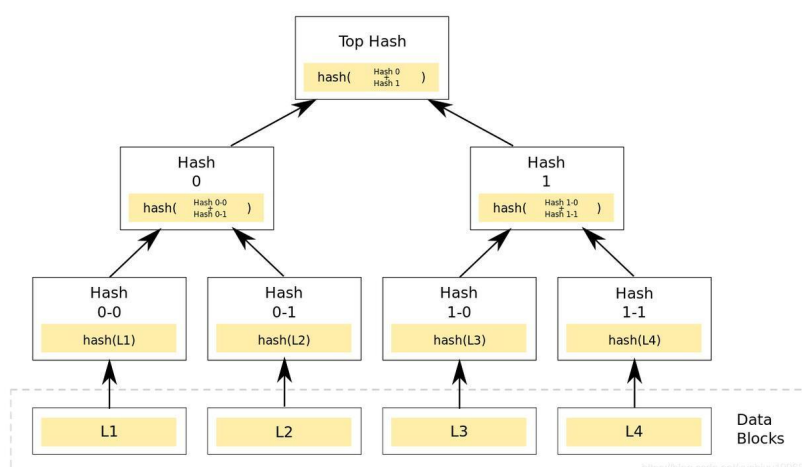


## 1、Merkel 树模拟

Merkel 树在数据安全领域应用广泛，如区块链。Merkel 树的主要思想是将数据分块进行哈希，并对哈希值进行逐层哈希，最终产生 Merkel 树根。从结构上看，Merkel 树是一棵完全二叉树，所有叶子结点为原始数据分块的哈希，均处于 Merkel 树的最底层；网上逐层为下一层的相邻两个结点的哈希，如下图所示。若原始数据分块不足  $2^n$ ，则复制最后一个数据块填充。

**实现要求：**

- (1) 给定一个文档，按段落或按行进行分块；
- (2) 使用 SHA256 算法进行构建 Merkel 树，其中上层对下层两个哈希值进行字符串拼接后再进行哈希计算；
- (3) 将原始数据分块及 Merkel 树根存储；
- (4) 修改某个数据块的值，查找修改后的值；重新计算 Merkel 树根，与之前存储的 Merkel 树根不等，则报错。



## 2、B+树的实现

B+树是 B 树的一种变形，在数据库索引中广泛应用。

m 阶 B+树有如下特征：

- (1) 每个结点的关键字个数与孩子个数相等，所有非最下层的内层结点的关键字是对应子树上的最大关键字，最下层内部结点包含了全部关键字；
- (2) 除根结点以外，每个内部结点有 到 m 个孩子；
- (3) 所有叶结点在树结构的同一层，并且不含任何信息(可看成是外部结点或查找失败的结点)，因此，树结构总是树高平衡的。

**实现要求：**

- (1) 实现 B+树，包括构建和查找，元素插入和删除；
- (2) 随机产生 10000 个数据，对构建、查找、插入、删除操作进行测试，输出查找过程的比较次数；
- (3) 使用 (2) 产生的数据构建二叉排序树，查找相同的元素并输出查找过程的比较次数；
- (4) 分析二叉排序树和 B+树的优缺点。

### 3、图的连通分量

连通性是图的重要性质，如何在非连通的图中找出连通分量在实际应用中有重要意义。设计算法实现对图的连通分量的求解，包括有向图的强连通分量和无向图连通分量。

**实现要求：**

- (1) 设计实现无向图的连通分量求解算法，并分析复杂度；
- (2) 设计实现有向图的强连通分量求解算法，并分析复杂度。

### 4、矩阵运算

矩阵运算是智能算法的基础。常用的矩阵运算包括加法、乘法、转置、求逆等。

**实现要求：**

- (1) 实现矩阵的存储、加法、乘法；
- (2) 实现矩阵的转置；
- (3) 实现矩阵的求逆矩。

### 5、博物馆打卡

博物馆是历史文化的积淀，博物馆打卡已经成为一种社会风尚。请为南京市所有博物馆建立旅游地图，并根据用户需求进行旅游路线推荐。

**实现要求：**

- (1) 以南京市所有博物馆为输入，路线按照实际的导航路线建立博物馆旅游地图，包括驾车、公共交通和步行三种交通方式，每种交通方式包括通行时间和距离；
- (2) 根据用户输入的两个博物馆推荐最优出行方式；
- (3) 根据用户的当前位置推荐下一个打卡地点；
- (4) 根据用户期望打卡的几个博物馆，推荐一条最优出行路线。

