

考试课程	数据结构与算法		考试日期	2016 年 1 月    日			成绩		
课 程 号	A101166D	教 师 号		任课教师姓名			丁宏, 僧德文, 彭伟民, 陈滨		
考生姓名		学号(8 位)		年级		专业		座位号	

一. 是非题（每题 2 分，共 10 分）（正确的打“√”，错误的打“×”。）

1. 在顺序存储结构中，有时也存储数据结构中元素之间的关系。（ × ）
2. 对任何数据结构链式存储结构一定优于顺序存储结构。（ × ）
3. 队列和栈都是运算受限的线性表，只允许在表的两端进行运算。（ × ）
4. 广义表的同级元素（直属于同一个表中的各元素）具有线性关系。（ √ ）
5. 完全二叉树中，若一个结点没有左孩子，则它必是树叶。（ √ ）

二. 选择题（每选 2 分，共 20 分）

1. 算法的时间复杂度取决于（ D ）。
- A. 问题的规模      B. 待处理数据的初态      C. 处理器的速度      D. A 和 B
2. 设一个链表最常用的操作是在末尾插入结点和删除尾结点，则选用（ D ）最节省时间。
- A. 单链表    B. 单循环链表    C. 带尾指针的单循环链表    D. 带头结点的双循环链表
3. 有六个元素 6，5，4，3，2，1 的顺序进栈，下列哪一个不是合法的出栈序列？（ C ）
- A. 5 4 3 6 1 2      B. 4 5 3 1 2 6      C. 3 4 6 5 2 1      D. 2 3 4 1 5 6
4. 若串 S=' software'，其子串的数目是（ B ）。
- A. 8      B. 37      C. 36      D. 9
5. 假设以行序为主序存储二维数组 A=array[1..100, 1..100]，设每个数据元素占 2 个存储单元，基地址为 10，则 LOC[5, 5]=（ B ）。
- A.      808      B.      818      C.      1010      D. 1020
6. 具有 10 个叶结点的二叉树中有（ B ）个度为 2 的结点。
- A. 8      B. 9      C. 10      D. 11
7. 在一个无向图中，所有顶点的度数之和等于所有边数的（ B ）倍，在一个有向图中，所有顶点的入度之和等于所有顶点出度之和的（ C ）倍。
- A. 1/2      B. 2      C. 1      D. 4
8. 对线性表进行二分查找时，要求线性表必须（ B ）。
- A. 以顺序方式存储    B. 以顺序方式存储，且数据元素有序
- C. 以链接方式存储    D. 以链接方式存储，且数据元素有序
9. 下列排序算法中，其中（ D ）是稳定的。

