复习资料

《数据结构》

第一大题：单项选择题

1、线性表的链式存储结构是一种( 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

2、算法分析的目的是( C )

A.找出数据结构的合理性

B.研究算法中的输入和输出的关系

C.分析算法的效率以求改进

D.分析算法的易懂性和文档性

3、在循环双向链表的p所指结点之后插入s所指结点的操作是( D )

A.p->right=s;s->left=p;p->right->left=s;s->right=p->right;

B.p->right=s;p->right->left=s;s->left=p;s->right=p->right;

C.s->left=p;s->right=p->right;p->right=s;p->right->left=s;

D.s->left=p;s->right=p->right;p->right->left=s;p->right=s;

4、从一个具有n个结点的单链表中查找其值等于x结点是，在查找成功的情况下，需要比较 ( D )个结点

A.n

B.n/2

C.C (n-1)/2

D.(n+1)/2

5、在一个链对中，假设f和r分别为队首和队尾指针，则删除一个结点的运算是( C )

A.r=f->next

B.r=r->next;

C.f=f->next;

D.f=r->next;

6、设串s1=‘ABCDEFG’，s2=‘PQRST’，函数con（x，y）返回x和y串的连接串，subs（s，i，j）返回串s的从序号id字符开始的j个字符组成的子串，len(s)返回串s的长度。则con(subs(s1，2，len(s2))，subs(s1，len(s2)，2))的结果是( D )

A.BCDEF

B.BCDEFG

C.BCPQRST

D.BCDEFEF

7、数组的两种基本操作是( C )

A.建立与删除

B.索引和修改

C.查找和修改

D.查找与索引

8、二维数组M的元素是4个字符（每个字符占一个存储单元）组成的串，行下标i的范围是0到4,列下标j的范围是0到5,M按行存储时元素M[3][5]的起始地址与M按列存储时的元素( B )的起始地址相同

A.M[2][4]

B.M[3][4]

C.M[3][5]

D.M[4][4]

9、采用顺序查找方法查找长度为n的线性表时，每个元素的平均查找长度为( C )

A.n

B.n/2

C.(n+1)/2

D.8(n-1)/2

10、线性表若采用链式存储结构时，要求内存中可用存储单元的地址( D )

A.必须是连续的

B.部分地址是连续的

C.一定不是连续的

D.连续不连续都可以

11、不带头节点的单链表head为空的判定条件是( A )

A.head==NULL

B.head->next==NULL

C.head->next==head

D.head!=NULL

12、带头节点的单链表head为空的判定条件是( B )

A.head==NULL

B.head->next==NULL

C.head->next==head

D.head!=NULL

13、在一个单链表中，已知q所指结点是p所指结点的前驱结点，若在q和p之间插入s结点，则执行( C )

A.s->next=p->next;p->next=s;

B.p->next=s->next;s->next=p;

C.q->next=s;s->next=p;

D.p->next=s;s->next=q;

14、串是一种特殊的线性表，其特殊性体现在( B )

A.可以顺序存储

B.数据元素是一个字符

C.可以连续存储

D.数据元素可以使多个字符

15、数组A中，每个元素A的长度为3个字节，行下标i从1到8,列下标j从1到10，从首地址SA开始连续存在存储器中，该数组按列存放时，元素A[5][8]的起始地址为( B )

A.SA+141

B.SA+180

C.SA+222

D.SA+225

16、二维数组M的成员是6个字符（每个字符占一个存储单元）组成的串，行下标i的范围是从0到8,列下标j的范围是1到10，M的第8列和第5行共占( A )字节

A.108

B.114

C.54

D.60

17、在线索二叉树中，t所指结点没左子树的充要条件是( B )

A.t->left==NULL

B.t->ltag==1

C.t->ltag==1 and t->left==NULL

D.以上都不对

18、已知某二叉树的后序遍历序列是dabec,中序遍历是debac，它的前序遍历序列是( D )

A.acbed

B.decab

C.deabc

D.cedba

19、如果T2是由有序树T转换而来的二叉树，那么T中结点的前序就是T2中结点的( A )

A.前序

B.后序

C.中序

D.层次序列

20、对于一个具有n个顶点和e条边的无向图，若采用邻接表表示，则表头向量的大小为( A )

