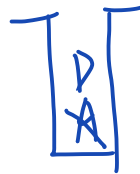
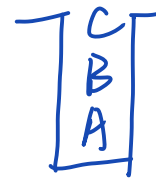


BCAED



D



数据结构2019-2020 (A) 参考答案

一 简答题

1、若元素的进栈序列为A, B, C, D, E, 运用栈操作, 能否得到出栈序列B, C, A, E, D和D, B, A, C, E, 为什么?

答: 评分标准 (1) 答对一个1分, (2) 2分

(1) 能得到出栈序列B, C, A, E, D。过程如下: A, B进, B出; 然后C进, C, A依次出栈; 接下来D, E先后入栈, 出栈得到E, D, 不能得到D, B, A, C, E

(2) 因为栈后进先出

可以进栈出栈不可不进

2、请给出四维数组的列主映射函数。

答: 评分标准 答对得4分。

假设四维数组A[u1,u2,u3,u4],任一元素A[j1,j2,j3,j4]的列主映射函数。

Map(j1,j2,j3,j4)=j4*u1*u2*u3+j3*u1*u2+j2*u1+j1

3、已知一棵二叉树的前序遍历为ABDEGCFH, 中序遍历为DBGEACFH, 求二叉树的后序遍历。

答: 评分标准

(1) 仅画出二叉树得2分

(2) 后序遍历正确2分 (直接得到后续遍历4分)

二叉树 A

B C

D E F

G H

后序遍历为: DGE BH FCA

D G E B H F C A



4、含有n个非叶结点的m阶B-树中至少包含多少个关键字。请给出分析过程。

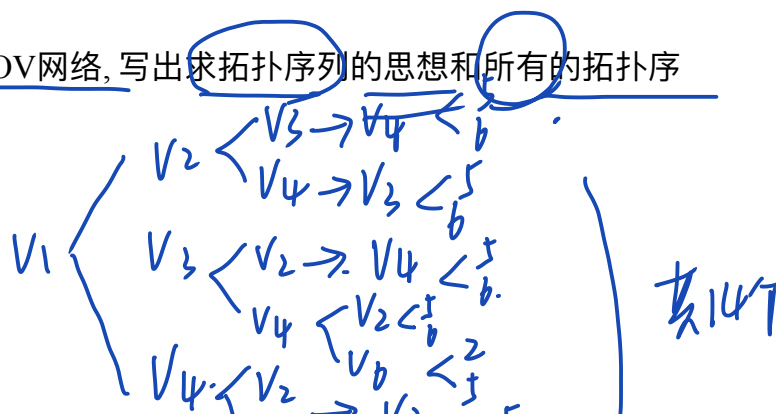
答: 评分标准

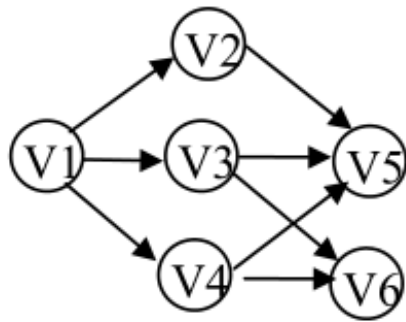
(1) B-树中的根结点至少有两个孩子也即一个关键字 得1分

(2) 其他的n-1个内部节点至少有 $\lceil m/2 \rceil - 1$ 个关键字 得2分。

(3) 最后结果 $(n-1) * (\lceil m/2 \rceil - 1) + 1$ 得 1分

5、下图是有向图表示的AOV网络, 写出求拓扑序列的思想和所有的拓扑序列。





答：评分标准：

(1) 写出求拓扑序列的思想（见书上428页）得2分。

(2) 写对14个拓扑序列2分，写对部分拓扑序列得1分。

V1 (V2,V3,V4全排列6种) (V5,V6全排列2种) 可得12种拓扑序列

V1 (V3, V4全排列2种) V6,V2,V5 可得2种拓扑序列
一共14种拓扑序列

二、应用题

1、设散列长度为13，散列函数 $\text{Hash}(k)=k\%11$ ，若输出序列为 {22,41,53,46,30,13,1,67}，解决溢出的方法为线性开型寻址散列。