- A. 堆排序，冒泡排序      B. 快速排序，堆排序      C. 直接选择排序，归并排序      D. 归并排序，冒泡排序

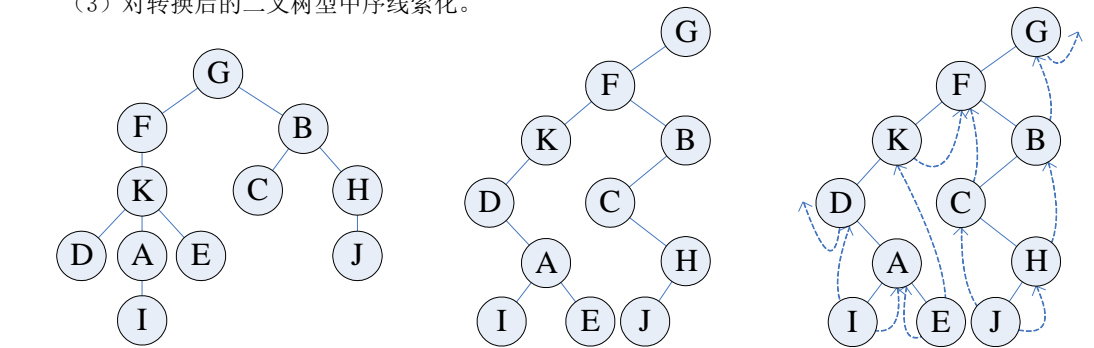
三. 填空题（每空 2 分，共 20 分）

1. 语句 s 的执行次数为 (n+3)(n-2)/2。
- FOR(i=l; i<n-l; i++)
- FOR(j=n; j>=i; j--)
- s;
2. 设单链表的结点结构为(data, next)，next 为指针域，已知指针 px 指向单链表中 data 为 x 的结点，指针 py 指向 data 为 y 的新结点，若将结点 y 插入结点 x 之后，则需要执行以下语句: py->next=px->next; px->next=py;。
3. 设循环队列存放在向量 sq.data[0:M] 中，则队头指针 sq.front 在循环意义下的出队操作可表示为 sq.front=(sq.front+1)%(M+1); return (sq.data[sq.front]);，若用牺牲一个单元的办法来区分队满和队空（设队尾指针 sq.rear），则队满的条件为 (sq.rear+1)%(M+1)==sq.front;。
4. 广义表运算式 HEAD(TAIL(((a,b,c),(x,y,z))))的结果是 (x,y,z)。
5. 设只含根结点的二叉树的高度为 0，则高度为 k 的二叉树的最大结点数为 2<sup>k+1</sup>-1，最小结点数为 k+1。
6. 在图 G 的邻接表表示中，每个顶点邻接表中所含的结点数，对于无向图来说等于该顶点的 度；对于有向图来说等于该顶点的 出度。

四. 结构问答题（每题 6 分，共 30 分）

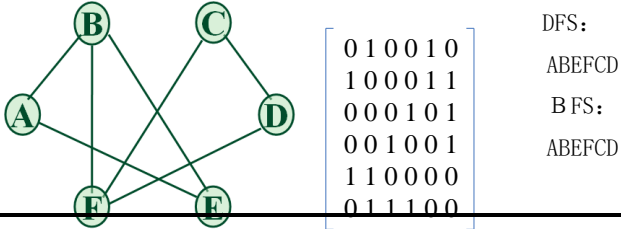
1. 已知某树的先根遍历次序为：GFKDAIEBCHJ，后根遍历次序为：DIAEKFCJHBG。

- （1）画出该树形。
- （2）用孩子-兄弟法将其转换成二叉树。
- （3）对转换后的二叉树型中序线索化。



2. 已知某无向图如右图所示：

- （1）画出该图的邻接表存储结构。
- （2）画出该图的邻接矩阵存储结构。
- （3）根据所绘制的邻接表给出 DFS 及 BFS 次序。

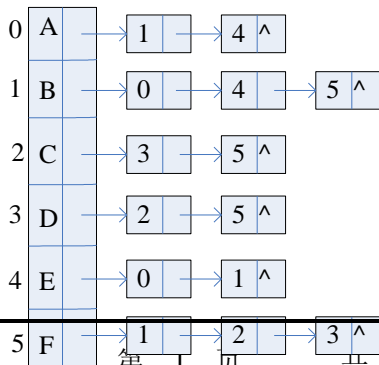


DFS:

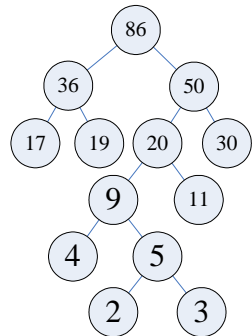
ABEFC D

BFS:

