

## 第二章 线性表

### 1、构造合适的算法，删除重复元素。

(1) 假设顺序表 L 中的元素按从小到大的次序排列，编写算法删除顺序表中“多余”的数据元素，即操作之后的顺序表中所有元素的值都不相同，**要求时间尽可能的少**；并对顺序表 A (1, 1, 2, 2, 2, 3, 4, 5, 5, 5, 6, 6, 7, 7, 8, 8, 8, 9) 模拟执行本算法，并**统计移动元素的次数**。

(2) 已知带头结点的单链表 L 中的元素以值非递减有序排列，写一高效算法删除表中所有值相同的多余元素，使得操作后的单链表中所有元素的值均不相同。

### 2、设计算法，完成相应交换功能。

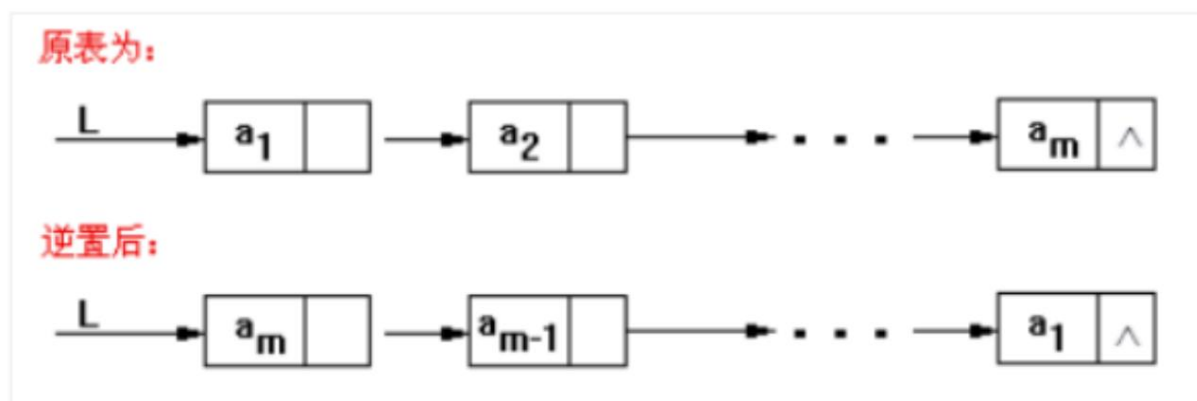
试设计一个算法，用尽可能少的辅助空间将顺序表中**前 m 个**元素和**后 n 个**元素进行互换，即将线性表 (a1,a2,.....am,b1,b2.....bn) 改变成 (b1,b2.....bn,a1,a2,.....am)。假设函数 **Invert(S,i,j)** 作用是将**数组 S 中下标自 i 到 j 的元素逆置**，请**调用该函数**完成该题算法。

```
typedef struct{
    ElemType elem[MAXSIZE];
    int listlen;
}SqList;

void exchange ( SqList &A,int m )
{ // 本算法实现顺序表中前 m 个元素和后 n 个元素的互换
}
```

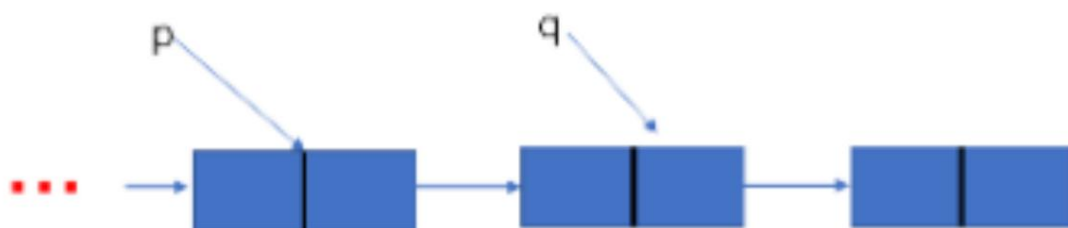
### 1、 线性表逆置算法

- (1) 已知线性表  $L = (a_1, a_2, \dots, a_n)$  顺序存储，编写算法将  $(a_1, a_2, \dots, a_n)$  逆置为  $(a_n, a_{n-1}, \dots, a_1)$ 。
- (2) 设计算法实现带头结点的单链表  $L$  就地逆置。
- (3) 设计算法将不带头结点的单链表  $L$  **就地逆置**，结果如图所示。