A.n

B.n+1

C.n-1

D.n2

21、有一个有序表为（1,3,9,12,32,41,45,62,75,77,82,95,100），当二分查找值为82的结点是，( C ) 次比较后查找成功

A.1

B.2

C.4

D.8

22、设哈希表长m=14，哈希函数H（key）=key%11，表中已有4个结点：addr(15)=4,addr(38)=5,addr(61)=6,addr(84)=7，其余地址为空，若用二次探测再散列处理冲突，关键字为49的结点的地址是 ( D )

A.8

B.3

C.5

D.9

23、在以下的叙述中，正确的是( B )

A.线性表的线性存储结构优于链表存储结构

B.二维数组是数据元素为线性表的线性表

C.栈的操作方式是先进先出

D.队列的操作方式是先进后出

24、线性结构的顺序存储结构是一种( A )存储结构

A.随机存取

B.顺序存取

C.索引存取

D.散列存取

25、非空的循环单链表head的尾结点（有p所指向）满足( C )

A.p->next==NULL

B.p==NULL

C.p->next=head

D.p==head

26、在一个单链表中，若p所指结点不是最后结点，在p之后插入s所指结点，则执行( B )

A.s->next=p;p->next=s;

B.s->next=p->next;p->next=s;

C.s->next=p->next;p=s;

D.p->next=s;s->next=p;

27、在一个单链表中，若删除p所指结点的后续结点，则执行( A )

A.p->next=p->next->next;

B.p=p->next;p->next=p->next=next;

C.p->next=p->next;

D.p=p->next->next;

28、设有连个串p和q，求q在p中首次出现的位置的运算称作( B )

A.连接

B.模式匹配

C.求子串

D.求串长

29、数组A中，每个元素A的长度为3个字节，行下标i从1到8,列下标j从1到10，从首地址SA开始连续存在存储器中，该数组按行存放时，元素A[8][5]的起始地址为( C )

A.SA+141

B.SA+144

C.SA+222

D.SA+225

30、稀疏矩阵一般的压缩存储方法有两种，分别是( C )

A.二维数组和三维数组

B.三元数组和散列

C.三元数组和十字链表

D.散列和十字链表

31、维数组M的成员是6个字符（每个字符占一个存储单元）组成的串，行下标i的范围是从0到8,列下标j的范围是1到10，若按行优先方式存储，元素M[8][5]的起始地址与当M按列优先方式存储的 ( B )元素的起始地址一致

A.M[8][5]

B.M[3][10]

C.M[5][8]

D.M[0][9]

32、如果T2是由有序树T转换而来的二叉树，那么T中结点的后序就是T2中结点的( B )

A.前序

B.后序

C.中序

D.层次序列

33、二叉树的前序遍历结点顺序是abdgcefh，中序遍历的结点顺序是dgbaechf，则其后序遍历结点顺序是( D )

A.bdgcefha

B.gdbecfha

C.bdgaechf

D.gdbehfca

34、按照二叉树的定义，具有3个结点的二叉树有( C )种

A.3

B.4

C.5

D.6

35、在一个有向图中，所有顶点的入度之和等于所有顶点的出度之和的( B )倍

A.1/2

B.1

C.2

D.4

36、有一个长度为12的有序表，按二分法查找，在表内个元素等概率情况下查找成功所需的平均比较次数为( B )

A.35/12

B.37/12

C.39/12

D.43/12

37、如果要求一个线性表既能较快查找，又能适应动态变化的要求，可以采用( A )查找

A.分块

B.顺序

C.二分

D.散列

38、数组A中，每个元素A的长度为3个字节，行下标i从1到8,列下标j从1到10，从首地址SA开始连续存在存储器中，存放该数组至少需要的单元数是( C )

A.80

B.100

C.240

D.270

39、如果二叉树的前序为stuwv，中序为uwtvs，则后序为( C )

A.uwvts

B.vwuts

C.wuvts

D.wutsv

40、对一个满二叉树，有m个叶子，n个结点，深度为h，则 ( D )

A.n=h+m

B.h+m=2n

C.m=h-1

D.n=2h-1

41、具有五层结点的二叉平衡树至少有( C )个结点

A.10

B.12

C.15

D.17

42、在一个图中，所有顶点的度数之和等于所有边数的( C )倍

A.1/2

B.1

C.2

D.4

43、采用分块查找时，若线性表共有625个元素，查找每个元素的概率相同，假设采用顺序查找来确定结点所在的块时，每块应分( B ) 个结点最佳

A.10

B.25

C.6

D.625

44、在数据结构中，从逻辑上可以数据结构分成( C )