(1) 请构造该散列表

(2) 搜索元素30和元素67所需要比较的次数是多少？

(3) 给出删除元素1后的散列表结构

答：评分标准 (1) 5分，部分答对酌情给分 (2) 3分，答对一个1.5分

(3) 2分，部分答对酌情给分

(1) 构造的散列表：

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Content	22	1	46	13	67				41	53	30		

(2) 搜索30:3次 搜索67:4次

(3) 删除元素1后的散列表：

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Content	22	67	46	13					41	53	30		

2、一个 $n \times n$ 的矩阵 M 是一个反对角矩阵当且仅当对于所有满足 $i+j \neq n+1$ 的 i 和 j 有 $M(i,j)=0$

(1) 给出一个 4×4 反对角矩阵的样例

(2) 设计一种映射模式，用来把一个反对角矩阵映射到一个大小为 n 的一维数组之中。

答：评分标准 (1) 5分 (2) 5分

(1)

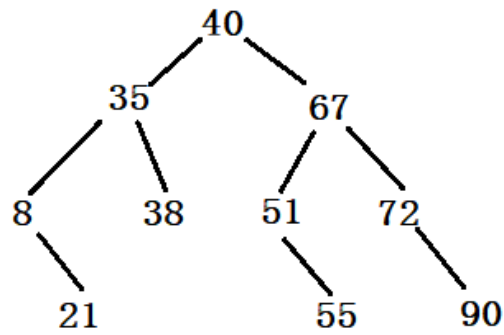
其中，非零数字1可以替换为任意整数数字

(2) 当 $i+j=n+1$ 时， $\text{map}(i,j)=n-j$ 或 $\text{map}(i,j)=n-i$

当 $i+j \neq n+1$ 时，值为0

3、按照序列 (40,72,38,35,67,51,90,8,55,21) 建立一棵AVL搜索树，画出该树。并求出在等概率的情况下，查找成功的平均查找长度。

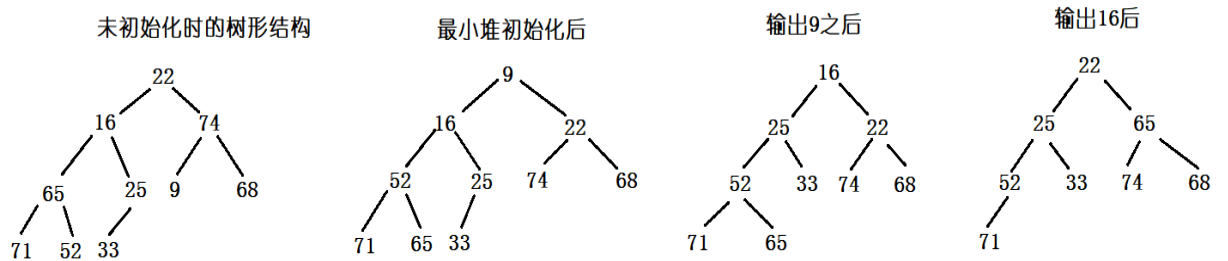
答：评分标准 画出AVL树8分，部分画对酌情给分；平均长度2分
AVL树如下：



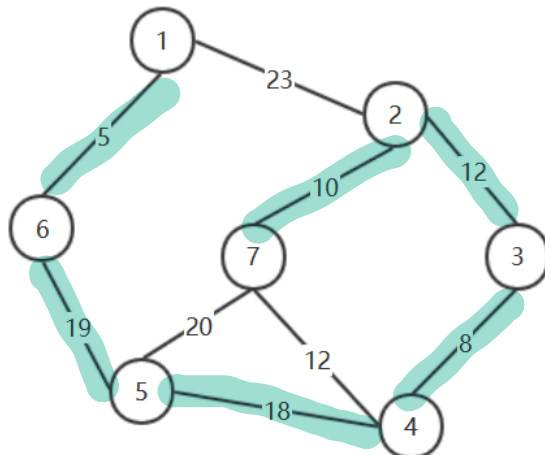
平均查找长度： $(1+2 \times 2 + 4 \times 3 + 3 \times 4) / 10 = 2.9$