#### 4、链表删除操作

如图所示，要删除图示链表中的  $p$  结点，请用三条语句完成。



#### 5、循环链表合并操作

已知  $L_1$ 、 $L_2$  分别为两循环单链表的头结点指针， $m, n$  分别为  $L_1$ 、 $L_2$  表中数据结点个数。要求设计一算法，用**最快速度**将两表合并成一个带头结点的循环单链表。

LinkedList Union(LinkedList L1, L2; int m, n)

```
{  
  
}
```

## 6、双向链表

(1) 按照下列题目中算法功能说明，将算法描述片段中的**错误改正**过来。

①下面的算法描述片段用于在双链表中删除指针变量 **p** 所指的结点：

```
p->next=p->prior->next;
```

```
p->prior=p->next->prior
```

```
delete p;
```

②下面的算法描述片段用于在双链表中指针变量 **p** 所指结点后插入一个新结点：

```
new(q);
```

```
q->prior=p;
```

```
p->next=q;
```

```
q->next=p->next;
```

```
q=p->next->prior;
```

(2) 设双向循环链表中结点的数据域、前驱和后继指针域分别为 **data**, **pre** 和 **next**，试写出在指针 **p** 所指结点之前插入 **s** 结点的 C 语言描述语句。

## 7、查找单链表的中间结点

已知一个带有表头结点的单链表结点结构为(data, next)，假设该链表只给出了头指针 L。请设计一个尽可能高效的算法，查找链表的中间结点。若查找成功，算法输出该结点的 data 值，并返回 1；否则，只返回 0。要求：

- (1) 描述算法的基本设计思想；
- (2) 描述算法的详细实现步骤；
- (3) 根据设计思想和实现步骤，采用程序设计语言描述算法，关键之处给出简要注释。

### 第三章 栈和队列

#### 栈操作

1、设一数列的输入顺序为 123456，若采用栈结构，并以 X 和 Y 分别表示入栈和出栈操作，试问通过入栈、出栈操作的合法序列：

- (1) 能否得到输出顺序为 325641 的序列。
- (2) 能否得到输出顺序为 154623 的序列。

2、一个栈的入栈序列为 1,2,3,...,n，其出栈序列是  $p_1, p_2, \dots, p_n$ 。若  $p_2=3$ ，则  $p_3$  可能的取值个数是多少？

3、利用栈结构，编写算法将一个十进制数 n 转换为对应的十六进制数。

```
void TenToHex(int n,int B)//参数 n 指要转换的十进制数，参数 B=16 是要转换的进制
{
}
}
```

## 递归问题

有递归函数如下：

```
int s(int n)
{
    if(n==0) sum=0;

    else { cin>>x; sum=s(n-1)+x; }

    return sum;
}
```

设初值  $n=4$ ，读入  $x=4,9,6,2$ 。问：

(1) 若  $x$  为局部变量时，该函数递归结束后返回调用程序时  $sum$  的值为多少？

(2) 若  $x$  为全局变量时，该函数递归结束后返回调用程序时  $sum$  的值为多少？

## 括号匹配问题几种情况

在栈的应用中，一个典型的应用是“括号的匹配检测”问题。请根据书中算法，归纳几种情况，如什么时候完全匹配，什么时候不匹配，什么时候左括号多了，什么情况下右括号多了？