ABEFC D



3. 已知关键字序列 R={11, 4, 3, 2, 17, 30, 19}, 构造一棵哈夫曼树, 并计算出它的带权路径长度 WPL。



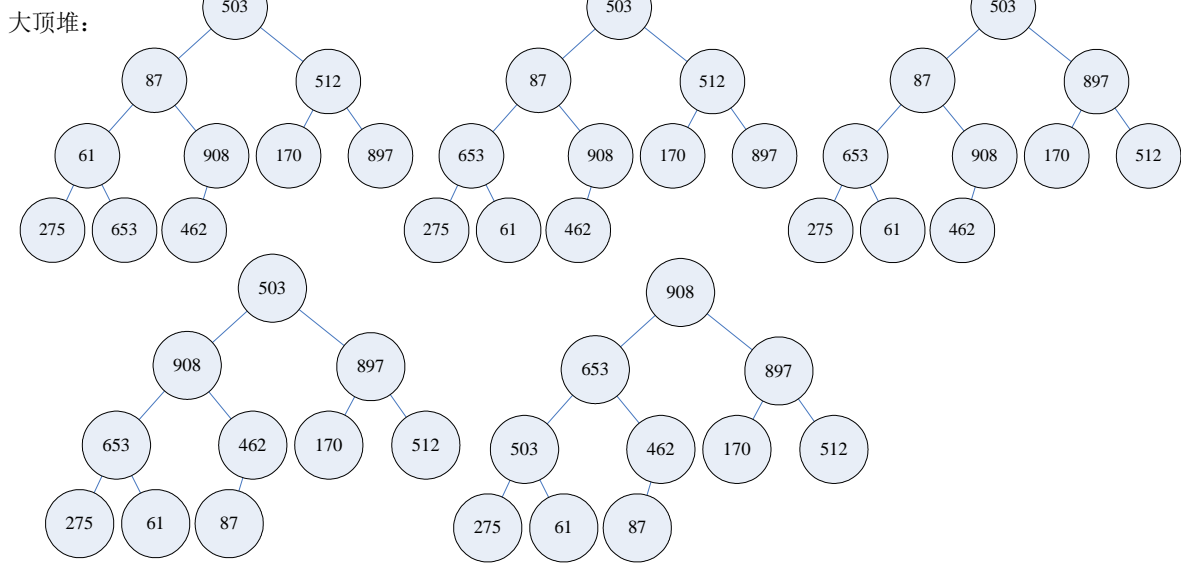
W P L :  
 $2 \times 5 + 3 \times 5 + 4 \times 4 + 11 \times 3 + 17 \times 2 + 19 \times 2 + 30 \times 2 = 206$

4. 设有一组关键字 {9, 01, 23, 14, 55, 20, 84, 27}, 采用哈希函数:  $H(\text{key}) = \text{key} \bmod 7$ , 表长为 10, 用开放地址法的二次探测再散列方法  $H_i = (H(\text{key}) + d_i) \bmod 10 (d_i = 1^2, 2^2, 3^2, \dots)$  解决冲突。要求: 对该关键字序列构造哈希表, 并计算查找成功的平均查找长度 ASL。

散列地址	0	1	2	3	4	5	6	7	8	9
关键字	14	01	9	23	84	27	55	20		
比较次数	1	1	1	2	3	4	1	2		

平均查找长度:  $ASL_{\text{succ}} = (1+1+1+2+3+4+1+2) / 8 = 15/8$

5. 已知待排序的序列为 (503, 87, 512, 61, 908, 170, 897, 275, 653, 462), 根据此序列建立一个堆, 画出每一步的结果图。



五. 算法分析题 (指出算法的功能, 并分析算法的时间复杂度。)(每小题 6 分, 共 12 分)

1. 函数 A1 如下: (功能: 如果 L 的长度不小于 2, 将 L 的首元结点变成尾元结点; 时间复杂度:  $O(n)$ )

void A1(LinkedList L)//L 是无表头结点的单链表, 表长为 n

```
{
    if (L && L->next) {
        Q=L; L=L->next; P=L;
        while (P->next) P=P->next;
        P->next=Q; Q->next=NULL;
    }
    return OK;
} //A1
```

2. 函数 A2 如下: (功能: 队列逆置; 时间复杂度:  $O(n)$ )

void A2(Queue &Q)//队列 Q 的长度为 n

```
{
    Stack S; int d;// 栈 S 的长度为 m
    InitStack(S);
    while (!QueueEmpty(Q)) {
        DeQueue(Q, d);
        Push(S, d);
    }
    while (!StackEmpty(S)) {
        Pop(S, d);
        EnQueue(Q, d);
    }
}
```

六. 算法设计题 (8 分)

1. 冒泡排序算法是把大的元素向上移 (气泡的上浮), 也可以把小的元素向下移 (气泡的下沉)。请给出上浮和下沉过程交替的冒泡排序算法。

**void BubbleSort2(int a[],int n)** //相邻两趟向相反方向起泡的冒泡排序算法  
{ change=1;low=0;high=n-1; //冒泡的上下界  
  **while**(low<high && change)  
  { change=0;           //设不发生交换  
    **for**(i=low;i<high;i++) //从上向下起泡  
      **if**(a[i]>a[i+1]){a[i]<-->a[i+1];change=1;} //有交换, 修改标志 change  
    high--; //修改上界  
    **for**(i=high;i>low;i--) //从下向上起泡  
      **if**(a[i]<a[i-1]){a[i]<-->a[i-1];change=1;}  
    low++; //修改下界  
  }**while**  
}**//BubbleSort2**