### 6、火车票务模拟

火车是当今最常用的通行方式之一，票务系统是保证火车正常运转的关键。请选择一个火车班次，模拟火车票务系统的查询、购票、退票、候补等功能。

**实现要求：**

- (1) 选择一个真实的火车班次和站点、时间等信息；
- (2) 实现发车和到站时间查询、余票查询、购票、退票、候补等功能。

### 7、朋友推荐

每个人都希望找到自己志趣相同的朋友，读书是大学生兴趣的一种外在表现。给定学校图书馆近一年的借阅记录，请帮助同学找到他志趣相同的朋友。

**实现要求：**

- (1) 编辑借书记录存入文件中；
- (2) 设计一种相似度标准用以计算同学之间志趣相同的程度，可以参考如下：

若同学 A 借书记录为{1, 2, 3, 4, 5}，同学 B 借书记录为{2, 3, 4, 5, 6}，则两人的相似度可以定义为 $|\text{借书记录交集}|/|\text{借书记录并集}|$ ，即 4/6。

(3) 从文件读入借书记录，建立“最受欢迎图书榜”和“最爱读书同学榜”，将图书按借阅次数的降序排序，将同学按借阅图书数量降序排序；

(4) 给定某位同学姓名，推荐他（她）志趣相同的同学，按相似度降序排序。

## 8、外部排序

外部排序算法是大数据时代的常用排序算法。若给定 1000 个记录，已知内存空间只能进行 100 个记录的排序，请设计排序算法实现对所有 1000 个记录的排序。

**实现要求：**

(1) 随机产生 1000 个记录（类型不限），存储到文件中，每次内排序时从文件读入内存排序；

(2) 输出排序的中间结果。

## 9、地图着色

地图着色是图论中的一个经典问题，为地图上的所有城市图上颜色，要求任意两个接壤的城市着不同颜色。

**实现要求：**

以实际的城市地图为输入，输出地图着色需要的最少颜色数量，以及详细的着色方案。

## 10、蚁族战士

蚂蚁王国新建了一批单身公寓共有  $N$  套，现有  $m$  个蚂蚁战士需要安置 ( $m \geq N$ )，国王决定采用哈希的方式进行住房分配。假设哈希函数采用  $H(\text{key}) = \text{key} \bmod N$ ，其中  $1 \leq \text{key} \leq m$  为战士编号，哈希冲突采用二次探测法解决。假设战士经常出任务，公寓随住随退，即某战士在出任务期间公寓可以自由分配给其他战士居住。

**实现要求：**

现给定套数  $N$  和战士数量  $m$ ，请随机产生  $k$  个战士并分配公寓，随机给  $x$  ( $x < K$ ) 个战士分配任务，然后查询某个战士是否住在公寓，如果在住的话给出房间号。

## 11、文字编辑

简单文字编辑主要实现对文本中给定单词和符号的统计、查找、替换和删除等功能。

**实现要求：**

(1) 从文件读入一篇简单的文章；

(2) 实现对单词数和符号数的统计、实现对单词的查找、替换和删除；

(3) 结果输出到文件中。

## 12、集合运算

集合运算是一类常用的数学运算，包括集合的并、交、差和对称差等。

**实现要求：**

(1) 输入两个集合，若集合非法提示用户；

(2) 根据用户选择进行集合的并、交、差和对称差计算；

(3) 用数组和链表两种存储结构实现，并分析算法的复杂度。

### 13、TrieTree 字典树

字典树(TrieTree)，又称单词查找树或键树，是一种树形结构，是一种哈希树的变种。典型应用是用于统计和排序大量的字符串（但不仅限于字符串），所以经常被搜索引擎系统用于文本词频统计。它的优点是：最大限度地减少无谓的字符串比较，查询效率比哈希表高。

TrieTree 的核心思想是空间换时间。利用字符串的公共前缀来降低查询时间的开销以达到提高效率的目的。它有 3 个基本性质：

- (1) 根节点不包含字符，除根节点外每一个节点都只包含一个字符。
- (2) 从根节点到某一节点，路径上经过的字符连接起来，为该节点对应的字符串。
- (3) 每个节点的所有子节点包含的字符都不相同。

**实现要求：**

- (1) 给定一篇英文文章，构建 TrieTree；
- (2) 在 TrieTree 上检索单词是否存在以及出现频率；
- (3) 统计 TrieTree 的构建和检索时间；
- (4) 用 KMP 算法实现单词的检索；
- (5) 比较 TrieTree 检索和 KMP 算法的性能。

### 14、二叉树的构建

给定二叉树的前序遍历、中序遍历和后序遍历中的一种或多种，判断是否能够唯一确定一棵二叉树。

**实现要求：**

- (1) 用户输入二叉树的前序、中序和后序的一种或多种遍历序列；
- (2) 判断是否能够唯一确定一棵二叉树；
- (3) 若能确定，则输出二叉树的所有遍历序列、叶子节点和深度；
- (4) 输出该二叉树对应的森林的遍历前序和后序遍历序列。

