清华大学 2001 年数据结构考研试题

一、试给出下列有关并查集(mfsets)的操作序列的运算结果：

union(1,2) ， union(3,4) ， union(3,5) ， union(1,7) ， union(3,6) ， union(8,9) ， union(1, 8) ， union(3, 10) ， union(3, 11) ， union(3, 12) ， union(3, 13) ， union(14, 15) ， union(16, 0) ， union(14, 16) ， union(1, 3) ， union(1, 14)。(union 是合并运算，在以前的书中命名为 merge)

要求

(1) 对于 union(i, j)，以 i 作为 j 的双亲； (5 分)

(2) 按 i 和 j 为根的树的高度实现 union(i, j)，高度大者为高度小者的双亲； (5 分)

(3) 按 i 和 j 为根的树的结点个数实现 union(i, j)，结点个数大者为结点个数小者的双亲； (5 分)

二、设在 4 地(A，B，C，D)之间架设有 6 座桥，如图所示：

要求从某一地出发，经过每座桥恰巧一次，最后仍回到原地

(1) 试就以上图形说明:此问题有解的条件是什么？(5 分)

(2) 设图中的顶点数为 n，试用 C 或 Pascal 描述与求解此问题有关的数据结构并编写一个算法，找出满足要求的一条回路. (10 分)

三、针对以下情况确定非递归的归并排序的运行时间(数据比较次数与移动次数)：

(1) 输入的 n 个数据全部有序；(5 分)

(2) 输入的 n 个数据全部逆向有序；(5 分)

(3) 随机地输入 n 个数据. (5 分)

四、简单回答有关 AVL 树的问题.

(1) 在有 N 个结点的 AVL 树中，为结点增加一个存放结点高度的数据成员，那么每一个结点需要增加多少个字位(bit)？(5 分)

(2) 若每一个结点中的高度计数器有 8bit，那么这样的 AVL 树可以有多少层?最少有多少个关键码？(5 分)

五、设一个散列表包含 hashSize=13 个表项，.其下标从 0 到 12，采用线性探查法解决冲突. 请按以下要求，将下列关键码散列到表中.

10 100 32 45 58 126 3 29 200 400 0

(1) 散列函数采用除留余数法，用%hashSize(取余运算)将各关键码映像到表中. 请指出每一个产生冲突的关键码可能产生多少次冲突. (7 分)

(2) 散列函数采用先将关键码各位数字折叠相加， 再用%hashSize 将相加的结果映像到表中的办法. 请指出每一个产生冲突的关键码可能产生多少次冲突. (8 分)

六、设一棵二叉树的结点定义为

```
struct BinTreeNode{
ElemType data;
BinTreeNode *leftChild,  *rightChild;
}
```

现采用输入广义表表示建立二叉树. 具体规定如下:

(1) 树的根结点作为由子树构成的表的表名放在表的最前面;

(2) 每个结点的左子树和右子树用逗号隔开. 若仅有右子树没有左子树， 逗号不能省略.

(3) 在整个广义表表示输入的结尾加上一个特殊的符号(例如”#”)表示输入结果.

例如，对于如右图所示的二叉树， 其广义表表示为 A(B(D，E(G，))，C(，F))

A
/ \
B C
/ \ \
D E F
/
G

此算法的基本思路是:依次从保存广义表的字符串 ls 中输入每个字符. 若遇到的是字母(假定以字母作为结点的值)， 则表示是结点的值， 应为它建立一个新的结点， 并把该结点作为左子女(当 k=1)或有子女(当 k=2)链接到其双亲结点上. 若遇到的是左括号”(“，则表明子表的开始，将 k 置为 1;若遇到的是右括号”)”， 则表明子表结果. 若遇到的是逗号”，”， 则表示以左子女为根的子树处理完毕，应接着处理以右子女为根的子树， 将 k 置为 2.

在算法中使用了一个栈 s， 在进入子表之前，将根结点指针进栈， 以便括号内的子女链接之用. 在子表处理结束时退栈. 相关的栈操作如下:

MakeEmpty(s) 置空栈
Push(s，p) 元素 p 进栈
Pop(s) 进栈
Top(s) 存取栈顶元素的函数

下面给出了建立二叉树的算法， 其中有 5 个语句缺失. 请阅读此算法并把缺失的语句补上. (每空 3 分)

```
Void CreateBinTree(BinTreeNode *&BT,  char ls){
Stacks; MakeEmpty(s);
BT=NULL; //置二叉树
BinTreeNode *p;
int k;
istream ins(ls); //把串 ls 定义为输入字符串流对象 ins
Char ch;
ins>>ch; //从 ins 顺序读入一个字符
While(ch!=”#”){ //逐个字符处理，直到遇到''#''为止
Switch(ch){
case’(‘: _______(1)_______
k=1;
break;
case’)’ : pop(s);
break;
case’，‘: _______(2)_______
break;
```

```
default: p=new BinTreeNode;
_______(3)_______
p->leftChild=NULL;
p->rightChild=NULL;
if(BT==NULL)
_______(4)_______
else if (k==1) top(s)->leftChild=p;
else top(s)->rightChild=p;
}
_______(5)_______
}
}
```

七、下面是一个用 C 编写的快速排序算法. 为了避免最坏情况，取基准记录 pivot 采用从 left，right 和 mid=[(left+right)/2]中取中间值， 并交换到 right 位置的办法. 数组 a 存放待排序的一组记录， 数据类型为 Type， left 和 right 是呆排序子区间的最左端点和最右端点.

```
Void quicksort(Type a&#;, int left, int right){
Type temp;
If(leftType pivot=median3(a, left, right);
Int I=left,  j=right-1;
For( ; ; ){
While(iWhile(iif(itemp=a[i]; a[j]=a[i]; a[i]=temp;
I++; j--;
}
else break;
}
if(a[i]>pivot)
{temp=a[i]: a[i]=a[right]; a[right]=temp;}
quicksort(a, left, i-1); //递归排序左子区间
quicksort(a, i+1, right); //递归排序右子区间
}
}
```

(1) 用 C 或 Pascal 实现三者取中子程序 median3(a, left, right); (5 分)

(2) 改写 quicksort 算法， 不用栈消去第二个递归调用 quicksort(a, i+1, right); (5 分)

(3) 继续改写 quicksort 算法， 用栈消去剩下的递归调用. (5 分)