一、**单项选择**

1. 数据结构就是一门研究非数值计算得程序设计问题中,数据元素得①__C__、数据信息在计算机中得②__A__以及一组相关得运算等得课程.

① A．操作对象　B．计算方法　C。逻辑结构　D.数据映象

② A.存储结构　B．关系　C。运算　D．算法

2. 以下数据结构中，__D__就是线性结构.

A.广义表　B.二叉树　C。稀疏矩阵　D。串

3. 从逻辑上可以把数据结构分为__C__两大类。

A.动态结构与静态结构　B.顺序结构与链式结构

C.线性结构与非线性结构　D.初等结构与构造型结构

4. 以下数据结构中，__D__就是线性结构。

A.广义表　B.二叉树　C。稀疏矩阵　D.串

5. 以下数据结构中，__D__就是非线性结构。

A。栈　B.二叉树　C。队列　D。字符串

6. 数据结构 DS（Data Struct)可以被形式地定义为 DS＝（D，R),其中 D 就是①__B__得有限集合,R 就是 D 上得②__D__有限集合。

① A.算法　B．数据元素　C。数据操作　D.数据对象

② A．操作　B.映象　C.存储　D。关系

7. 线性表得顺序存储结构就是一种①__A__得存储结构，线性表得链式存储结构就是一种得②__B__存储结构。

A。随机存取　B。顺序存取　C．索引存取　D.散列存取

8. 线性表得逻辑顺序与存储顺序总就是一致得，这种说法__B__。

A、 正确　B、 不正确

9. 下面那一条就是顺序存储结构得优点？ （A)

A 、 存储密度大　B、 插入运算方便　C、 删除运算方便

D、 可以方便得用于各种逻辑结构得存储表示

10. 线性表采用链式存储结构时，要求内存中可用得存储单元得地址 _____、

A 、 必须就是连续得　　B、 部分地址必须就是连续得　C、 一定不连续
D、 连续与不连续都可以

11. 表长为 n 得顺序存储得线性表， 当在任何位置上插入与删除一个元素得概率相等时, 插入一个元素所需要移动元素得平均次数为 E ， 删除一个元素所需要移动元素得平均次数为 A

A、 （n－1）/ 2　　B、n　　C、 n+ 1　　D、 n－1　　E、 n/2　　F、 (n+1)/2　　G、 (n－2）/ 2

12. 带头结点得单链表 head 为空得判定条件就是 B.

A、 head = =NULL　　B、 head—>next = =NULL
C、 head－>next= =head　　D、 head！=NULL

13. 在一个单链表中， 若删除 p 所指向结点得后继结点, 则执行 A。

A、 p—>next= p—>next-〉 next　　B、 p=p->next; p—〉 next = p—〉 next－〉 next
C、 p = p—>next —>next　　D、 p= p —〉 next

14. 若已知一个栈得入栈序列就是 1，2，3，…，n，其输出序列为 p1，p2，p3，…,pn,若 p1=n，则 pi 为 C。

A、 i　　B、 n=i　　C、 n—i+ 1　　D、 不确定

15. 设栈得输入次序为： 1 ， 2， 3， 4，5，则 不可能就是其出栈序列、

A、 5 4321　B、 4 5 3 2 1　C、 4 3512　　D、 12 3 45

16. 一个递归算法必须包括 B

A、 递归部分　　B、 终止条件与递归部分
C、 迭代部分　　D、 终止条件与迭代部分

17. 用链接方式存储得队列， 在进行删除操作时 D

A 仅修改头指针　　B、 仅修改尾指针
C、 头尾指针都要修改　　D、 头尾指针可能都要修改

18. 数组 A［m]存放循环队列得元素， 其头尾指针分别就是 front 与 rear， 则当前队列得元素个数就是 A。

A、 （rear-front +m)%m　B、 （front—rear+m）%m　C、 front－rear＋1　D、 rear-front +1

19. 栈与队列得共同特点 C。

A、 都就是先进先出　　B、 都就是先进后出
C、 允许在端点插入与删除元素　D、 没有共同点

20. 一个栈得入栈序列 a，b，c，d， e ，则栈得输出序列就是__A___.