## 双栈操作

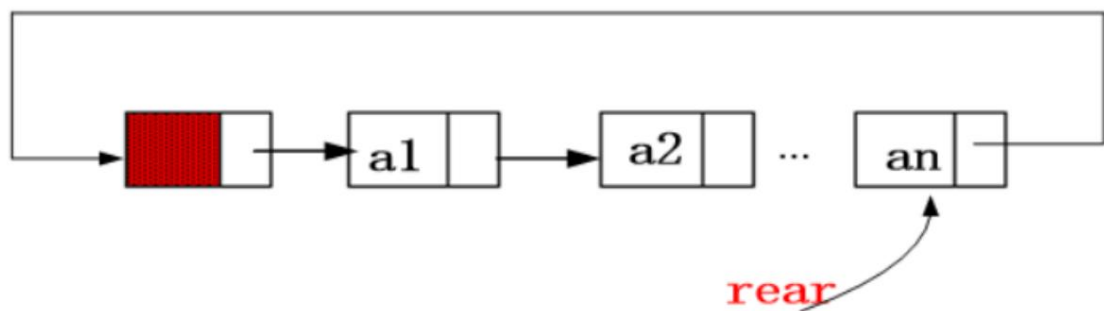
用一个数组  $S$ （设大小为  $M$ ）作为两个栈的共享空间。请说明：

- (1) 共享方法；
- (2) 给出栈满/栈空的判断条件

(3) 写出入栈操作  $\text{push}(i, x)$ ，其中  $i$  用于表示栈号，其值为 1 或 2， $x$  为入栈值。

## 循环单链表模拟队列操作

利用循环单链表模拟实现队列操作，请给出入队  $\text{EnQueue}$  和出队  $\text{DeQueue}$  过程，要求它们的时间复杂性都是  $O(1)$



```
void EnQueueq(LinkList &rear, ElemType e)
{
    //请在此描述进栈相应代码
}

void DeQueue(LinkList &rear, ElemType &e)
{
    //请在此描述出栈相应代码
}
```

## 栈和队列实现字符串特定操作

$S = "S_1S_2 \cdots S_N"$  是一个长为  $N$  的字符串，存放在一个数组中，请结合栈和队列数据结构编写算法将  $S$  改造之后输出：

- (1) 将  $S$  的所有第偶数个字符按照其原来的下标从大到小的次序放在  $S$  的后半部分；
- (2) 将  $S$  的所有第奇数个字符按照其原来的下标从小到大的次序放在  $S$  的前半部分；

例如：S=‘ABCDEFGH IJ K L’，则改造后的 S 为‘ACEGIKLJHFDB’。

## 第四章 串

### 1、子串的计算

设 S 为一个长度为 n 的字符串，其中的字符各不相同，试计算串 S 中的互异的非平凡子串（非空且不同于 S 本身）的个数。

### 2、字符串函数应用实例

已知三个字符串分别为 S=“ab...abcaabcbca...a”，S1=“caab”，S2=“bcb”。利用**所学字符串基本运算函数**，求得结果串 S3=“caabcbca...aca...a”，要求写出得到该结果串 S3 所用的函数及算法步骤。

### 3、字符串原子操作实现字符串其他函数功能

字符串的**原子操作**主要是串赋值 StrAssign，串比较 StrCompare，求子串 SubString，串连接 Concat 以及求串长 StrLength 等**5 个基本函数**组成。试**利用字符串的原子操作**实现字符串**删除**操作。

说明：字符串删除操作函数为 StrDelete(&S,pos,len)，其功能为从串 S 中删除第 pos 位置起长度为 len 的子串。

```
StrDelete(&S,pos,len)
```

```
{  
}
```

### 4、字符串模式匹配

设主串  $S = \text{"abaabaabcabaabc"}$ ，模式串  $T = \text{"abaabc"}$ ，采用 BF 算法和 KMP 算法进行模式匹配，到匹配成功时为止，在匹配过程中进行的单个字符间的比较次数分别是多少？

## 第五章 数组和广义表

### 矩阵的压缩存储

设对称矩阵  $A$ ，如下：

$$A = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 5 \\ 2 & 0 & 5 & 0 \end{bmatrix}$$

- (1) 画出对其压缩存储的存储表  $S$ ，试求出  $A$  中任一元素  $a_{ij}$  ( $1 \leq i, j \leq 4$ ) 与  $S$  中元素的下标  $K$  之间的关系；
- (2) 若将  $A$  视为稀疏矩阵，画出其三元组表形式压缩存储表。

### 稀疏矩阵快速转置

已知稀疏矩阵



$$M = \begin{bmatrix} 0 & 12 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 24 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & -7 & 0 & 0 & 0 \end{bmatrix}_{6 \times 7}$$

假设第一个元素存放的位置为 1, 根据快速矩阵转置算法, 计算表 5.1。

表 5.1 快速转置辅助向量表计算

col	1	2	3	4	5	6	7
Num[col]							
Cpot[col]	1						

## 寻址计算

1、二维数组 A 的元素都是 6 个 ASCII 字符组成的字符串, 行下标 i 的取值范围从 0 到 8, 列下标 j 的范围从 1 到 10。请解答下列问题。

