

# NO.1

## 一、判断题 (每小题 1 分, 共 10 分)

1. 数据的逻辑结构与各数据元素在计算机中如何存储有关。
2. 凡是为空的单链表都是不含任何节点的。
3. 对顺序栈进行进栈、出栈操作, 不涉及元素的前、后移动问题。
4. 含有  $n$  个字符的字符串中所有子串的个数为  $n*(n+1)/2 + 1$ 。
5. 递归算法的执行效率比功能相同的非递归算法的执行效率高。
6. 在  $n$  ( $n > 3$ ) 阶三对角矩阵中, 每一行都有三个非零的元素。
7. 霍夫曼树中不存在度为 1 的节点。
8. 强连通图不能进行拓扑排序。
9. 二叉排序树是用来进行排序的。
10. 排序的稳定性是指排序算法中比较的次数保持不变, 且算法能够终止。

## 二、填空题 (每小题 1 分, 共 10 分)

1. 一个算法具备的 5 个特性分别是可行性、有穷性、\_\_\_\_、输入和输出。
2. 在有  $n$  个元素的顺序表中的任意位置插入一个元素所需移动元素的平均次数为\_\_\_\_\_。
3. 栈是一种具有\_\_\_\_\_特性的线性表。
4. 无论是顺序队列还是链式队列, 插入和删除运算的时间复杂度都是\_\_\_\_\_。
5. 递归函数  $f(1)=1$ ,  $f(n) = f(n-1) + n(n>1)$  的递归出口是\_\_\_\_\_。
6. 完全二叉树中节点个数为  $n$ , 则编号最大的分支节点的编号是\_\_\_\_\_。



---

节点，则该树有\_\_\_\_\_个叶子节点。

A、 $1 + 2b + 3c$

B、 $1 + 2b + 3c$

C、 $2b + 3c$

D、 $1 + b + 2c$

7. 若二叉树的中序遍历序列是 abcdef，且 c 为根节点，则\_\_\_\_\_。

A、c 左子树中有两个节点

B、二叉树有两个度为 0 的节点

C、二叉树的高度为 5

D、以上都不对

8. 无向图的邻接矩阵是一个\_\_\_\_\_。

A、对称矩阵

B、零矩阵

C、上三角矩阵

D、对角矩阵

9. 在如图 1 所示的平衡二叉树中插入关键字 50 后，得到一棵新平衡二叉树，在新平衡二叉树中，关键字 37 所在的节点的左右孩子节点中保存的关键字分别是\_\_\_\_\_。

A、13，50

B、24，50

C、24，53

D、24，90

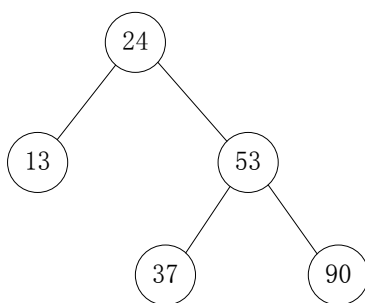


图 1 一棵平衡二叉树

10. 如果从无向图的任一顶点出发进行一次深度优先搜索即可访问所有顶点，则该图一定是\_\_\_\_\_。

A、完全图

B、连通图

C、有回路

D、一棵树

11. 采用分块查找时，若线性表中共有 625 个元素，查找每个元素的概率



- 
1. 以数据集合 {2, 5, 7, 9, 13} 为权值构造一棵霍夫曼树，并计算其带权路径长度。

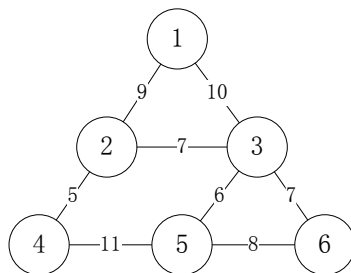


图 2 带权无向图

2. 给定如图 2 所示的带权无向图 G，给出采用克鲁斯卡尔算法构造的最小生成树的过程。
3. 从空二叉排序树开始，依次读入关键字序列 (7, 16, 4, 8, 20, 9, 18, 5) 构造一棵二叉排序树，画出该二叉排序树，并计算在等概率情况下，该二叉排序树查找成功时的平均查找长度 ASL。
4. 以关键字序列 {265, 301, 751, 129, 937, 863, 742, 694, 076, 438} 为例，写出使用快速排序算法的各趟排序后，关键字序列的状态。
5. 对含有 n 个互不相同元素的线性表，试设计程序同时找最大值和最小值元素。

## 五、算法设计题（每小题 10 分，共 20 分）

1. 有 3 个带头结点并且节点值递增的单链表 h1、h2 和 h3，他们的节点个数分别为 m、n 和 k，单链表的节点类型如下：

```
typedef struct node
{
    int data ;
    struct node * next ;
} LinkList ;
```

---

试设计一个算法：

void merge(LinkList \* &h, LinkList \* h1, LinkList \* h2, LinkList \* h3), 将 h1、h2 和 h3 的所有节点归并成一个新的递增单链表 h, 要求空间复杂度为  $O(1)$ , 时间复杂度为  $O(m + n + k)$ 。

2. 试写一递归算法, 从大到小输出二叉排序树中所有其值不小于 x 的关键字。