A.动态结构和静态结构

B.紧凑结构和非紧凑结构

C.线性结构和非线性结构

D.内部结构和外部结构

第二大题：填空题

1、在各种查找方法中，平均查找长度与结点个数n无关的查找方法是(哈希查找 )

2、已知一个图的邻接矩阵表示，删除所有从i个结点出发的边的方法是( 将邻接矩阵的第i行全部置0)

3、对n个元素的序列进行冒泡排序，最少的比较次数( 0)

4、在插入和选择排序中，若初始数据基本正序，则选择(插入排序 )

5、已知二维数组A[10...20][5...10]采用行序为主方式存储，每个元素占4个存储单元，并且A[10][5]的存储地址是1000，这A[18][9]的地址是(1208 )

6、空格串的长度等于(空格个数)

7、可以使用( )表示树形结构

8、串的两种基本存储方式是 链接存储 和(顺序存储 )

9、广义表(((a)))的表头是( )

10、下面程序的时间复杂度(n )

i=s=0;

while(s<n)

{

i++;

s+=i;

}

11、在栈顶指针为HS链栈中，判定栈空的条件是( HS==NULL)

12、单链表是(线性表 ) 链接存储表示

13、广义表((a),((b),c),(((d))))的表头是( )

14、已知二维数组A[10][20]采用列序为主方式存储，每个元素占一个存储单元，并且第一个元素的存储地址是200，这A[6][12]的地址是( 326)

15、二分查找的存储结构仅限于(顺序存储结构 )

16、已知一个图的邻接矩阵表示，计算第i个结点的入度方法是(第i列元素之和 )

17、在插入排序、希尔排序、选择排序、快速排序、堆排序中，稳定的排序有( )

18、在堆排序中，快速和归并排序中，若只从平均情况下排序最快考虑，应选取( )方法

19、下面程序的时间复杂度(O(n2) )

s=0;

for(i=0;i<n;i++)

for(j=0;j<n;j++)

s+=B[i][j];

sum=s;

20、非空的循环单链表head的尾结点（由p所指向），满足条件( p->next==head)

21、在双链表中，每个结点有两指针域，一个指向(前续 )结点，一个指向后续结点

22、两个串相等的充分必要条件是(两个串长度相等且对应位置字符相同 )

23、广义表(a,(a,b),d,e,((i,j),k))的长度为( )

24、已知二维数组A[m][n]采用行序为主方式存储，每个元素占k个存储单元，并且第一个元素的存储地址是LOC(A[0][0])，这A[i][j]的地址是( LOC(A[0][0]) + (i \* n + j) \* k)

25、在分块查找中，首先查找( )

26、在无向图G的领结矩阵A中，若A[i][j]等于1，则A[j][i]等于(1 )

27、在利用快速排序方法对记录（54,38,96,23,15,72,60,45,83）进行排序时，递归调用二使用的栈所能达到的最大深度为( )

28、在对一组记录（54,38,96,23,15,72,60,45,83）进行直接插入排序时，当把第7个记录60插入到有序表时，为寻找插入位置需要比较( 3)次

29、下面程序的时间复杂度(m\*n )

for(i=0;i<n;i++)

for(j=0;j<m;j++)

A[i][j]=0;

30、在HQ的链队中，判定只有一个结点的条件是(HQ->next==NULL )

第三大题：应用题

1、设向量有A，B，C，其中A有元素n个且任何元素都不为0。阅读后补全程序，并回答问题。

void f(vector A,int n, vector B,int \*p, vector C,int \*q)

{

int i;

\*p=0;\*q=0;

for(i=0;i<=n-1;i++)

{

if(A[i]>0)

{

\_\_\_\_\_（1）\_\_\_\_\_ ;

B[\*p]=A[i];

}

if(A[i]<0)

{

\_\_\_\_\_（2）\_\_\_\_\_;

C[\*q]=A[i];

