

安徽大学 20_21 —20_22 学年第_2_学期

《 数据结构 》考试试卷（A 卷）

（闭卷 时间 120 分钟）

考场登记表序号_____

题 号	一	二	三	四	五	六	七	总分
得 分								
阅卷人								

得 分

一、算法分析题（每小题 5 分，共 25 分）

1. 分析下面算法的时间复杂度。

```
int Func1( int n )
{
    int sum=0, m, t ;
    for (m=1; m<=n; m++)
    {
        p=1;
        for (t=1; t<=m; t++)
            p*=t ;
        sum+=p ;
    }
    return  sum;
}
```

2. 假如当前带头单链表里的数据元素依次为：1，2，2，3，6，4，3，5，4，请写出运行如下算法后的单链表内的数据元素。

```
void Func2(LinkList L)
{
    LNode *p, *q, *r;
    p=L->next;
    while(p != NULL)
    {
        r = p;
        q = p->next;
        while(q != NULL)
        {
            if(p->data == q->data)
            {
                r->next = q->next;
                free(q);
            }
        }
        p = r->next;
    }
}
```

```

        q = r->next;
    }
    else
    {
        r = q;
        q=q->next;
    }
}
p=p->next;
}
}

```

3. 如下算法程序的功能是实现链队列的出队操作，请补充完整：

```

Status DeQueue(LinkQueue & Q, QElemType & e)
{
    if(Q.front == Q.rear)
        return ERROR;
    p=Q.front->next;
    e=p->data;
    ①;
    ②;
    delete p;
    return OK;
}

```

①: _____

②: _____

4. 如下算法程序的功能是使用 KMP 算法实现串的模式匹配，其中①-⑤中有两处代码有误，请找出来，并更正

```

int Index_KMP(SString S, SString T, int pos)
{
    int i = pos;
    int j = 1;
    while(i <= S.length && j <= T.length)           ①
    {
        if(S.ch[i] == T.ch[j])                       ②
        {
            i++;j++;                                   ③
        }
        else
        {
            i = i - j + 2;                             ④
            j = next[j];                               ⑤
        }
    }
}

```

```
}  
}  
}
```

错误 1: _____

错误 2: _____

5. 已知 print 函数用来打印字符数组，该函数的定义如下：

```
void print(int i)  
{  
    if(i<=n && (A[i] != '0'))  
    {  
        Print(2*i);  
        printf("%c", A[i]);  
        Print(2*i+1)  
    }  
}
```

已知 A 数组定义如上表所示，请写出 Print(1)的输出结果

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	D	E	F	G	0	0	H	0	I	J	K	L

得分

二、简答题（每小题 5 分，共 20 分）

6. 已知广义表 $L=((a,b),(c,d))$ ，请计算 $\text{Tail}(\text{Head}(\text{Tail}(L)))$ 的运算结果。

7. 有三维数组 $a[0..7,0..8,0..9]$ 采用按行序优先存储，数组的起始地址是 1000，每个元素占用 4 个字节，请计算元素 $a[2,5,6]$ 的起始地址。

8. 请画出由下面森林所转换的二叉树。



9. 有三个元素的出栈顺序为 A、B、C，列出它们所有可能的入栈顺序。

得分	
----	--

三、应用题（10 和 11 题每题 10 分，12 题 15 分，共 35 分）

10. 已知某棵二叉树的先序遍历和中序遍历结果分别为 ABGCEHFD 和 ACGHEDFB，请画出这棵二叉树。

学号

姓名

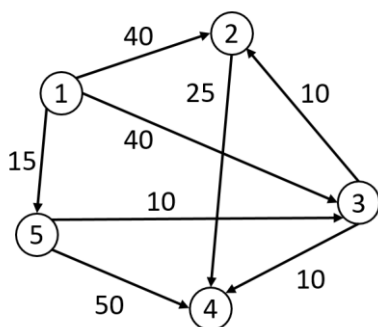
专业

年级

院/系

11. 已知某一组记录的关键字序列 $R=(10, 3, 5, 20, 80, 67, 50, 30)$ ，请给出每趟归并排序后的关键字序列。（注：排序按照关键字由小到大排列）