- (1) 存放 A 至少需要的字节?
- (2) A 的第 6 行和第 8 列共占多少字节?
- (3) 若 A 以行序为主序存放, 计算元素 [8, 5] 的起始地址。
- (4) 若 A 以列序为主序存放, 计算元素 [5, 8] 的起始地址。

2、对角线矩阵如图 5.8 所示，其中  $n$  为奇数，除中心点外，其他行都有两个元素。

$$\begin{bmatrix}
 a_{11} & 0 & 0 & \cdots & 0 & 0 & a_{1n} \\
 0 & a_{22} & 0 & \cdots & 0 & a_{2,n-1} & 0 \\
 0 & 0 & \ddots & 0 & \vdots & 0 & 0 \\
 \vdots & \vdots & 0 & a_{\frac{n+1}{2}, \frac{n+1}{2}} & 0 & \vdots & \vdots \\
 0 & 0 & \vdots & 0 & \ddots & 0 & 0 \\
 0 & a_{n-1,2} & 0 & \cdots & 0 & a_{n-1,n-1} & 0 \\
 a_{n1} & 0 & 0 & \cdots & 0 & 0 & a_{nn}
 \end{bmatrix}$$

图 5.8 对角线矩阵

如果用一维数组  $B$  按行序存储对角线矩阵  $A$  的非零元素，问：

- (1) 请给出对角线矩阵  $A$  中非零元素的行下标与列下标的关系；
- (2) 假定矩阵中每个元素占一个存储单元且  $B$  的起始地址为  $A0$ ，给出利用  $a_{ij}$  在  $B$  中的位置公式物理地址公式。
- (3) 给出  $A$  中非零元素  $a_{ij}$  在  $B$  中的序号公式。

## 广义表运算

1、已知广义表  $L=((x,y,z), a, (u,t,w))$ ，写出利用取表头函数与取表尾函数从  $L$  表中取出原子项  $t$  的运算步骤。

2、请将香蕉 banana 用 Head( )和 Tail( )从广义表  $L$  中取出。

$$L=(apple,(orange,(strawberry,(banana)),peach),pear)$$

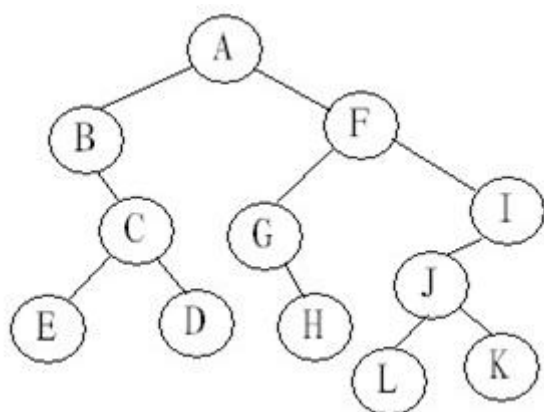
## 第六章 树和二叉树

1、给定二叉树的中序遍历序列为 BFDAEGC，层次遍历序列为 ABCDEFG，试画出该二叉树并写出它的后序遍历序列。

2、某森林F对应的二叉树为T，若T的先序遍历序列是 a, b, d, c, e, g, f，中序遍历序列是 b, d, a, e, g, c, f，试画出该森林F。

3、已知一算术表达式的中缀形式为  $A+B*C-D/E$ ，请写出该表达式的前缀形式和后缀形式。

1、 已知由森林转换得到的二叉树如下图所示，试画出原来的森林并写出它的中序遍历序列。



5、算法设计题，以下二叉树均采用二叉链表存储结构，定义如下：

```
typedef struct BiTNode{  
    char data;  
    struct BiTNode *lchild, *rchild;  
}BiTNode, *BiTree;
```