A、 edcba　　B、 de c b a　　C、 d c eab　　D、 ab c de

21. 栈得特点就是__B___，队列得特点就是__A__。

A、 先进先出　　B、 先进后出

22. 从一个栈顶指针 HS 得链表中删除一个结点， 用 x 保存被删除得结点值，执行得语句为__C__。

A、 x ＝HS; H S=HS－>ne x t　　B 、 HS＝H S—>next; x =HS—>d a ta

C、 x=HS-> d a t a; H S=HS－>nex t　　D、 HS—> n ext＝H S; x=H S-＞data

23. 在链队列 Q 中， 插入 s 所指向得结点执行得语句为__C__.

A、 Q、fro n t —〉next=s;　　B、 Q、rea r ->ne x t=s; Q 、re a r =s

C、 s—〉next=Q、re a r; Q 、 r ear=s　　D、 s —〉next＝Q 、front; Q、front= s

24. 空串与空格串就是相同得，这种说法__B__。

A、 正确　　B、 不正确

25. 下面关于串得叙述， 哪一个就是不正确得__B__。

A、 串就是字符得有限序列　　B、 空串就是由空格构成得串

C、 匹配模式就是串得一种重要运算　　D 、 串可以采用链式存储结构

26. 设有两个串 p 与 q,求 q 在 p 中首次出现得位置得运算称作___B__。

A、 连接　　B、 模式匹配

C、 求子串　　D、 求串长

27. 若串 s=’so f tware', 其子串得数目为____B____

A 、 8　　B、 37　　C、 3 6　　D、 9

28. 二维数组 A 中,每个元素 A 得长度为 3 个字节，行下标 i 从 0 到 7 ，列下标 j 从 0 到 9,从首地址 S A 开始连续存放在存储器内，该数组按行存放时,数组元素 A［7］[4]得起始地址为__C__。

A 、 SA+1 4 1　　B、 S A +144　　C、 S A+22 2　　D、 SA+ 2 25

29. 对稀疏矩阵进行压缩存储得目得就是____C__、

A、 便于进行矩阵运算　　B、 便于输入输出

C 节省存储空间　　D、 降低运算得时间复杂度

30. 在以下叙述中正确得就是__B____
A. 线性表得线性存储结构优于链表存储结构
B. 二维数组可以瞧成就是其数据元素为线性表得线性表
C. 栈得操作方式就是先进先出
D. 队列得操作方式就是先进后出
31. 广义表((a)，a)得表头为___C_____，　表尾为____C____、
A、 ()　　B、 a　　C、 (a)　　D、 ((a))
32. 已知广义表L=((x,y,z),a,(u，t,w))，　从L中取出原子项t得运算为___D__.
A、 Head(Tail(Tail(L)))　　B、 Tail(Head(Head(Tail(Tail(L)))))
C、 Head(Tail(Head(Tail(Tail(L)))))　　D、 Head(Tail(Head(Tail(Tail(Tail(L))))))
33. 树最适合用来表示____B_____
A、 有序得数据元素　　B、 数据之间具有分支层次关系得数据
C、 无序得数据元素　　D、 无太多关系得数据元素
34. 如果某二叉树得前根次序遍历结果为stuwv,中序遍历为uwtvs，那么该二叉树得后序为____B___。
A、 uwvts　　B、 vwuts　　C、 wuvts　　D、 wutsv
35. 某二叉树得前序遍历结点访问顺序就是abdgcefh，中序遍历得结点访问顺序就是dgbaechf,则其后序遍历得结点访问顺序就是__D__。
A、 bdgcefha　　B、 gdbechfa　　C、 bdgaechf　　D、 gdbehfca
36. 在一非空二叉树得中序遍历序列中，根结点得右边____A__。
A、 只有右子树上得所有结点　　B、 只有右子树上得部分结点
C、 只有左子树上得部分结点　　D、 只有左子树上得所有结点
37. 设m与n就是一棵二叉树上得两个结点，在中序遍历，　n在m前得条件就是__C__
A、 n在m得右方　　B、 n就是m得祖先
C、 n在m得左方　　D、 n就是m得子孙
38. 深度为5得二叉树至多有___C__个结点。
A、 16　　B、 32　　C、 31　　D、 10