12. 采用 Dijkstra 算法，计算下图中从源顶点 1 到其它顶点间最短路径。将此过程填入下表中。



终点	D[W] 从 V1 到各终点的长度和最短路径			
V2	40 {V1,V2}			
V3	40 {V1,V3}			
V4	∞			
V5	15 {V1,V5}			
Vj	V5			
S	{V1,V5}			

四、算法设计题（每小题 10 分，共 20 分）

得 分	
-----	--

13. 给定 n 个整数记录序列，存放在 `int` 型的数组 `r[1..n]`, 请采用快速排序的思想设计算法，找到第 k 小的数字。

学号

姓名

专业

年级

院/系

线
订
装
超
勿
题
答

14.给定一个非空二叉排序树，其所有节点的数据均不相同，其采用二叉链表结构存储，根节点的指针为 T，链节点的结构为（lchild, data, rchild），请设计递归算法，在树中查找值为 x 的节点，若查找成功，则从小到大依次输出所有值大于等于 x 的节点的数据，否则返回“notfound”。

该二叉排序树的二叉链表存储表示如下：

```
typedef struct BSTNode{  
    int data;  
    struct BSTNode *lchild, rchild;  
}BSTNode, *BSTree;
```

安徽大学 20 21—20 22 学年第 2 学期

《 数据结构与算法 》 A 卷 参考答案及评分标准

一、算法分析题（每小题 5 分，共 25 分）

1. 答案：

$O(n^2)$

评分规则：没有写 O 或者括号扣两分

2. 答案：

运行后的状态为：1->2->3->6->4->5

评分准则：顺序或者值错一个扣一分

3. 答案：

①：Q.front->next = p->next;

②：if(Q.rear==p) Q.rear=Q.front

评分准则：每空 2.5 分

4. 答案：

②应为 if(j==0 || S.ch[i] == T.ch[j])

④删除

评分规则：每空 2.5 分

5. 答案：

输出结果为：DBHEAIFJCKGL

评分规则：顺序错 1 个扣一分

二、简答题（每小题 5 分，共 20 分）

6. 答案：

(1) Tail(L)=(c,d) (2分)

(2) Head(Tail(L))=(c,d) (2分)

(3) Tail(Head(Tail(L)))=(d) (1分)

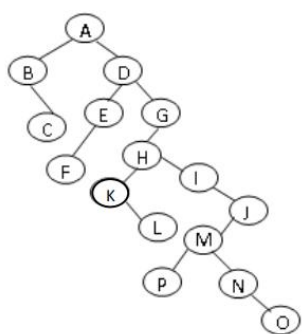
7. 答案：

$\&a[2,5,6] = \&a[0,0,0] + (2*9*10 + 5*10 + 6)*4$

$= 1000 + (180 + 50 + 6)*4 = 1944$

评分规则：地址计算正确 5 分

8. 答案：



评分标准：每一对子树间的关系正确得 1 分，所有关系正确得 5 分

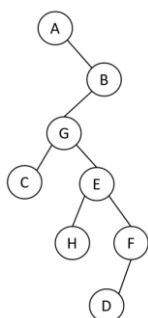
9. 答案：

ABC、CBA、BAC、ACB、CAB

评分标准：每一种顺序正确得 1 分

三、应用题（10 和 11 题每题 10 分，12 题 15 分，共 35 分）

10. 答案：



评分标准：每一对节点间的关系正确得 1 分，所有关系正确得 10 分

11. 答案：

第一趟：R=(3, 10, 5, 20, 67, 80, 30, 50)

第二趟：R=(3, 5, 10, 20, 30, 50, 67, 80)

第三趟：R=(3, 5, 10, 20, 30, 50, 67, 80)