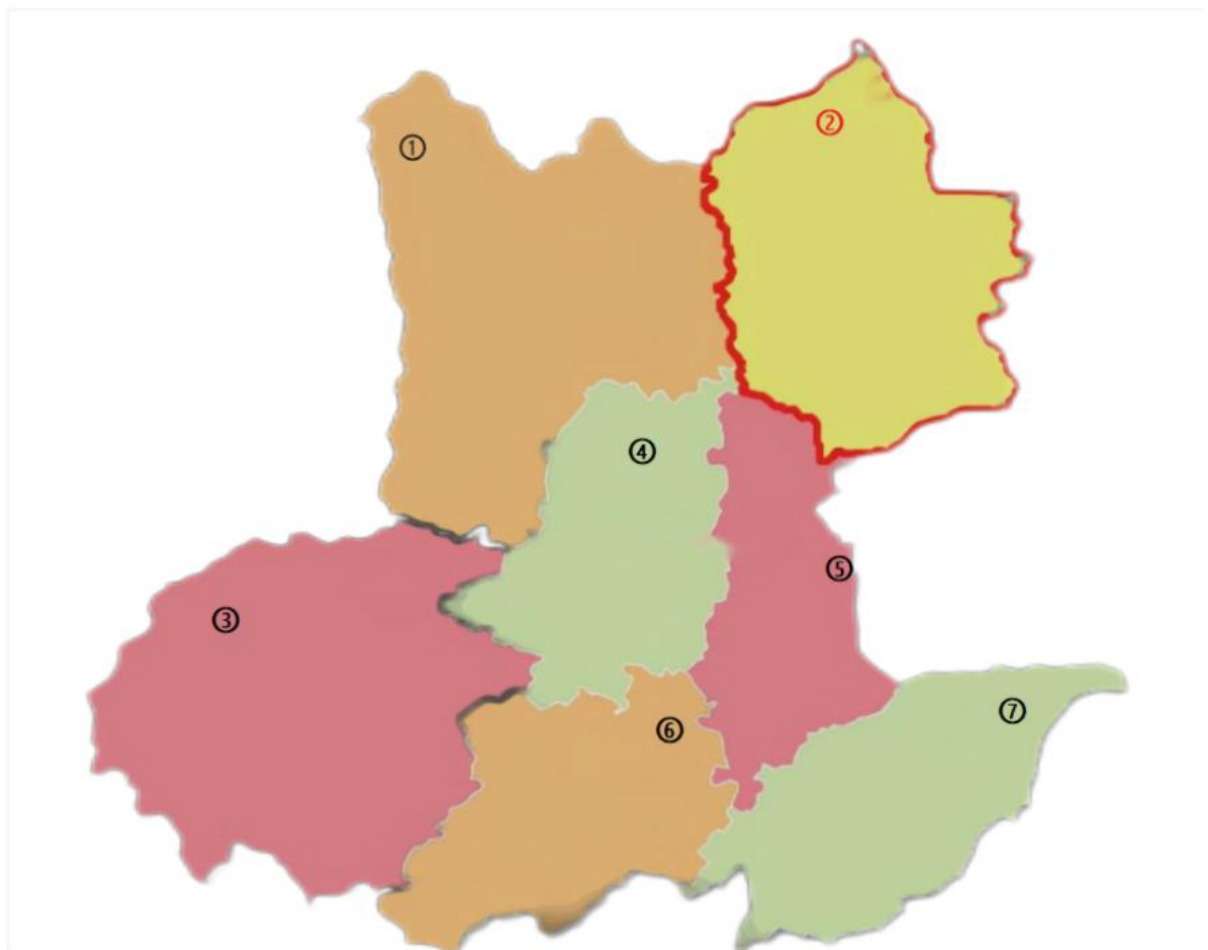
① 编写递归函数计算二叉树 T 中所有叶子结点的个数。

② 编写递归函数计算二叉树 T 中所有结点的个数。

- ③ 写一个递归函数，计算二叉树中度数为 1 的结点个数。
- ④ 写一个递归函数，计算二叉树 T 的深度。
- ⑤ 写一个递归函数，交换二叉树 T 中所有结点的左子树和右子树。
- ⑥ 设计算法输出二叉树后序遍历序列的逆序结果。