4、对关键字序列 (22,16,74,65,25,9,68,71,52,33) 进行堆排序，输出两个最小关键字后的剩余堆是什么？

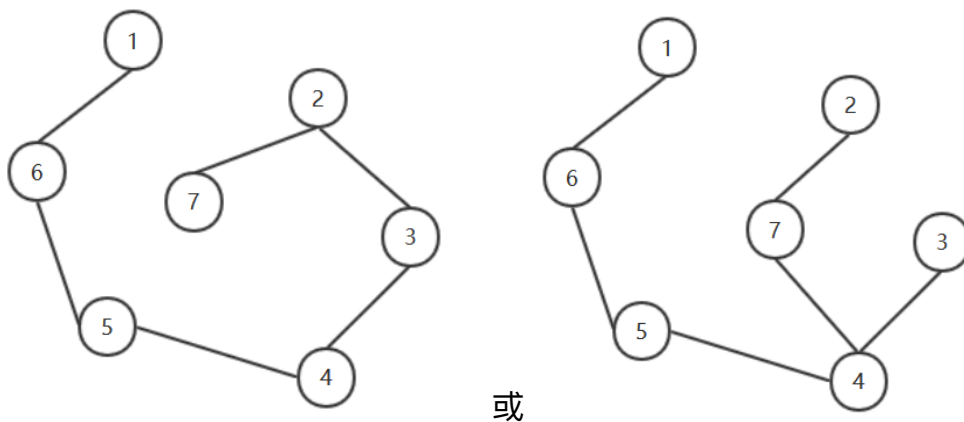
答：评分标准10分 部分正确酌情给分



5、试用prim(以顶点4为起点)算法计算出如下连通网的最小生成树。



答：评分标准10分 以下两种情况任意一种都满分，每种情况若部分正确酌情给分



三 算法题

1、设计将整数数组所有奇数移到所有偶数之前的算法，要求算法时间复杂度最优。叙述算法思想并用C++实现，分析其时间复杂度。

评分标准：

算法思想：可以使用类似快排的方法进行比较，定义左右两个指针，左边向右找偶数，右边向左找奇数，如果遇到则进行交换，直到左指针大于等于右指针结束交换。得8分

```
void quickpass(int r[], int s, int t)
{
    int i=s,j=t,x=r[s];
    while(i<j)< p="">
    {
        while (i<j && r[j]%2==0) j=j-1; if (i<j) {r[i]=r[j];i=i+1;}
        while (i<j && r[i]%2==1) i=i+1; if (i<j) {r[j]=r[i];j=j-1;}
    }
    r[i]=x;
}
```

得4分

上述算法的时间复杂度为 $O(n)$ ，得3分

若使用冒泡或其他算法，描述正确的，酌情给分

2、编写算法，将二叉树的叶结点按从左到右的顺序连成一个单链表，表头指针为head，链接时用叶结点的右指针域来存放单链表指针。叙述算法思想并给出算法实现，分析算法复杂性。

已知二叉树采用链式存储结构，节点结构如下：

lchild	data	rchild
--------	------	--------

其中data表示节点存储的数据，lchild和rchild分别表示指向左子节点的指针和指向右子节点的指针。

答：评分标准：

通常我们所使用的先序、中序和后序遍历对于叶结点的访问都是从左到右的顺序，这里我们选择中序递归遍历。算法思想：设置前驱结点指针pre，初始为空。第一个叶结点由指针head指向。遍历到叶结点时，就将它前驱的rchild指针指向它，最后一个叶结点的rchild为空。得7分。

算法实现如下：

```
LinkedList head, pre=NULL; //全局变量
LinkedList InOrder(BiTree bt){
    if(bt){
```

```

InOrder(bt->lchild);           //中序遍历左子树
if(bt->lchild==NULL&&bt->rchild==NULL) //叶结点
    if(pre==NULL){
        head=bt;
        pre=bt;
    }                           //处理第一个叶结点
    else{
        pre->rchild=bt;
        pre=bt;
    }                           //将叶结点链入链表
InOrder(bt->rchild);           //中序遍历右子树
pre->rchild=NULL;               //设置链表尾
}
return head;
}