评分标准：每趟排序正确得 3 分，三趟全对的 10 分，每趟结果中错两个序列扣 0.5 分

12. 答案：

终点	D[W] 从 V1 到各终点的长度和最短路径			
V2	40 {V1,V2}	35 {V1,V5,V3V2}	35 {V1,V5,V3V2}	35 {V1,V5,V3V2}
V3	40 {V1,V3}	25 { V1,V5,V3}	25 { V1,V5,V3}	25 { V1,V5,V3}
V4	∞	65 {V1,V5,V4}	60 {V1,V5,V3,V2,V4}	60 {V1,V5,V3,V2,V4}
V5	15 {V1,V5}	15 {V1,V5}	15 {V1,V5}	15 {V1,V5}

Vj	V5	V3	V2	V2
S	{V1,V5}	{V1,V5,V3}	{V1,V5,V3,V2}	{V1,V5,V3,V2,V4}

评分标准：每个空正确的 1 分（注：最后两行，每列算一个空）

四、算法设计题（每小题 10 分，共 20 分）

13. 答案：

```
int select(int r[], int low, int high, int k){
    r[0]=r[low];
    first=low;
    end=high;
    while(low<high){
        while(low<high &&L.r[high]>=pivotkey){
            high--;
        }
        if(low<high){
            r[low]= r[high];low++;
        }
        while(low<high &&L.r[low]<=pivotkey){
            low++;
        }
        if(low<high){
            r[high]=r[low]; high--;
        }
    }
    if(low-first==k-1){
        return r[0];
    }
    else(low-first>k-1){
        return select(r[], first, low-1, k);
    }

    else{
        return select(r[], low+1, end, k-(low-first+1));
    }
}
```

评分标准：根据算法完整程度酌情给分

14. 答案：

算法思想为：

从根节点进行查找，如果其数据域的值等于 x，则中序遍历右子树，输出所有值大于 x 的节点数据；

否则如果其数据域的值小于 x ，且左子树不为空，则递归查找左子树；若左子树为空，则查找失败，返回 **notfound**。

如果其数据域的值大于 x ，如果右子树不为空，则递归查找右子树。若右子树为空，则查找失败，返回 **notfound**。

```
void Print (BSTreeT){
    if(T){
        Print(T->lchild);
    cout<<T->data;
        Print(T->rchild);
    }
}

void Search_Print(BSTree T, int x){
    if(T->data == x){
        cout<<T->data;
    InOrderTraverse(T->rchild);
    }
    else if(T->data >x){
        if(T->lchild){
            Search_Print(T->lchild, int x);
        }else{
            cout<< "not found" ;
        }
    }
    else{
        if(T->rchild){
            Search_Print(T->rchild, int x);
        }else{
            cout<< "not found" ;
        }
    }
}
```

评分标准：根据算法完整程度酌情给分

安徽大学 20_21 —20_22 学年第_2_ 学期

《 数据结构与算法 》考试试卷 (B 卷)

(闭卷 时间 120 分钟)

考场登记表序号_____

题 号	一	二	三	四	五	六	七	总分
得 分								
阅卷人								

得 分

一、算法分析题 (每小题 5 分, 共 25 分)

1. 分析下面算法的时间复杂度。

```
int Function( int n )
{
    int i , j, sum;
    sum = 0;
    for (i=1; i<=n; i++)
    {
        sum+=i
        for (j=1; j <= n-i; j++)
            sum *=j ;
    }
    return  sum;
}
```

2. 下列算法是将线性表 L 中第 i 个数据元素删, 请补充完整。

```
Status ListDeleteLT(LinkList &L,int i,ElemType &e){
    p=L;j=0;
    while(p->next &&j<i-1){
        p=p->next; ++j;
    }
    if(!(p->next)||j>i-1) return ERROR;
    ①;
    ②;
    e=q->data;
    delete q;
    return OK;
}
```

① _____

② _____

3. 阅读并分析下面的函数，回答问题。

void Function(AdjList GL, int i, int n) //Adjlist 表示邻接表类型

```
{
    Queue Q;
    InitQueue(Q);
    cout<<i<<' ';
    visited[i]=true;
    QInsert(Q,i);
    while(!QueueEmpty(Q)) {
        int k=QDelete(Q);
        edgenode* p=GL[k];
        while(p!=NULL)
        {
            int j=p->adjvex;
            if(!visited[j])
            {
                cout<<j<<' ';
                visited[j]=true;
                QInsert(Q,j);
            }
            p=p->next;
        }
    }
}
```