### 15、一元多项式运算

设有一元多项式  $A_m(x)$  和  $B_n(x)$ 。 $A_m(x)=A_0+A_1x^1+A_2x^2+A_3x^3+\dots+A_mx^m$ ， $B_n(x)=B_0+B_1x^1+B_2x^2+B_3x^3+\dots+B_nx^n$ 。求解  $M(x)=A_m(x)+B_n(x)$ 、 $N(x)=A_m(x)-B_n(x)$ 、 $P(x)=A_m(x)*B_n(x)$ 。

**实现要求：**

- (1) 分别采用顺序和链式存储结构实现；
- (2) 结果  $M(x)$ 、 $N(x)$ 、 $P(x)$  中无重复项和无零系数项；
- (3) 要求输出结果按指数的升幂和降幂两种排列情况输出。

### 16、教学计划编制

大学的每个专业都要制定教学计划。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。试在这样的前提下设计一个教学计划编制程序。

**实现要求：**

- (1) 输入参数包括：学期总数，一学期的学分上限，每门课的课程号(固定占 3 位的字母数字串)、学分和直接先修课的课程号；
  - (2) 判断是否可以在规定的学期内完成所有课程学习；
  - (3) 允许用户指定下列两种编排策略之一：一是使学生在各学期中的学习负担尽量均匀；二是使课程尽可能地集中在前几个学期中。若根据给定的条件问题无解，则给出告警信息；否则将教学计划输出到用户指定的文件中。
- 参考数据：学期总数：6；学分上限：10；该专业共开设 12 门课，课程号从 C01 到 C12，学分顺序为 2,3,4,3,2,3,4,4,7,5,2,3。课程关系如下：

课程编号	课程名称	先修课程
C1	程序设计基础	无
C2	离散数学	C1
C3	数据结构	C1, C2
C4	汇编语言	C1
C5	语言的设计和分析	C3, C4
C6	计算机原理	C11
C7	编译原理	C5, C3
C8	操作系统	C3, C6
C9	高等数学	无
C10	线性代数	C9
C11	普通物理	C9
C12	数值分析	C9, C10, C1

**17、哈夫曼编码/译码器**

给定一个英文文本，对其中的每个字母和符号进行哈夫曼编码。要求对于文本中的任意一段文字能够进行哈夫曼编码和解码。

**实现要求：**

- (1) 对输入的文字和符号进行哈夫曼编码，将每个符号的编码存储到文件；
- (2) 对文本中指定句子或段落进行编码，输出编码后的内容到文件；
- (3) 对于一段编码进行解码，如果可以正确阶码则输出原始文本，否则输出告警信息。

**18、餐厅自助叫号系统**

某餐厅由于排队人数太多，造成管理不便，拟开发自助叫号系统。根据餐桌大小分成大桌（8 人桌）、中桌（4 人桌）和小桌（2 人桌）三类的号，大桌和中桌提供拼桌服务，每桌同时就餐人数不得超过餐桌限额。每组就餐人员必须先到前台根据就餐人数取号，不超过 2 人取小桌号，不超过 4 人取中桌号，不超过 8 人取大桌号，超过 8 人需按照最少分组和最少空座的原则分桌就餐，如 10 人就餐则取一个大桌号和一个中桌号，12 人就餐则取一个大桌号和一个中桌号，14 人就餐则取两个大桌号。

每桌取号需要选择是否接受拼桌。每一类餐桌单独编号，如大桌 1 号、中桌 1 号、小桌 1 号，

每类餐桌按顺序叫号。若当前就餐的餐桌有空余餐位，且顾客同意拼桌，则在同意拼桌的等待号中选择取号时间最早且就餐人数不超过空余餐位的号，直接进入就餐。如当前大桌 3 号的 5 位顾客正在就餐且同意拼桌，则在所有等待号中选择取号时间最早的就餐人数少于 3 人的号直接进入该桌就餐。

餐厅的所有餐桌编号，小桌 1...m，中桌 m+1...n，大桌 n+1...k；

用户取号时，输入就餐人数以及是否同意拼桌，系统自动产生就餐序号；

当有空桌时，系统通知当前等待的最小同类就餐序号的顾客进入就餐，并告知餐桌号； 若某餐桌就餐完成，系统自动为当前餐桌叫号，每次叫号仅针对同类餐桌的就餐序号； 若某桌当前就餐序号愿意拼桌，且桌上有空座位，则为拼桌客户叫号，每桌最多允许两组就餐人员同时就餐；