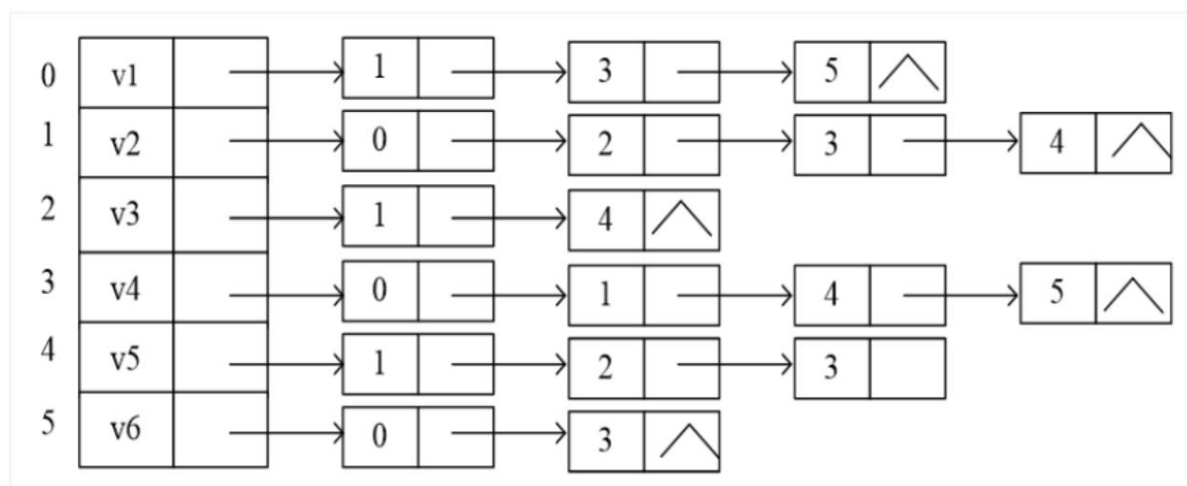
## 工程问题——哈夫曼应用

下图是某行政区域图，依次编号为①到⑦，其中有个区域出现一例新冠疫情阳性患者，根据疫情传播特点，假设①到⑦区域是疫情高风险的概率分别是 20%，30%，20%，10%，10%，5%，5%。**为快速准确发现阳性患者，确定高风险区域**，在节省成本的情况下，混检是一种常用的新冠疫情检测手段。请问**采用什么混检方案确定高风险区域，使得平均混检次数最小，也即成本最小**？答案是不是唯一的？请解释为什么。



## 一、图遍历算法

1、邻接表存储示意图如下，若从顶点 **v1** 出发对该图进行遍历，分别给出一个按深度优先遍历和广度优先遍历的顶点序列，并画出对应的生成树或者森林。



2、已知一个无向图的顶点集  $V$  和边集  $E$  分别为： $V = \{v_0, v_1, v_2, v_3, v_4\}$ ；

$E = \{(v_0, v_3), (v_0, v_2), (v_0, v_1), (v_1, v_4), (v_2, v_4), (v_3, v_4)\}$ ，试给出图的邻接矩阵和邻接表存储示意图，写出以  $v_0$  为起点，根据该图的邻接表进行深度优先搜索和广度优先搜索遍历得到的顶点序列。

3、已知一个有向图的顶点集  $V$  和弧集  $E$  分别为： $V = \{A, B, C, D, E, F, G, H\}$ ；

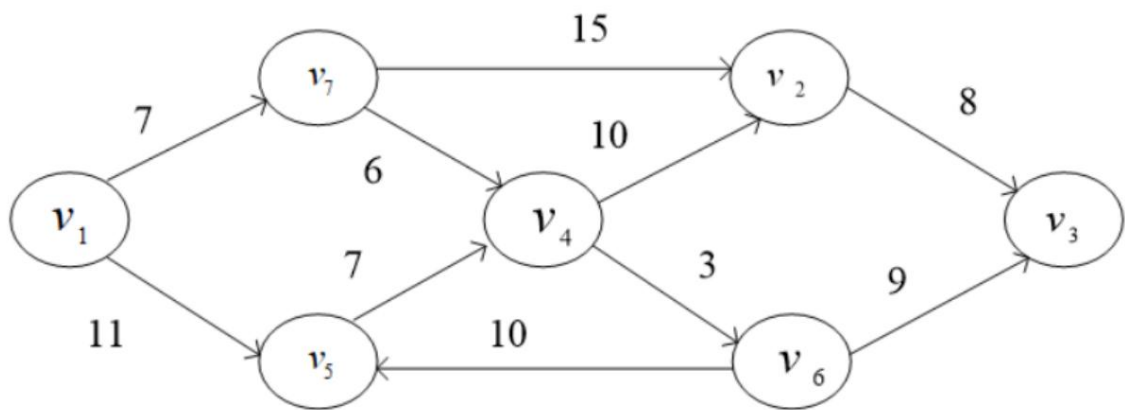
$E = \{\langle A, C \rangle, \langle A, D \rangle, \langle C, F \rangle, \langle D, G \rangle, \langle F, G \rangle, \langle B, D \rangle, \langle B, E \rangle, \langle B, H \rangle, \langle E, H \rangle, \langle H, G \rangle\}$ ，请完成：