```

得5分。

上述算法的时间复杂度为 $O(n)$ ，辅助变量使用head和pre，栈空间复杂度为 $O(n)$ 。
得3分。

1、设散列长度为13，散列函数 $\text{Hash}(k)=k\%11$ ，若输出序列为 {22,41,53,46,30,13,1,67}，解决溢出的方法为线性开型寻址散列。

- (1) 请构造该散列表
- (2) 搜索元素30和元素67所需要比较的次数是多少？
- (3) 给出删除元素1后的散列表结构

0	1	2	3	4	5	6	7	8	9	10	11	12
22	1	46	13	67				41	53	30		
1	1	1	2	4				1	1	3		

(2) 30: 3次 67: 4次

0	1	2	3	4	5	6	7	8	9	10	11	12
22	67	46	13					41	53	30		
1	1	1	2					1	1	3		

答：评分标准 (1) 5分，部分答对酌情给分 (2) 3分，答对一个1.5分

(3) 2分，部分答对酌情给分

(1) 构造的散列表：

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Content	22	1	46	13	67				41	53	30		

(2) 搜索30:3次 搜索67:4次

(3) 删除元素1后的散列表：

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Content	22	67	46	13					41	53	30		

2、一个 $n \times n$ 的矩阵 M 是一个反对角矩阵当且仅当对于所有满足 $i+j \neq n+1$ 的 i 和 j 有 $M(i,j)=0$

- (1) 给出一个 4×4 反对角矩阵的样例
- (2) 设计一种映射模式，用来把一个反对角矩阵映射到一个大小为 n 的一维数组之中。

0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0

$$A[i, n+1-i] = a[n-i]$$

答：评分标准 (1) 5分 (2) 5分

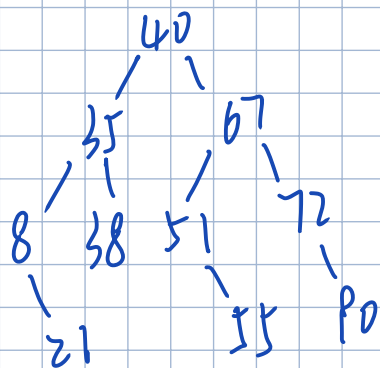
(1)

其中，非零数字1可以替换为任意整数数字

(2) 当 $i+j=n+1$ 时， $\text{map}(i,j)=n-j$ 或 $\text{map}(i,j)=n-i$

当 $i+j \neq n+1$ 时，值为0

3、按照序列 (40,72,38,35,67,51,90,8,55,21) 建立一棵AVL搜索树，画出该树。并求出在等概率的情况下，查找成功的平均查找长度。

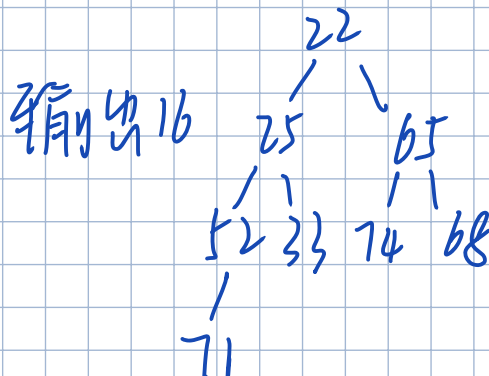
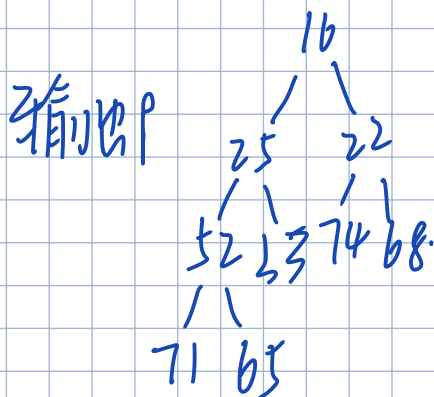
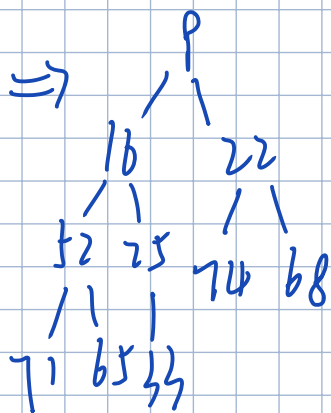
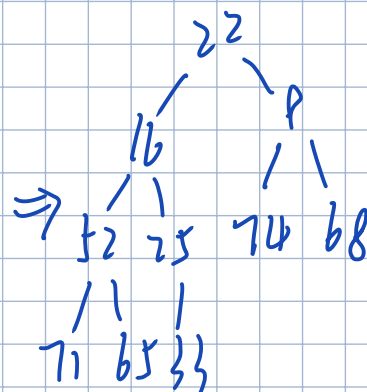
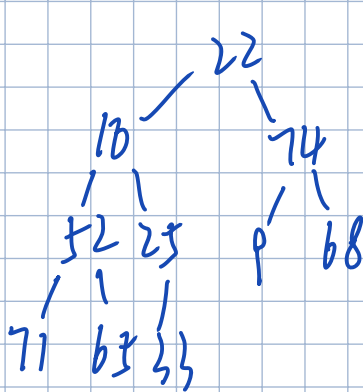
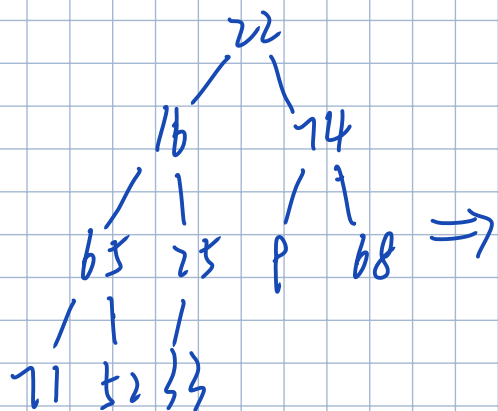