对于不愿拼桌的顾客，每桌每次仅允许一组就餐人员就餐。

#### **实现要求：**

模拟实现取号、叫号和结账操作。

### **19、医务室模拟**

假设只有一位医生，在一段时间内随机地来几位病人；假设病人到达的时间间隔为 0~14 分钟之间的某个随机值，每个病人所需处理时间为 1~9 分钟之间的某个随机值。试用队列结构进行模拟。

#### **基本要求：**

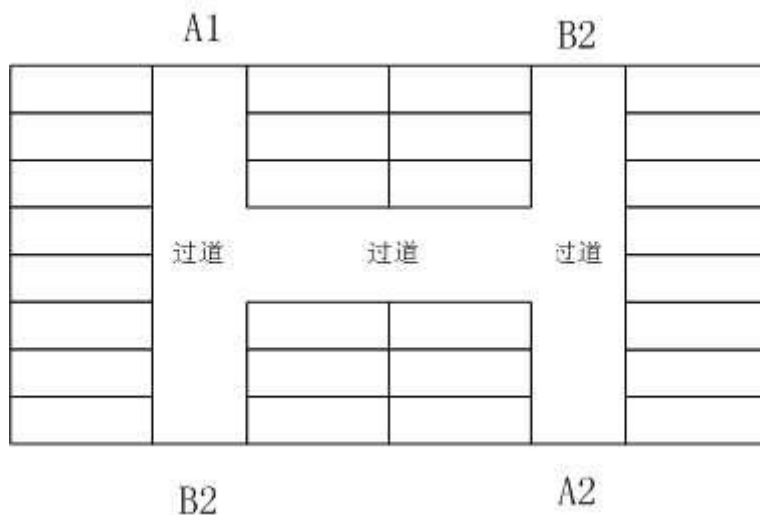
- (1) 输出每个病人的到达时间、等待时间、看病开始时间和结束时间；
- (2) 输出医生的诊治时间和等待时间；
- (3) 输出医生的总等待时间和病人的平均等待时间。

### **20、停车管理系统**

某车库有两个入口 A1 和 A2，两个出口 B1 和 B2，每个出入口每次只允许一辆车通行，如果有多辆车则排队等候，连接四个口的通道类似 H 型。停车场的最左和最右两排分别有 m 个停车位，中间通道的两个分别由 m/2-1 辆停车位组成。

#### **实现要求：**

- (1) 在入口的屏幕上显示当前车库的使用情况；
- (2) 输入用户的车牌号，可以查询该车辆停靠在哪个停车位；
- (3) 若需要停车，推荐用户从 A1 口和 A2 口进入停车场，才能保证用户等待时间最短；
- (4) 若需要离开，推荐用户从 B1 口或从 B2 口离开，才能保证用户等待时间最短；
- (5) 在出口处设计计费系统，根据停车时间收费，每小时 2 元，不足一小时则按一小时计算。



## 21、家谱管理

家谱（或称族谱）是一种以表谱形式，记载一个以血缘关系为主体的家族世系繁衍和重要人物事迹的特殊图书体裁。家谱是中国特有的文化遗产，是中华民族的三大文献（国史，地志，族谱）之一，属珍贵的人文资料，对于历史学、民俗学、人口学、社会学和经济学的深入研究，均有其不可替代的独特功能。

### 实现要求：

- （1）实现家谱的构建与存储
- （2）实现家谱的查找，包括：根据姓名查找是否在家谱中、查找给定姓名的上下三代以内亲属；
- （3）实现家谱的合并，将某一支家谱并入另一支家谱中作为某个分支；
- （4）实现家谱的删除，将某人或某一支从家谱中删除。

## 22、浮点数表示

IEEE754 标准是计算机中常用的浮点数表示方法，将浮点数按照“数符、阶码、尾数”的顺序用二进制存储，均为二进制表示。其中，数符为浮点数的正负符号，0 为正、1 为负；尾数为规格化原码表示后取出隐藏位；阶码为变形补码。IEEE754 标准单精度浮点数，数符为 1 位、阶码为 8 位、尾数位 23 位。

### 实现要求：

- （1）给定十进制数据，转换为二进制；
- （2）将（1）中二进制数表示为 IEEE754 标准单精度格式；
- （3）将（2）中 IEEE754 形式表示以 16 进制数输出；
- （4）给定十六进制数表示的 IEEE754 单精度表示形式，转换为十进制数。

提示：先把十进制数转换为二进制，符号位即为 IEEE754 标注形式的数符，0 正 1 负；对二进制数移位，表示为  $0.1XXX \times 2^k$  形式，数据右移一位则阶码+1，数据左移一位则阶码-1；将尾数  $0.1XXX$  保留 XXX 即为 IEEE754 标准形式的尾数，不足 23 位则后面补 0；阶码  $k+127$  并转换成二进制形式即为 IEEE754 标准形式的阶码。