# 《数据结构》试题答案（开卷） **A**卷

（电信系本科**2005**级 **2006**年**11**月）

姓名班级

题 号

一

二

三

总分

题 分

40

30

30

100

得 分

得

分

# 一、回答下列问题 (每题5分，共40分)

**1**．给定序列（**67**，**45**，**87**，**19**，**55**，**32**，**70**，**60**，**90**， **23**），写出它的初始堆序列。

答：调整后的初始堆序列(小根堆)为：19，23，32，45，55，87，70，

60，90，67

或者是大根堆：90, 67, 87, 60, 55, 32, 70, 45, 19, 23

19

23

32

45

55

87

70

60

90

67

**2**．设一个序列奇数项和偶数项分别由小到大有序，用什么方法可以最快得到一个有序序列，分析它的时间复杂度。

答：把奇数项和偶数项分为2个有序序列，然后进行合并，时间复杂度为O(n)。实际上就是把2个有序表合并为一个有序表。见教科书算法

2.7。

**3**．二叉排序树中的最大值在二叉排序树的何处？答：最大值应该位于二叉排序树中根的右子树的最右叶子上。

**4**．在**2048**个互不相同的关键码中选择最小的**5**个关键码，用堆排序是否比用锦标赛排序更快？为什么？

答：此题用锦标赛排序比堆排序要快。理由是；

①在首次求最小值时，锦标赛排序对2048个结点建树得到最小码只需比较n-1(即2047)次，而此时堆排序建初始堆得到最小码却可能需要比较

4072次 (因为每个结点的调整都要与左右两边的孩子相比。从第1024个结点往前调整，有512个结点可能调整1次，但要与左右孩子都比较，有

256个结点可能调整2次，每次都要与左右孩子比较，有128个结点可能调整3次，……有32个结点调整5次，……根结点可能要调整10次，每次都会与左右孩子比较，所以可能会比较2036×2=4072次)。

而两种算法对求后面4个次小码的平均效率相同，都是log2n，所以，此

题用锦标赛排序会比堆排序快。

**5**． **n**个顶点、**m**条边的全连通图，至少去掉几条边才能构成一棵树？

答：因为树的结构是一对多，即n个结点的树只有n-1条边与双亲结点相连。只要再多添一条边就会成为图结构。所以，m条边的图要去掉m-(n-

1)=m-n+1条边才能构成一棵树。这棵树也就是最小生成树。

**6**．设模式串为：**liuwenliuyuliuyingliyu,** 求该模式串的**next**函数。

答： **Next[j]=0 1 1 1 1 1 1 2 3 4 1 1 2 3 4 1**

**7**．一个二叉树按层次遍历的顺序存储结构如下，请画出该二叉树 **(φ**为空**)** 。

**1 2 3 4 5 6 7 8 9 10 11 12 13 14 15**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **D** | φ | **C** | φ | **E** | φ | φ | **F** | **G** | φ | φ | **H** | φ |

答：画出二叉树如下：

**8**．设数组**A[ 1..10, 1..8 ]** 的基地址为**2000**，每个元素占**2**个存储单元，若以列序为主序存储（按列存储），则元素**A[ 4**，**5**

**]** 的存储地址是多少？

答：**A[ 4**，**5 ]** 的存储地址是**2086**

# 二、综合题（每题10分，共30分）

|  |
| --- |
| 得分 |
|  |

**1**．输入一序列（**58**，**18**，**29**，**22**，**38**，**81**，**19**，**14**），现分别采用顺序查找和二叉排序树查找，求等概率条件下二者的平均查找长度**ASL**；若改用哈希查找（哈希函数为：**Hash(key)=key mod 11**，哈希表的大小为**11**，采用线性探测法进行冲突处理），求等概率条件下的平均查找长度**ASL**，并对这三种查找方法进行比较。

答：采用二叉排序树查找时，二叉排序树为：采用二叉排序树查找时，

ASL=1/8[1×1+2×2+3×1+4×2+5×1+6×1]=27/8=**3.375**

采用顺序查找时，ASL=1/2（8+1）=**4.5**

采用哈希查找时，查找表为：

0 1 2 3 4 5 6 7 8 9 10

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 22 |  |  | 58 | 81 | 38 | 14（3） | 18 | 29（1） | 19（1） |  |

ASL=1/8（5×1+2×2+1×4）=13/8=**1.625**

**2**．已知用线性有序链表存储整数集合的元素。阅读下面算法，并回答下列问题：

1. 写出执行**ABC**（**a, b**）的返回值，其中**a**和**b**分别为指向存储集合

**{ 2**，**4**，**5**，**7**，**9**，**12 }** 和 **{ 2**，**4**，**5**，**7**，**9 }** 的链表的头指针；

1. 简述算法**ABC**的功能；（**3**）写出算法**ABC**的时间复杂度。

**int ABC (LinkList ha,LinkList hb)**

**{**  // LinkList是带有头结点的单链表

// ha和hb分别为指向存储两个有序整数集合的链表的头指针

**LinkList pa, pb; pa=ha->next; pb=hb->next; while(pa && pb && pa->data==pb->data)**

**{ pa=pa->next; pb=pb->next;**

**}**

**if(pa==NULL && pb==NULL) return 1;**

**else return 0;**

**}**

答：

(1） ABC（a,b）=0;