请回答 Function 函数的功能是？

4. 如下为二分查找的非递归算法，部分代码有误，并找出来，并更正。

Int Binsch(ElemType A[],int n, KeyType K)

```
{
    int low=0;
    int high=n-1;
    while (low<high)
    {
        int mid= (low+high)/2;
        if (K==A[mid].key) return mid;
        else if (K<=A[mid].key)
            high=mid-1;
        else low=mid+1;
    }
    return -1;
}
```

错误 1: _____

错误 2: _____

5. 阅读并分析下面的排序算法，回答问题。

```
void Function(int arr[], int n)
{
    int i,j, temp;
    for (i = 0; i <n- 1; i++)
    {
        for (j = 0; j <n- 1 - i; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}
```

- (1) 请分析上述 Function(int arr[], int n)算法的功能。
- (2) 若 arr[] = { 23,18,16,33,29,12}，执行 Function (arr,6)后,其各趟排序结果为多少？

得分	
----	--

二、简答题（每小题 5 分，共 20 分）

6. 将以下算法按照最坏时间复杂度从小到大进行稳定排序。

- ① 顺序表的插入 ②构造哈夫曼树 ③Dijkstra算法 ④二叉排序树的查找
- ⑤平衡二叉树的查找 ⑥快速排序 ⑦归并排序

7. 三个元素的出栈顺序为A、B、C，列出它们所有可能的入栈顺序。

8. 若先序序列为ABCDE，列出一个后序序列，使得该先序序列和后序序列可以唯一地确定一棵二叉树。

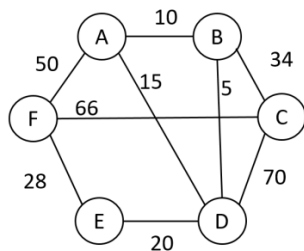
9. 若有七个关键字 1、2、3、4、5、6、7，列出一个输入序列，使得依此序列构建出的二叉排序树具有最小的平均查找长度。

得分	
----	--

三、应用题（10 和 11 题每题 10 分，12 题 15 分，共 35 分）

10. 某系统在通讯时，只出现了 A, B, C, D, E 五种字符，其出现频率分别为：0.1, 0.3, 0.5, 0.06, 0.04，试设计哈夫曼编码。

11. 已知有 A, B, C, D, E, F 六个城市, 每个城市间建立通信线路的成本如下图所示, 请设计通信网络, 在保证任意两个城市均能通讯前提下, 尽可能降低线路建造成本。



12. 已知某一组记录的关键字序列 $R=(38, 3, 56, 20, 93, 67, 50, 30, 49, 76)$, 请给出每趟希尔排序后的关键字序列。(注: 排序按照关键字由小到大排列, 且 dk 取值为 5, 3, 1)

四、算法设计题 (每小题 10 分, 共 20 分)

得分

13. 通过键盘输入无序整数记录序列, 以 9999 作为结束输入标志, 建立一棵二叉链表表示方式存储的二叉排序树, 并打印输出对其由大到小遍历结果输出, 请设计满足要求的算法。

该二叉排序树的二叉链表存储表示如下:

```
typedef struct BSTNode{
    int data;
    struct BSTNode *lchild, *rchild;
}BSTNode, *BSTree;
```


14. 2. 给定 n 个整数记录序列，存放在 `int` 型的数组 `r[1..n]`, 请采用快速排序的思想设计算法，找到第 k 大的数字。

安徽大学 20 21—20 22 学年第 2 学期

《 数据结构与算法 》 B 卷 参考答案及评分标准

一、算法分析题（每小题 5 分，共 25 分）

1. 答案：

$O(n^2)$

评分规则：没有写 O 或者括号扣两分

2. 答案：

① $q=p \rightarrow next;$

② $p \rightarrow next=q \rightarrow next;$

评分准则：每空 2.5 分

3. 答案：

算法功能是：从初始点 v_i 出发广度优先搜索由邻接表 GL 所表示的图。

评分准则：提及广度优先搜索 3 分，全对得 5 分

4. 答案：

① while ($low < high$) 应为 while ($low \leq high$)