39. 由权（8, 2 ， 5, 7)得四个叶子结点构造一棵哈夫曼树, 该树得带权路径长度为__D____

A、 2 3 B、 3 7 C、 46 D、 4 3

40. 利用二叉链表存储树， 则根结点得右指针就是___C____

A 、 指向最左孩子 B、 指向最右孩子 C、 空 D、 非空

41. 下列存储方式中， 哪一个不就是树得存储形式？ ___D____

A、 双亲表示法 B、 孩子链表表示法 C、 孩子兄弟表示法 D 、 顺序存储表示法

42. 在一个无向图中，所有顶点得度数之与等于所有边数得__C___倍。

A 、 1 / 2 B 、 1 C、 2 D、 4

43. 具有 n 个顶点与多于 n-1 条边得无向图___B___、

A、 有可能就是树 B 、 一定不就是树 C 、 一定就是树 D、 以上答案都不对

44. 具有 6 个顶点得无向图至少应有__A__条边才能确保就是一个连通图。

A、 5 B、 6 C、 7 D、 8

45. 无向图 G=（V，E），其中： V={a，b， c ，d,e，f}， E={(a,b)，(a ,e)，(a ,c)， (b ， e)，(c ， f) ,(f，d)，(e,d)}， 则对该图进行深度优先遍历， 得到得序列为：____D____

A、 ab e cdf B 、 a cfe b d C、 aebcfd D、 ae d f c b

46. 下述几种排序方法中，要求内存量最大得就是__D___.

A、 插入排序 B、 选择排序 C、 快速排序 D 、 归并排序

47. 排序方法中，从未排序序列中依次取出元素与已排序序列(初始时为空）中得元素进行比较,将其放入已排序序列得正确位置上得方法，称为___C__。

A、 希尔排序 B 、 起泡排序 C 、 插入排序 D、 选择排序

48. 在待排序得元素序列基本有序得前提下，效率最高得排序方法就是__A__。

A. 插入排序 B、 选择排序 C、 快速排序 D 、 归并排序

49. 下列排序算法中， 哪一个就是稳定得排序算法? ___B____

A、 直接选择排序 B、 二分法插入排序 C、 希尔排序 D、 快速排序

50. 将两个各有 n 个元素得有序表归并成一个有序表, 其最少得比较次数___A__

A、 n B、 2n—1 C、 2 n D、 n-1

二、填空题

1. 算法得五个重要特性就是__有穷性，确定性，可行性,输入与输出、
2. 数据得树型结构与图（网）状结构合称__非线性结构__、
3. 抽象数据类型得定义仅取决于它得一组__逻辑特性__，而与__数据在计算机中得表示与实现__无关、
4. 评价算法质量得指标就是__正确性，易读性，健壮性,高效性、
5. 数据结构中评价算法得两个重要指标就是:__时间复杂度与空间复杂度、
6. 分析下面算法（程序段）,得时间复杂度就是__O (mn)__。

```
s =0;
f o r  （i＝0；i＜ n ;i+＋）
  fo r （j =0; j ＜m;j++)
      s+= B［i][j];
```

7. 当线性表元素得总数基本稳定，且很少进行删除与插入操作时，但就是要求以最快得速度存取线性表中得元素，应该采取__顺序__存储结构、
8. 顺序表中逻辑上相邻得元素得物理位置__必定__相邻，而单链表中逻辑上相邻得元素得物理位置__不一定__相邻、
9. 在各个结点查找概率相等得情况下，从 n 个结点得单链表中查找一个结点，平均要访问__n ／2__个结点、
10. 在单链表中设置头指针得作用就是：__简化操作，减少边界条件得判断、
11. 在单链表中，除首元结点外，任一结点得存储位置由__其直接前驱得指针域__指示、
12. 对于一个具有 n 个结点得单链表，在已知 p 所指向结点后插入一个新结点得时间复杂度就是__O（1）__，在值域为给定值得结点后插入一个新结点得时间复杂度为 O(n)、
13. 在双链表中,每个结点有两个指针域，一个指向__前驱结点__,另一个指向__后继结点__.
14. 根据线性表得链式存储结构中每一结点包含得指针个数，将线性表分成__单链表__与__多重链表、
15. 在非空双向链表中，在结点 q 得前面插入结点 p 得过程如下，请补充

```
  p －＞pr i or＝ q —>prior;
q-〉 pr i or—>ne x t= p ;
p—〉 n e xt=q;
q—＞ p rior＝p;
```

