

上海交通大学试卷 (_A_ 卷)

(2010 至 2011 学年 第 一 学期)

班级号 _____ 学号 _____ 姓名 _____

课程名称 _____ 《数据结构 (A 类)》 _____ 成绩 _____

1. 填空题 (每格 2 分, 共 20 分) :

1. 求二叉树的后序遍历的第一个结点 firstPost ():

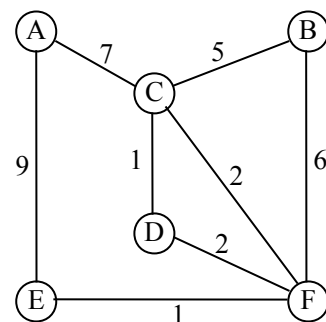
```
bnode* firstPost ( bnode* root ) {  
    bnode *ptr = root;  
    if ( root == NULL ) return NULL;  
    while ( _____ )  
        if ( _____ ) ptr = ptr -> llink;  
        else ptr = ptr -> rlink;  
    return ptr;  
}
```

2. 下列算法实现在顺序散列表中查找值为 k 的关键字, 请在下划线处填上正确的语句。

```
struct record { int key; int others; };  
int hashsqsearch ( struct record hashtable[ ], int k )  
{  
    int i, j; j = i = k % p;  
    while ( hashtable[j].key != k && hashtable[j].flag != 0 )  
    {  
        j = ( _____ ) % m;  
        if ( i == j ) return (-1);  
    }  
    if ( _____ ) return (j);  
    else return (-1);  
}
```

3. 深度为 k 的完全二叉树中最少有 _____ 个结点。
(假设树根深度为 0)

4. 右图中, 利用 Prim 算法求从结点 A 出发的最小生成树的过程中,
逐 次 添 加 的 结 点 为 :
。



2. 选择题（每题2分，共20分）：

- 总 7 页 第 2 页

5. 对某个无向图的邻接矩阵来说, 下列说法错误的是 _____。
- A) 第 i 行上非零元素个数和第 i 列上非零元素个数一定相等
 B) 任一行上的元素不可能全都是零
 C) 矩阵中非零元素个数等于图中边数的 2 倍
 D) 矩阵的列数等于图中的结点数
6. 磁盘文件有 m 个初始归并段, 采用 k 路归并时, 所需要的归并遍数是 _____。
- A) $\log_2 k$ B) $\log_2 m$ C) $\lceil \log_k m \rceil$ D) $\log_k m$
7. 下列说法错误的是 _____。
- A) 求最短路径的 Dijkstra 算法中边的权值不可以为负
 B) Dijkstra 算法允许图中有回路
 C) Prim 算法允许图中有回路
 D) Floyd 算法中边的权值不可以为负
8. 简单插入排序、冒泡排序、快速排序、选择排序、归并排序、堆排序在平均情况下的时间复杂度分别是 _____。
- A) $O(n^2)$, $O(n^2)$, $O(n^2)$, $O(n^2)$, $O(n \log_2 n)$, $O(n \log_2 n)$
 B) $O(n^2)$, $O(n^2)$, $O(n \log_2 n)$, $O(n^2)$, $O(n \log_2 n)$, $O(n \log_2 n)$
 C) $O(n^2)$, $O(n^2)$, $O(n^2)$, $O(n \log_2 n)$, $O(n \log_2 n)$, $O(n \log_2 n)$
 D) $O(n^2)$, $O(n^2)$, $O(n^2)$, $O(n \log_2 n)$, $O(n \log_2 n)$, $O(\log_2 n)$
9. 以下序列中, _____ 是最大化堆。
- A) 75, 65, 30, 15, 25, 45, 20, 10 B) 75, 65, 45, 10, 30, 25, 20, 15
 C) 75, 45, 65, 30, 15, 25, 20, 10 D) 75, 45, 65, 10, 25, 30, 20, 15
10. 设有序顺序表中有 n 个数据元素, 则利用二分查找法查找数据元素 X 的最多比较次数不超过 _____。
- A) $\log_2 n + 1$ B) $\log_2 n - 1$
 C) $\log_2 n$ D) $\log_2(n + 1)$

3. 简答题（每题 6 分，共 12 分）：

1. 已知字母使用频率（见下表），求这 8 个字符的 Huffman 编码值。要求写出 Huffman 树的建立过程，约定在子树合并时，根的权值小的子树为左子树，大的为右子树。

字母	C1	C2	C3	C4	C5	C6	C7	C8
使用频率（%）	5	25	3	6	9	12	36	4

2. 简述如何利用邻接表和邻接矩阵判断有向图是否存在环。

4. 程序分析题（8分）：

```
void swap ( int ary[ ], int i, int j ) {
    int temp = ary[i];
    ary[i] = ary[j];
    ary[j] = temp;
}

void push_down ( int ary[ ], int first, int last ) {
    int r = first, temp;
    while ( 2*r <= last ) {
        if ( 2*r == last ) {
            if ( ary[r] > ary[2*r] ) swap ( ary, r, 2*r );
            break;
        } else
            if ( ary[r] > ary[2*r] && ary[2*r] <= ary[2*r+1] ) {
                swap ( ary, r, 2*r );
                r = 2*r;
            } else if ( ary[r] > ary[2*r+1] && ary[2*r+1] <= ary[2*r] ) {
                swap ( ary, r, 2*r+1 );
                r = 2*r+1;
            } else
                break;
    }
}

void bubble_up ( int array[ ], int i )
{ int j, temp;
    while ( i >= 2 ) {
        j = i/2;
        if ( array[j] <= array[i] ) break;
        else {
            swap ( array, i, j );
            i = j;
        }
    }
}

int heapDelete ( int Array[ ], int i, int n ) {
    int elem = Array[i];
    Array[i] = Array[n];
    if ( i > 1 && Array[i] < Array[i/2] )
        bubble_up ( Array, i );
    else
        if ( Array[i] > Array[2*i] || Array[i] > Array[2*i+1] )
            push_down ( Array, i, n-1 );
    return elem;
}
```

试分析上述程序的功能，包括时间复杂度。

5. 算法设计题（每题 8 分，共 16 分）：

1. 求元素按递增排列的单链表 A 和 B 的交集，并利用原结点空间存放。

2. 给出一系列表示父子关系的结点对，如何判断这些结点是否组成一棵树。

6. 程序题（每题 12 分，共 24 分）：

1. 试编写一个完整程序，用于删除非递减排列的单链表 L 中所有值相同的元素（只保留一个值相同的元素）。
2. 已知一棵完全二叉树以顺序存储的方式保存在一个数组 `Array[1..n]` 中，写出在其上按中序遍历的方式打印结点的程序代码：`void print_in_order (char Array[], int n)`。

7. 附加题（8 分）

设计算法求有向图任意两结点之间是否存在道路。设图的邻接矩阵为 `A`，结点标号为 `0, 1, ... N`。