1.数据结构中评价算法的两个重要指标是 \_\_\_\_\_\_\_\_\_\_\_\_。

2.数据的运算最常用的有5种，它们分别是 。

3.对数据存储无外乎是\_\_\_\_\_\_\_\_\_\_\_\_等四种。

4.以下程序的功能是实现带附加头结点的单链表数据结点逆序连接，请填空完善

void reverse(\*head)

/\* head为附加头结点指针/

linklist \*p,\*q;

p=h->next; h->next=NULL;

While\_\_\_\_\_\_\_\_\_\_\_

{q=p; p=p->next; q->next=h->next; h->next=(2)\_\_\_\_\_\_\_\_ }

}

5.下列程序判断字符串s 是否对称，对称则返回1，否则返回0；如 f("abba")返回1，f("abab")返回0,完善之

int f \_\_\_\_\_\_\_\_

{ int i=0,j=0;

while (s[j])\_\_\_\_\_\_\_\_;

for(j--; i<j && s[i]==s[j]; i++,j--);

Return \_\_\_\_\_\_\_

}

6.对广义表执行以下操作的结果：GetHead【GetTail【GetHead【((a,b),(c,d))】】】\_\_\_\_\_\_\_\_\_。

7.已知单链表L，(\*p)结点即不是开始结点，也不是尾结点，(\*q)结点为（\*p）结点的直接前趋结点，则删除(\*p)结点的语句组为（共两句）： 。

8.分析以下程序段的时间复杂度为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

k=1;

While (k<=n)

k = k\*3;

9.假设循环队列设front和rear指示队头、队尾元素位置，maxsize为循环队列最大容量，则循环队列长度为： 。

10.计算一棵二叉树的高度，系统采用栈来完成。请将下列算法填空完整。

int Height(Btree \*t)

{ int h1,h2;

if(t==NULL)

return 0;

else if ( )

return 1;

else

{ h1= ;

h2= ;

Return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

}

11.已知q是指向中序线索二叉树上某个结点的指针，本函数返回中序下\*q的后继的指针，请完善之

BiTree InSucc(BiTree q){

r=q->rchild;

while(!r->ltag)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Return\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}//ISucc

12.设t是给定的一棵二叉树，下面的递归程序count(t)用于求得:二叉树t中具有非空的左,右两个儿子的结点个数N2;只有非空左儿子的个数NL;只有非空右儿子的结点个数NR和叶子结点个数N0。N2、NL、NR、N0都是全局量，且在调用count(t)之前都置为0.

typedef struct node

{int data; struct node \*lchild,\*rchild;}node;

int N2,NL,NR,N0;

void count(node \*t)

{if (t->lchild!=NULL) if \_\_\_\_\_\_\_\_\_ N2++; else NL++;

else if \_\_\_\_\_\_\_\_\_\_\_ NR++; else \_\_\_\_\_\_\_\_\_\_\_\_\_\_;

if(t->lchild!=NULL)\_\_\_\_\_\_\_\_\_\_\_\_\_\_; if (t->rchild!=NULL) \_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

13.设要将序列（Q, E, C, Y, P, A, M, B, R, D, Z, X）中的关键码按字母序的升序重新排列，

则：冒泡排序一趟扫描的结果是 ；

初始步长为4的希尔（shell）排序一趟的结果是 ；

二路归并排序一趟扫描的结果是 ；

快速排序一趟扫描的结果是(采用初始数据为支点，两头交替逼近的实现方法) ；

给定关键字序列11，36，10，1，3，9，19，13，21试分别用顺序查找、二分查找（假设已排序）、二叉排序树查找、散列查找(开地址，用线性探查法（模取11的HASH函数）和拉链法)来实现查找的平均查找长度ASL值分别是

顺序查找\_\_\_\_\_\_\_\_\_\_\_;

二分查找\_\_\_\_\_\_\_\_\_\_;

二叉排序树查找\_\_\_\_\_\_\_\_\_\_\_\_\_;

线性探查法\_\_\_\_\_\_\_\_\_\_\_;

拉链法\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.