16. 一般情况下， 将递归算法转换成等价得非递归算法应该设置___栈___、
17. 在解决计算机主机与打印机速度不匹配问题时，通常设置一个打印数据缓冲区， 该缓冲区通常就是一个_____队列_____数据结构、
18. 循环队列得引入， 目得就是为了克服___假溢出___现象、
19. 在栈顶指针为 HS 得链栈中， 判断栈空得条件就是____HS=NULL 、
20. 在具有 n 个单元得循环队列中， 如果不专门设置队满标志， 则队满时共__n—1__有个元素、
21. 实现字符串拷贝得函数如下， 请补足

    Void strcpy（char ＊s, char ＊t)

    {while（_____（＊s＋＋=＊t++)!='\0'_____）;}
22. 空格串就是____由一个或多个空格字符组成得串____，其长度等于__其包含得空格个数__.
23. 空串就是___不包含任何字符得串___， 其长度为__0__、
24. 设 s=′I__AM__A__STUDENT′， 其长度为:____14____、
25. 组成串得元素只能就是：__字符__、
26. 设 s1=’Good’， s2=′ ′， s3='bye!’， 则 s1,s2 与 s3 连接得结果就是_Good bye!
27. 若广义表中每个元素都就是原子时， 广义表便成为__线性表__、
28. 广义表得表尾就是指除第一个元素外，____剩余元素组成得表__、
29. 广义表 A=(（a，b，c，d))得表头为__（a，b，c，d）__，表尾为 ___()___、
30. 数组得存储结构采用___顺序___存储方式、
31. 设二维数组 a［0、、5， 0、、6］， 其每个字节占 5 个字节， 第一个元素得存储地址为 1000， 若按列存储， 则元素 a[5，5］存储地址为__1175__、
32. 高度为 k 得完全二叉树至少有_____个叶子结点、
33. 若一棵二叉树有 50 个叶子结点，则该二叉树得总结点数至少就是_99、
34. 有 n 个叶子结点得哈夫曼树得结点总数为__2n—1__、
35. 根据二叉树得定义， 具有三个结点得二叉树有___4___种、
36. 某棵二叉树得中序遍历序列为 a bcdefg，后序遍历序列为 bdcaf g e，则该二叉树得前序遍历序列__ea c bdgf，该二叉树对应得森林包含 ___2___棵二叉树、
37. 若二叉树采用二叉链表存储结构， 要交换其所有分支结点得左，右子树得位置，利用_____中序___遍历方法最为合适、

38. 线索二叉树得左线索指向其__前驱__，右线索指向其_后继__、
39. 树所对应得二叉树其根结点得__右__子树一定为空、
40. 利用树得孩子兄弟表示法存储， 可以将一棵树转化成__二叉树、
41. 设无向图得顶点个数为 n， 则该图最多有_____n(n—1)/2_条边、
42. n 个顶点得连通图至少有_____n－1___条边、
43. 已知一个图用领接矩阵表示， 计算第 i 个结点得入度得方法就是__求第 i 列非零元素得与__、
44. G 就是一个非连通得无向图， 共有 28 条边， 则该图至少有__9___个顶点、
45. 一个图得__邻接矩阵__表示法就是唯一得，而___邻接表_表示法就是不唯一得。
46. 从邻接矩阵可以瞧出， 该图共有____3_____个顶点， 如果就是无向图， 则共有____2__条边、
47. n 个顶点得连通图用邻接矩阵表示时，则该矩阵至少有__2（n-1)_个顶点、
48. 设图中有 n 个顶点，e 条边， 如果用邻接表表示图， 进行深度优先搜索遍历得时间复杂度为_O(n＋e）_，如果用邻接矩阵表示图，时间复杂度为
49. 从平均时间性能而言，_____快速排序_____排序最佳、
50. 堆排序就是一种__选择__排序，堆实质上就是 _一棵完全二叉树_结点得层次序列、 对于含有 n 个元素得排序， 堆排序得时间复杂度为 __、 所需附加得存储结点就是_O(1）__、