}

}

}

答案

（1）\_(\*p)++\_\_\_\_\_\_\_\_\_

（2）\_(\*q)++\_\_\_\_\_\_\_

（3）该程序的功能是什么？

将A中大于0的元素放入B集合，小于0的元素放入C集合，并返回B，C集合各放入元素的个数

2、

阅读下面代码，分析程序实现的功能

typedef struct tnode

{

ElemType data;

int ltag,rtag;

struct tnode \*left,\*right;

}tbtree;

void f(tbtree \*p, tbtree \*pre)

{

if(p!=NULL)

{

f(p->left,pre);

if(p->left==NULL)

{

p->ltag=1;

p->left=pre;

}

else \_\_\_\_\_（1）\_\_\_\_\_ ;

if(pre!=NULL && pre->right==NULL)

{

pre-rtag=1;

pre->right=p;

}

else \_\_\_\_\_（2）\_\_\_\_\_;

pre=p;

f(p->right,pre);

}

}

1． \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2． \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3．该程序的功能是什么？

答案：

（1）p->ltag = 0;

（2）pre->rtag = 0;

（3）二叉树中序线索化

3、

阅读下面代码，分析程序实现的功能

typedef struct bnode

{

ElemType data；

struct bnode \*left，\*right；

}btree；

btree \*f(btree \*root,btree \*p,btree \*p)

{

btree \*s[Maxsize],\*anor[Maxsize],\*b,\*r;

int tag[Maxsize],find=0;

int top=0;

b=root;

int i,k,j;

do

{

while(\_\_\_\_\_\_\_（1）\_\_\_\_\_\_\_)

{

top++;

s[top]=b;

tag[top]=0;

b=b->left;

}

if(top>0)

{

if(tag[top]==1)

{

if(s[top]==p)

for(i=1;i<=top;i++)

anor[i]=s[i];

if(s[top]==q)

{

j=top;

while(!find)

{

k=i-1;

while(k>0 && s[j]!=anor[k]) k--;

if(k>0)

{

find=1;

r=anor[k];

}

else

j--;

}

}

top--;

}

else

{

b=s[top];

if(top>0 && !find)

{

b=b->right;

tag[top]=1;

}

}

}

}while(!find && (b!=NULL || top!=0));

\_\_\_\_\_\_\_（2）\_\_\_\_\_\_\_ r;

}

(1) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(2) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(3) 该程序的功能是什么？

答案：

*（1）*b!=NULL

*（2）return*

*（3）计算p和q两结点的共同最近的祖先*

4、

阅读下面代码，分析程序实现的功能

typedef struct linknode

{

ElemType data；

struct linknode \*next；

}node；

void f(node \*h1,node \*h2)

{

node \*p,\*q,\*s;

h2=(node \*)malloc(sizeof(node));

q=h2;p=h1;

while(\_\_\_\_\_\_\_\_\_\_（1）\_\_\_\_\_\_\_\_\_\_ )

{

s=(node \*)malloc(sizeof(node));

s->data=p->data;

q->next=s;

q=s;

p=p->next;

}

q->next=NULL;

\_\_\_\_\_\_\_\_\_\_（2）\_\_\_\_\_\_\_\_\_\_;

h2=h2->next;

free(p);

}

（1） \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

（2） \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

（3） 该程序的功能是什么？

5、

阅读完成下列程序，并回答问题

typedef struct bnode

{

ElemType data；

struct bnode \*left，\*right；

}btree；

void f(btree \*b)

{

btree \*stack[MAXSIZE], \*p;

int top=0;

p=b;

do

{

while(\_\_\_\_\_\_\_\_\_\_（1）\_\_\_\_\_\_\_\_\_\_ )

{

top++;

stack[top]=p;

p=p->left;

}

if(top>0)

{

\_\_\_\_\_\_\_\_\_\_（2）\_\_\_\_\_\_\_\_\_ ;

top--;

printf("%d ",p->data);

p=p->right;

}

}while(p!=NULL || top!=0)

}

(1) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(2) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(3) 该程序的功能是什么？

答案：

（1）p != *NULL*

（2）p = stack[top];

（3）二叉树前序遍历

6、

阅读下列程序，回答问题

#define MAXITEM 100

struct element

{

keytype key;

elementtype data;

};

typedef struct sqlist[MAXITEM]

int f(sqlist r, int k,int n)

{

int i=0;

while(i<n && r[i].key!=k) i++;

if(i>n) i=-1;

return(i);