(1) 画出该图；

(2) 画出以  $A$  为起点遍历该图所得到的 DFS（深度优先搜索）生成森林和 BFS（广度优先搜索）生成森林。

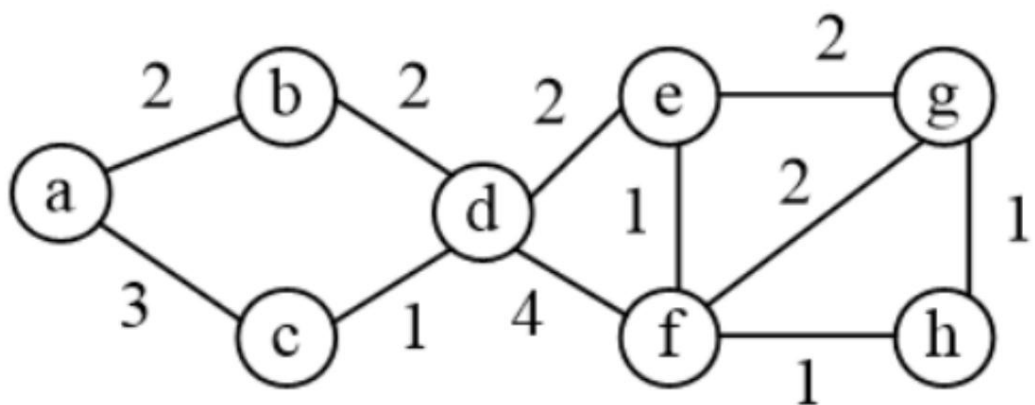
## 二、最短路径问题

对于图所示的带权有向图，求从源点 **v1** 到其他各顶点的最短路径。



### 三、最小生成树问题

对下图所示的连通图，请用 **Prim** 算法构造其**最小生成树**。



### 哈希函数法

对下面的关键字集 {30, 15, 21, 40, 25, 26, 36, 37} 若查找表的装填因子为

0.8，采用线性探测再散列方法解决冲突，做：

(1) 设计哈希函数；

(2) 画出哈希表；

(3) 计算查找成功和查找失败的平均查找长度；

### 二叉排序树问题

一个正整数序列（53,17,12,66,58,70,87,25,75,60），试完成下列各题。

- （1）按次序构造一棵二叉排序树。
- （2）若序列中每个元素的查找概率相等，求平均查找长度。

排序问题

给出一组关键字 {39，28，52，47，58，22，51，20}，写出堆排序、快速排序变化过程。

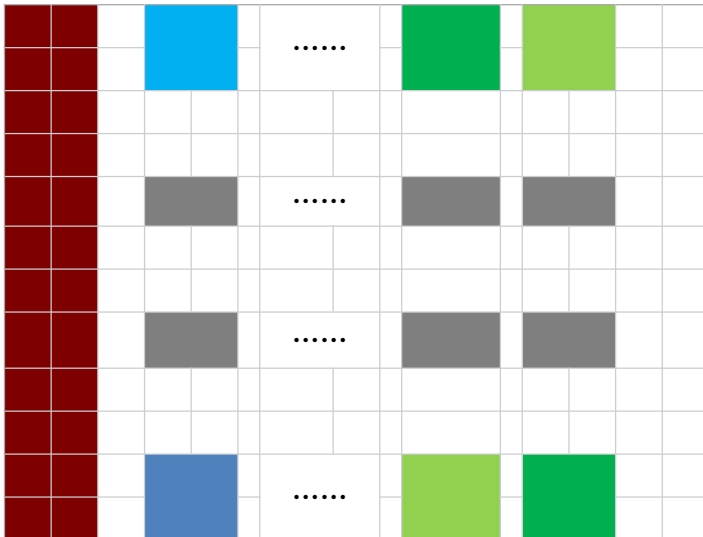
工程设计问题——排序算法

随着大数据和深度学习技术的快速发展，聊天机器人技术也逐渐成熟；ChatGPT 推出，更进一步推动了人工智能研究的火热。而分类算法属于人工智能领域经典的算法，主要用于将数据集中的样本分为不同的类别。如下图，根据条件属性{A,B,C}，通过分类算法，不难发现：由于在 3 个条件属性下{x1，x4}取值都是 121，所以划分为一类；同理，{x2，x6}取值都是 212，所以它们也属于一类，以此可以完成 3 个条件属性下数据的分类操作。