$$1 + 2 \times 2 + 3 \times 4 + 4 \times 3$$

$$= 1 + 4 + 12 + 12 = 29$$

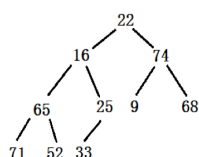
$$29 / 10 = 2.9$$

4、对关键字序列 (22,16,74,65,25,9,68,71,52,33) 进行堆排序，输出两个最小关键字后的剩余堆是什么？

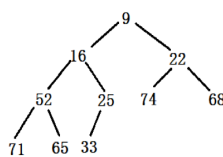


答：评分标准10分 部分正确酌情给分

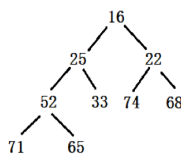
未初始化时的树形结构



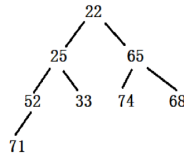
最小堆初始化后



输出9之后



输出16后



1、设计将整数数组所有奇数移到所有偶数之前的算法，要求算法时间复杂度最优。叙述算法思想并用C++实现，分析其时间复杂度。

评分标准：

利用快排思想

设置左右游标分别指向第一个和最后一个元素, 左游标从前往后, 检索到偶数停止, 右游标从后往前, 检索到奇数停止
每一轮交换左右游标数值, 直至左右游标重合。

```
void change(int *A, int n){  
    int left=0, right=n-1;  
    while (left < right){  
        while (A[left]%2==1) left++;  
        while (A[right]%2==0) right--;  
        int temp = A[left];  
        A[left] = A[right];  
        A[right] = temp;  
    }  
}
```

复杂度 $O(n)$

2、编写算法, 将二叉树的叶结点按从左到右的顺序连成一个单链表, 表头指针为head, 链接时用叶结点的右指针域来存放单链表指针。叙述算法思想并给出算法实现, 分析算法复杂性。

已知二叉树采用链式存储结构, 节点结构如下:

lchild	data	rchild
--------	------	--------

其中data表示节点存储的数据, lchild和rchild分别表示指向左子节点的指针和指向右子节点的指针。

```
void ml(binaryNode * root){  
    binarynode a[1000]; int count=0;  
    method(root);  
    for (int i=0; i<count; i++){
```



```
    a[i] → rchild = a[i+1];  
}
```

```
}
```

```
void method( binaryNodex root ){  
    if( root != null ){  
        if( root → lchild == null && root → rchild == null ){  
            a[count++] = root;  
        }  
        method( root → lchild );  
        method( root → rchild );  
    }  
}
```