstart； //块起始地址

}BlkHeader，\*Link；

typedef struct{

int blksize；

Link first；

}ListHeader；

主要变量是:

ListHeader avail[m+1]；//可用空间表

Link allocated；//占用表的表头指针

在这里，我们把每块的块头分离出来，通过blkstart域与相应的块建立联系。每个块一旦被分配，其块头就进入占用表，其中的各块头由rlink域链接在一起。tag域实际上不起作用。

选作内容：用直观的图示方式显示状态。

## B树的应用

MySQL支持诸多存储引擎，而各种存储引擎对索引的支持也各不相同，因此MySQL数据库支持多种索引类型，如BTree索引，哈希索引，全文索引等等。参考http://blog.codinglabs.org/articles/theory-of-mysql-index.html。实现MySQL BTree索引仿真， 并给出实验结果。

## 只用2GB内存在20亿个整数中找到出现次数最多的数

题目：有一个包含20亿个全是32位整型的大文件，在其中找到出现次数最多的数。

要求：内存限制为2GB。

思路：要找出出现次数最多的数，通常的做法是使用哈希表对出现的每一个数做词频统计，哈希表的key是某一个整数，value是这个数出现的次数。就本题来说，一共有20亿个数，哪怕只是一个数出现了20亿次，用32位的整数也可以表示其出现的次数而不会产生溢出，所以哈希表的key需要占用4B，value也是4B。那么哈希表的一条记录(key,value)需要占用8B，当哈希表记录数为2亿个时，需要至少1.6GB内存。

但如果20亿个数中不同的数超过2亿种，最极端的情况是20亿个数都不同，那么在哈希表中可能需要产生20亿条记录，这样内存会不够用，所以一次性用哈希表统计20亿个数的方法是有很大风险的。

解决方法是把20亿个数用哈希函数分成16个小文件，根据哈希函数的性质，同一种数不可能被哈希到不同的小文件上，同时每个小文件中不同的数一定不会大于2亿种，假设哈希函数足够好。然后对每一个小文件用哈希表来统计其中每种数出现的次数，这样我们就得到了16个小文件中各自出现次数最多的数，还有各自的统计次数。接下来只要选出16个小文件各自的第一名中谁出现的次数最多即可。

把一个大的集合通过哈希函数分配到多台机器中，或者分配到多个文件里，这种技巧是处理大数据面试题时常有的技巧之一。但是到底分配多少台机器、分配多少文件，在解题时一定要确定下来。可能是在与面试官沟通的过程中由面试官指定，也可能是根据具体的限制来确定，比如本题确定分成16个文件，就是根据内存限制2GB的条件来确定的。

## 40亿个非负整数中找到没出现的数

题目：32位无符号数的范围是0~4294967295，现在有一个正好包含40亿个无符号整数的文件，所以在整个范围中必然有没出现过的数。可以使用最多1GB的内存，怎么找到所有没出现过的数？

思路：若用哈希表来保存出现过的数，那么如果40亿个数都不同，则哈希表的记录数为40亿条，存一个32位整型需要4B，所以最差情况下需要40亿 \* 4B=160亿字节，大约需要16GB的空间，这是不符合要求的。