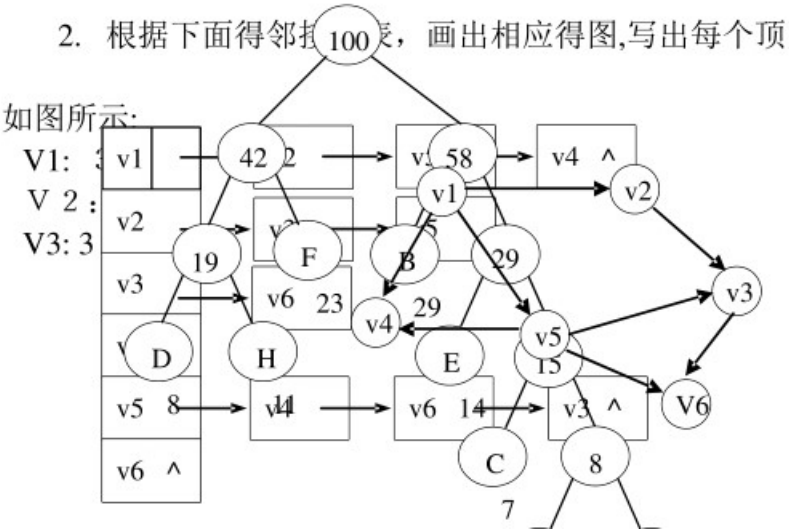
三、用图表回答下列问题

1. 设某通信系统使用 A，B ，C，D， E，F,G， H 个字符，出现得频率 w={5，29，7， 8,14， 23,3，11},试构造对应得哈夫曼树(请按左子树根结点得权小于右子树树根结点得权得次序构造)?

答案如图：

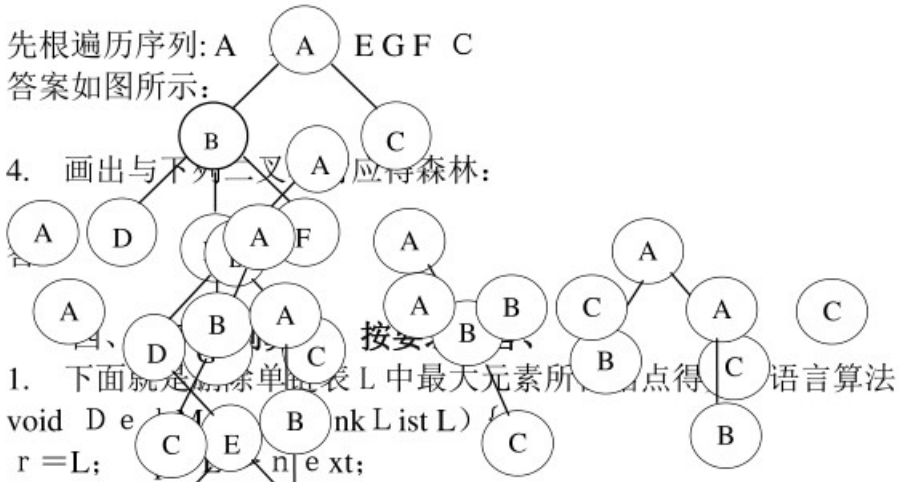
2. 根据下面得邻接表，画出相应得图,写出每个顶点得度， 并用邻接矩阵表示、

答案如图所示：



V 4 ： 2
V5: 4
V6： 2

3. 画出下列树对应得二叉树，并写出其先根遍历序列:



先根遍历序列: A E G F C
答案如图所示：

4. 画出与下列二叉 应得森林：

按要求回答下列问题。

1. 下面就是删除单链表 L 中最大元素所在结点得 语言算法， 请补足缺失部分使其完整、

```
void  D e l M a x n k L ist L）{
 r =L;  n e xt;
 if（p）{
   m= p  d at a ;  ____（1）____ ;  p=p-〉 n ext;
   whil e (p) {
     i f （____（2）____）
     {____（3）____ ; m=p—〉 data; }
      ____（4）____ ; p=p—>next;
     }
     q=r—>n e xt; ____（5）____ ;  f ree（q）;
 }
```

```
}
```
答案：（1）L->next=NULL；(2)p!=NULL;（3）q!=NULL；（4）p->next=r—>next
（5）r—>next=p、

2. 阅读下列算法,说明该算法得作用。

```
Status algorithm1(LinkQueue &Q){
    SqStack◦ Stack;
    QElemType◦Element;
    InitStack（Stack）;
    while（! QueueEmpty(Q)）{
    ◦DeQueue（Q， Element);
    ◦Push（Stack， Element);
    }
    while(!StackEmpty(Stack)){
        Pop(Stack,Element);
◦   EnQueue(Q,Element);
◦}
}
```

答：利用栈实现队列得逆置、

3. 阅读下列算法,说明该算法得作用。

```
Status algorithm2(Stack S， int e){
    Stack T;
◦int◦d;
    InitStack（T);
◦while（! StackEmpty(S)）{
   ◦Pop（S，d）;
◦◦if(d!=e) Push(T，d);
```

```
}
    whi l e (! S tackEm p ty（T）) {
    Pop（T， d ）;
Push( S ， d);
    }
}
```