}

该程序的功能是什么？

答案：找到数组中第几个元素的key为k

7、

阅读下列程序，回答问题

#define MAXITEM 100

struct element

{

keytype key;

elementtype data;

};

typedef struct sqlist[MAXITEM]

int f(sqlist r, int k,int n)

{

int i,low=0,high=n-1,mid,find=0;

while(low<=high && !find)

{

mid=(low+high)/2;

if(k<r[mid].key) high=mid-1;

else if(k>r[mid].key) low=mid+1;

else

{

i=mid;

find=1;

}

if(!find) i=-1;

return(i);

}

}

该程序的功能是什么？

：答案：找到数组中第几个元素的key为k

8、

阅读完成下列程序，并回答问题

#define MAXITEM 100

struct element

{

keytype key;

elementtype data;

};

typedef struct sqlist[MAXITEM]

void f(sqlist r,int n)

{

int i,j;

struct element w;

for(i=1;i<=n-1;i++)

for(j=n;j>=i+1;j--)

if(r[j].key<r[j-1].key)

{

w=r[j];

\_\_\_\_\_\_\_\_\_\_（1）\_\_\_\_\_\_\_\_\_\_=r[j-1];

r[j-1]= \_\_\_\_\_\_\_\_\_\_（2）\_\_\_\_\_\_\_\_\_\_;

}

}

（1） \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

（2） \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

（3）该程序的功能是什么？

答案：

（1）r[j]

（2）w

（3）将数组按元素的key升序排列

9、设向量有A，B，C，其中A有个m个元素，B有n个元素。

阅读完成下列程序，并回答问题。

int link(vector a,int m,vector b,int n,vector c)

{

int i=0,j=0,l,k=0;

while(i<m && j<n)

{

if(a[i]<b[j])

c[k++]=a[i++];

else if(a[i]>b[j])

\_\_\_\_\_\_\_\_\_\_（1）\_\_\_\_\_\_\_\_\_\_;

else

{

c[k++]=b[j++];

i++;

}

}

if(i==m)

for(l=j;l<n;l++)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_（2）\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ;

if(j==n)

for(l=i;l<m;l++)

c[k++]=b[l];应该是c[k++]=a[l]

return k;

}

（1） \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

（2） \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

（3） 该程序的功能是什么？

答案：

（1）c[k++]= b[j++];

（2）c[k++] = b[l];

（3）将a,b两个集合按顺序合并到c集合中

10、

阅读完成下列程序，并回答问题。

typedef struct bnode

{

ElemType data；

struct bnode \*left，\*right；

}btree；

int f(btree \*b)

{

int num1,num2;

if(\_\_\_\_\_\_\_\_\_\_\_\_（1）\_\_\_\_\_\_\_\_\_\_\_\_ )

return 0;

else

{

num1=f(b->left);

num2=f(b->right);

if(b->left!=NULL && b->right!=NULL)

return (num1+num2+1);

else

return (\_\_\_\_\_\_\_\_（2）\_\_\_\_\_\_\_\_);

}

}

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.上述代码完成什么任务？

答案：

（1）b!=NULL

（2）0

（3）统计树中左右节点都不为空的节点

11、阅读完成下列程序，并回答问题

type struct bnode

{

ElemType data；

struct bnode \*left，\*right；

}btree；

int height(btree \*t)

{

int he,he1,he2;

if(\_\_\_\_\_\_\_\_\_\_ (1)\_\_\_\_\_\_\_\_\_\_ ) return(0);

else

{

he1=height(t->left);

he2=height(t->right);

if(he1>he2)

he=he1+1;

else

\_\_\_\_\_\_\_\_\_\_ (2)\_\_\_\_\_\_\_\_\_\_\_\_;

return(he);

}

}

(1) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(2) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(3) 上述程序完成什么任务？

答案：

（1）t==*NULL*

（2）he = he2 + 1;

（3）统计树的层级

第四大题：程序设计题

1、假设二叉树采用链表存储结构，设计一个算法计算二叉树中指定结点的层数的程序（使用递归算法）

函数原型：void level（btree \*b， btree \*p，int \*h， int h1）

/\*

h返回p所指定结点的高度，初始值为-1，树为空返回0；

h1指示b树高度，初始值为1\*/

type struct bnode

{

ElemType data；

struct bnode \*left，\*right；

}btree；

void level(btree\* b, btree\* p, int\* h, int h1)