1. 此函数是判两链表是否相等的函数。若二表完全相同将返回函数值 1，不相同则返回的函数值为0.
2. 算法ABC的时间复杂度为O(min(LinkList a.length, LinkList\_b.length))

**3.** 对于下面的一串字符，根据各字符出现的频度求各个字母的哈夫曼编码，并且建树要遵循二叉树左边结点的权值**≤**二叉树右边结点的权值，请写出详细的求解过程。

**ABCCCEBAAADCCCAEECCCEDE**

答：此题分三步求解，第①步2分，②和③ 各4分。 1 先统计各字母出现的频度，得到下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字母 | A | B | C | D | E |
| 频度 | 5 | 2 | 9 | 2 | 5 |

23

9 14

4 5 (A) **5(E)** 9(C)

2(B) 2(D)

2 哈夫曼树构造如下：

③ 各个字母的哈夫曼编码如下：

字母

A

B

C

D

E

哈夫曼编

码

01

000

11

001

10

得

分

三、 算法设计题（每题10分，共30分）

**1**．采用二叉链表作为存储结构，试编写一个算法求二叉树中结点**p**的双亲结点和孩子结点。

答：采用先、中、后序遍历都可完成，基本思想是：在遍历过程

中，寻找其左孩子或者右孩子是p的节点，当访问到p节点以后，再把其左右孩子写出。只需将遍历算法的visit（q）一句更换为： if (q->rchild==p || q->lchild==p)

{ printf（’the parent of p is：’q->data）; printf（’the lchild of p is：’ , p->lchild->data）; printf（’the rchild of p is：’ , p->rchild->data）;

exit(或return)

}

**2**．若借助栈由输入序列为**1**，**2**，**…**，**n**得到的输出序列为**P1**

**P2 … Pn ,** 设计一个算法输出所有可能的序列，并分析该算法的时间复杂度和空间复杂度。答：算法如下：

OutputStack(String t, int i, int j)

//输出序列t中的第i个到第j个的元素

{

if (j > i) print("\n"); // 换行打印 else {

e = GetElem(t,i); // 取t中的第i个元素 push(s, e);

OutputStack(t, i+1, j);

if (s栈不空)

{ pop(s, e); print(e);

OutputStack(t, i+1, j);

}

}

}

时间复杂度：O(n) =O( n！）＝ O(2n)。

空间复杂度：由于需要栈保存递归参数，因此空间复杂度是递归操作中栈的深度，故为O(n)。

**3**．在游戏软件和图形软件设计中，经常遇到下面的情况：**n** 条直线把屏幕分成**m**个区域，见图。假设每条直线的方程和交点的坐标已知，设计一个数据结构表示这些区域，并设计一个算法，判断鼠标（**x**，**y**坐标）落在哪个区域，并分析此算法的时间复杂度。

4

2

6

7

1

3

5

答：区域树是一棵二叉树，随着直线的逐条输入而生成。

树中每个结点有四个域

|  |  |  |  |
| --- | --- | --- | --- |
| lchild | nodenum | linenum | rchild |

nodenum为结点编号，剖分过程形成的对应的区域编号。

linenum是直线的编号，当结点对应的是不被分割的基本区域时记为－

1，当在分割过程记下？分割它的直线的编号。

lchild，rchild对应每个分割直线的左右两侧的生成区域。

-1

-3

-2

-4

-5

-6

3

4

5

6

0

－2直线

－1

1

2

-3

3

-2

-4

-5

-6

11

4

10

7

0

-1

9

12

8

1

2

0

0

-2

1

-2

2

-4

3

-1

11

-1

4

-6

5

-1 12 -6

6

-1

9

-1 10 -1

7

-1 8 ^

^

^

^

^

^

^

^

^

^

^

^

^

^

^

^

^

^

^

^

^

^

^

^ 建立3个表：

交点表：0，1，2…..

区域表：0，1，2，3…10，11，12及区域所围成的边直线表：直线方程式

显然，此区域树的叶子节点表示划分的区域。

判断鼠标落在哪个区域的算法

输入mouse的坐标（x，y），从区域树的根节点开始，将x，y带入此节点的直线方程，大于0，走左子树，否则，走右子树，直至叶子节点，则此叶子节点表示的区域即为鼠标落在的区域。

算法复杂度分析：

算法的时间复杂度为区域二叉树的平均深度，因为划分的区域为 n！个，也即叶子有n！个，则二叉树的平均深度为o(n)。