答： 利用辅助栈 T， 将栈 S 中得元素 e 删除、

4. 将下面程序改写成递归过程、

```
voi d  alg o r it h m3 （ i nt  n ）{
int i=n;
        while（i〉 1 ) {
pr i n t f(i-－);
}
}
```

答:

```
voi d  algo r ithm4（in t  j){
     if（ j 〉 1) {
          p ri n tf( j );
          alg o rithm4（j—1）
     }
}
```

5. 阅读下列算法，说明该算法得作用。

```
B i  T r e e  algorithm5（Elem T y pe P r e[], ElemT y p e In[］）{
    int  P r e Le n ， I n L e n;
int i, j ;
    B i Tree  B T;
```

```
ElemType *subPre, *subIn;
    PreLen = strlen(Pre);
InLen = strlen(In);
    if (PreLen != InLen || PreLen == 0) return NULL;
for (i=0; i<InLen && In[i]!=Pre[0]; i++);
    if (i == InLen) return NULL;
    BT = (BiTNode*) malloc (sizeof(BiTNode));
    BT—〉data = Pre[0];
    subPre = (ElemType*) malloc ((i+1)*sizeof(ElemType));
subIn = (ElemType*) malloc((i+1)*sizeof(ElemType));
for (j=0; j<i; j++) {
    subPre[j] = Pre[j+1];
        subIn[j] = In[j];
}
subPre[j] = '\0'; subIn[j] = '\0';
    BT—〉lchild = CreatBT(subPre, subIn);
subPre = (ElemType*) malloc((PreLen—i)*sizeof(ElemType));
    subIn = (ElemType*) malloc((PreLen—i)*sizeof(ElemType));
for (j= i+1; j〈PreLen; j++) {
subPre[j-i—1] = Pre[j];
    subIn[j—i-1] = In[j];
    }
    subPre[j—i-1] = '\0'; subIn[j—i—1] = '\0';
    BT—> rchild = CreatBT(subPre, subIn);
return BT;
}
```

答：　利用一棵二叉树得先序遍历与中序遍历还原该二叉树、

五、算法设计题

1. 设顺序表L中得数据元素递增有序、 试写一个算法, 将e插入顺序表中， 要求插入后保持该表得有序性、

```
void InsertElem(SqList &L, ElemType){
   j=L、length-1;
  while(L、elem[j]>e){
    L、elem[j+1]=L、elem[j];
    j++
   }
  L、elem[j+1]=e;
  L、length++;
}
```

2. 已知la就是带头结点得单链表得头指针, 试编写一个逆序输出表中各个元素得递归算法、

```
void TraverseLink(LinkList p){
   if (p—>next) TraverseLink(p->next);
  printf(p—〉data);
}
```

3. 写一算法,统计二叉树得结点得总个数。

```
//利用中序遍历方法,或者先序、后序均可以
void Leaf(BiTree T, int &m) {
  if (T!=NULL) {
      Leaf(T-〉lchild, m);        //中根遍历左子树
      m++;                       //计算结点
      Leaf(T—〉rchild, m);       //中根遍历右子树
   }
}
```

4. 写一算法，求二叉树得高度。

```
int height (BiTree T) {
    if (T == NULL) return 0;
      else {
        int hl=height (T->lchild);
       int hr=height (T—〉 rchild);
       return 1 + (hl> hr?hl:hr);    //1 加上 hl 与 hr 得较大值
       }
}
```

5. 写一算法，交换二叉树得左右子树。

```
BiTNode Permute_child (BiTNode *T){
        BiTNode *temp;
      if (T){
            Permute_child (T—>lchild);
             Permute_child(T—〉rchild);
            temp=T-〉lchild;
            T-〉 lchild=BT-〉rchild;
             T->rchild =temp;
       }
}
```

6. 已知二叉树采用二叉链表存储， 设计算法， 判定两棵二叉树就是否相似、

```
Statues like(BiTree S,    BiTree T) {
       if (T==NULL && S==NULL) return 1;
       else if(S && T)
            return (like(S—>lchild ,  T-〉lchild) &&like(S—〉rchild , T-〉rchild));
      else  return 0;
```

}