{

if (!b||!p)

{

return;

}

if (b->data == p->data)

{

\*h = h1;

return ;

}

level(b->left, p, h, h1 + 1);

level(b->right, p, h, h1 + 1);

}

2、设计一个程序实现深度优先搜索（使用递归算法）

函数原型：int dfs（adjlist adj, int v）

#include <stdio.h>

#define MAXVEX 30

struct edgenode

{

int adjvex;

char info;

struct edgenode \*next;

};

struct vexnode

{

char data;

struct edgenode \*link;

};

typedef struct vexnode adjlist[MAXVEX];

int visited[MAXVEX];

int dfs(adjlist adj, int v)

{

visited[v] = 1;

edgenode \*p;

p = adj[v].link;

while (p != *NULL*)

{

if (adj[v].link && !visited[p->adjvex])

{

dfs(adj, p->adjvex);

}

p = p->next;

}

}

3、设计一个程序实现宽度（广度）优先搜索（使用递归算法）

函数原型：int bfs（adjlist adj, int v）

#include <stdio.h>

#define MAXVEX 30

struct edgenode

{

int adjvex;

char info;

struct edgenode \*next;

};

struct vexnode

{

char data;

struct edgenode \*link;

};

typedef struct vexnode adjlist[MAXVEX];

int visited[MAXVEX];

int queue[MAXVEX];

void BFS(adjlist adj, int i)

{

if (i>= MAXVEX)

{

return;

}

int front, rear;

int temp;

front = 0;

rear = 0;

visited[i] = 1;

queue[rear++] = i;

edgenode\* p;

while (rear > front) {

temp = queue[front++];

p = adj[temp].link;

while (p) {

if (!visited[p->adjvex]) {

queue[rear++] = p->adjvex;

visited[p->adjvex] = 1;

}

p = p->next;

}

}

BFS(adj, i + 1);

}

第五大题：综合题

1、

已知序列{17,18,60,40,7,32,73,65,85}，请给出采用冒泡排序法对该序列排序时每一趟的结果

（1）7,18,60,40,17,32,73,65,85,

（2）7,17,60,40,18,32,73,65,85

（3）7,17,18,60,40,32,73,65,85

（4）7,17,18,32,60,40,73,65,85

（5）7,17,18,32,40,60,65,73,85

2、对关键字序列(15,22,10 13 30, 16,12,17)按从小到大进行快速排序

(1)写出排序过程中前两趟的划分结果；

(2)快速排序是否是稳定的排序方法?

答案：

（1）[12,13,10],15,[30,16,22,17]

（2）不稳定

3、已知序列{503,87,512,61,908,170,897,275,653,462}，请给出采用快速排序法对该序列做升序排序时的每一趟的结果

1）[462,87,275,61,170],503,[897,908,653,512]

2）[170,87,275,61],462,503,[897,908,653,512]

3）[61,87],170,[275],462,503,[897,908,653,512]

4) 61,[87,170],[275],462,503,[897,908,653,512]

5) 61,87[,170],[275],462,503,[897,908,653,512]

6) 61,87,170,[275],462,503,[897,908,653,512]

7)61,87,170,275,462,503,[897,908,653,512]

8) 61,87,170,275,462,503,512,[653,]897[908]

9) 61,87,170,275,462,503,512,653,897[,908]

10) 61,87,170,275,462,503,512,653,897,908

4、对于直接插入排序，希尔排序，冒泡排序，快速排序，直接选择排序，堆排序和归并排序等排序方法，分别写出：

（1）平均时间复杂度低于O（n2）的排序方法；

（2）所需辅助空间最多的排序方法；

（3）最好情况和最坏情况下的时间复杂度相同的排序方法。

（1）

（2）

（3）

答案

（1）希尔排序，快速排序，堆排序，归并排序

（2）归并排序 ？这个不确定

（3）直接选择排序，堆排序，归并排序

5、已知序列{70,83,100,65,10,32,7,9},请给出采用插入排序法对该序列做升序时的每趟结果

1）(70,83),100,65,10,32,7,9

2）(70,83),100,65,10,32,7,9

3) 65,70,83,100,10,32,7,9

4) 10,65,70,83,100,32,7,9

5) 10,32,65,70,83,100,7,9

6) 7,10,32,65,70,83,100,9

7) (7,9,10,3265,70,83,100,)