② else if ($K \leq [mid].key$) 应为 else if ($K < [mid].key$)

评分准则：每个错误 2.5 分

5. 答案：

① 算法功能是冒泡排序

② 18,16, 23, 29,12,33

16, 18, 23, 12, 29,33

16, 18, 12, 23, 29,33

评分准则：功能正确 2 分，每趟排序正确得 1 分

二、简答题（每小题 5 分，共 20 分）

6. 答案：

⑤①④⑦②③⑥ （注：①n ② n^2 ③ n^2 ④n⑤logn⑥ n^2 ⑦nlogn）

评分准则：错一对顺序扣 1 分

7. 答案：

ABC、CBA、BAC、ACB、CAB

评分准则：每种入栈顺序得 1 分

8. 答案：

CDBEA 或 BDECA 均正确

评分准则：错一对顺序扣 1 分

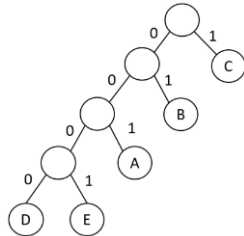
9. 答案:

4213657

评分准则: 只要保证 26 在 4 之后, 13 在 2 之后, 57 在 6 之后均正确

三、应用题 (10 和 11 题每题 10 分, 12 题 15 分, 共 35 分)

10. 答案:



A: 001

B:01

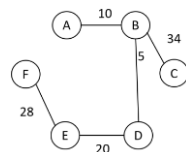
C:1

D:0000

E:0001

评分标准: 每个编码正确得 2 分 (注: 只要保证各字符加权和为 1.8 均为正确)

11. 答案:



通信网络:

评分标准: 每条边正确得 2 分 (注: 只要保证构造成本为 99 均为正确)

12. 答案:

第一趟: $R=(38,3,30,20,76,67,50,56,49,93)$

第二趟: $R=(20,3,30,38,56,49,50,76,67,93)$

第三趟: $R=(3,20,30,38,49,50,56,67,76,93)$

评分标准: 每错两个序列扣 1 分, 每趟排序结果正确得 5 分

四、算法设计题 (每小题 10 分, 共 20 分)

13. 答案:

算法的思想: 动态构建二叉排序树, 并对其进行逆中序遍历的方式即可达到要求。

```
void Insert_BSTree(BSTree&T, int key){
    if(!T){
        s=new BSTree();
        s->data=key;
        s->lchild =NULL;
        s->rchild = NULL;
        T=s;
    }
}
```

```

        else if(T->data>key)
        {
            Insert_BSTree(T->lchild, key)
        }
        else if(T->data<key)
        {
            Insert_BSTree(T->rchild, key)
        }
    }

void Creat_BSTree(BSTree&T)
{
    int key;
    scanf("%d", &key);
    while(key!=9999){
        Insert_BSTree(T,key);
        scanf("%d", &key);
    }
}

void Inorder(BSTree T)
{
    Inorder(T->rchild)      ;
    cout<<T->data;
    Inorder(T->lchild);
}

```

评分标准：根据算法完整程度酌情给分

14. 答案：

```

int select(int r[], int low, int high, int k){
    r[0]=r[low];
    first=low;
    end=high;
    while(low<high){
        while(low<high &&L.r[high]<=pivotkey){
            high--;
        }
        if(low<high){
            r[low]= r[high];low++;
        }
        while(low<high &&L.r[low]>=pivotkey){
            low++;
        }
        if(low<high){
            r[high]=r[low]; high--;
        }
    }
}

```

```
    }  
}  
if(low-first==k-1){  
    return r[0];  
}  
else(low-first>k-1){  
    return select(r[], first, low-1, k);  
}  
  
else{  
    return select(r[], low+1, end, k-(low-first+1));  
}  
}
```

评分标准：根据算法完整程度酌情给分