试结合《数据结构》学习过的**基数排序算法**，完成下面数据集划分过程。

样本	A	B	C
X1	1	2	1
X2	2	1	2
X3	1	2	2
X4	1	2	1
X5	2	1	1
X6	2	1	2

工程设计问题——双哈希函数



随着大数据和深度学习技术的快速发展，聊天机器人技术也逐渐成熟；ChatGPT 推出，更进一步推动了人工智能研究的火热。而分类算法属于人工智能领域经典的算法，主要用于将数据集中的样本分为不同的类别。如下图，根据条件属性{A,B,C}，通过分类算法，不难发现：由于在 3 个条件属性下{x1, x4}取值都是 121，所以划分为一类；同理，{x2, x6}取值都是 212，所以它们也属于一类，以此可以完成 3 个条件属性下数据的分类操作。

试结合《数据结构》学习过的**基数排序算法**，完成下面数据集划分过程。

样本	A	B	C
X1	1	2	1
X2	2	1	2
X3	1	2	2
X4	1	2	1
X5	2	1	1
X6	2	1	2



表 1 决策表

	$a$	$b$	$c$	$d$	$D$
$X1$	1	1	1	1	0
$X2$	2	2	2	1	1
$X3$	1	1	1	1	0
$X4$	2	3	2	3	0
$X5$	2	2	2	1	1
$X6$	3	1	2	1	0
$X7$	1	2	3	2	2
$X8$	2	3	1	2	3
$X9$	3	1	2	1	1
$X10$	1	2	3	2	2
$X11$	3	1	2	1	1
$X12$	2	3	1	2	3
$X13$	4	3	4	2	1
$X14$	1	2	3	2	3
$X15$	4	3	4	2	2

$front[0] \rightarrow X1 \rightarrow X3 \rightarrow X7 \rightarrow X10 \rightarrow X14 \leftarrow end[0],$   
 $front[1] \rightarrow X2 \rightarrow X4 \rightarrow X5 \rightarrow X8 \rightarrow X12 \leftarrow end[1],$   
 $front[2] \rightarrow X6 \rightarrow X9 \rightarrow X11 \leftarrow end[2],$   
 $front[3] \rightarrow X13 \rightarrow X15 \leftarrow end[3].$

第1趟“收集”结果为

$\rightarrow X1 \rightarrow X3 \rightarrow X7 \rightarrow X10 \rightarrow X14 \rightarrow X2 \rightarrow X4 \rightarrow X5 \rightarrow$   
 $X8 \rightarrow X12 \rightarrow X6 \rightarrow X9 \rightarrow X11 \rightarrow X13 \rightarrow X15.$

第2趟“分配”结果为

$front[0] \rightarrow X1 \rightarrow X3 \rightarrow X6 \rightarrow X9 \rightarrow X11 \leftarrow end[0],$   
 $front[1] \rightarrow X7 \rightarrow X10 \rightarrow X14 \rightarrow X2 \rightarrow X5 \leftarrow end[1],$   
 $front[2] \rightarrow X4 \rightarrow X8 \rightarrow X12 \rightarrow X13 \rightarrow X15 \leftarrow end[2].$

第2趟“收集”结果为

$\rightarrow X1 \rightarrow X3 \rightarrow X6 \rightarrow X9 \rightarrow X11 \rightarrow X7 \rightarrow X10 \rightarrow X14 \rightarrow$   
 $X2 \rightarrow X5 \rightarrow X4 \rightarrow X8 \rightarrow X12 \rightarrow X13 \rightarrow X15.$

由第4步得到  $U / \{a, b, c, d\}$  为

$\{X1, X3\}, \{X6, X9, X11\}, \{X2, X5\}, \{X8, X12\},$   
 $\{X7, X10, X14\}, \{X13, X15\}, \{X4\}.$

U	A	Outlook	Temperature	Humidity
1		sunny	hot	high
2		sunny	hot	high
3		overcast	hot	high
4		rain	mild	high
5		rain	cool	normal
6		rain	cool	normal
7		overcast	cool	normal
8		sunny	mild	high

9	sunny	cool	normal
10	rain	mild	normal