《数据结构》参考答案

一、选择题**（每小题2分，共20分）**

1-5 AACDD

6-10 CBCAA

二、填空题**（每小题2分，共20分）**

1、50

2、m-1

3、 8， 8

4、快速，堆

5、19/7

6、CBDA

7、6

8、3

9、2

10、{71,66,57,13,25,34}

三、名词解释题**（每小题5分，共15分）**

1.数据：数据是描述客观事物的符号,是能够被计算机输入 ,识别, 处理的各种符号, 是计算机化的信息。

数据项：数据不可分割的最小单位, 一个元素由若干个数据项构成。

数据元素：它是组成数据的基本单位,是数据集合中的个体,在计算机程序中, 通常作为一个整体进行考虑和处理。

数据对象：是性质相同的数据元素的集合,是数据的一个子集。

2. 数据类型：数据类型是一个值的集合和定义在这个值集上的一组操作的总称。

抽象数据类型：是指一个数学模型以及定义在该模型上的一组操作。抽象数据类型的定义取决于它的一组逻辑特性 , 而与其在计算机内部如何表示和实现无关

3. 满二叉树：深度为 K,且有2K -1个结点的二叉树。

完全二叉树：对满二叉树的结点从上到下,从左到右进行依次进行编号,若有一棵二叉树的每一个结点都与深度为K的满二叉树中编号都一一对应时, 只是最后一层不满,称做完全二叉树。

四、应用题**（每小题6分，共30分）**

**1.** 因为装填因子为0.7，有7个元素，故散列表长m=7/0.7=10。

H(7)=（7×3） Mod 7=0 H(8)=（8×3） Mod 7=3 H(30)=（30×3） Mod 7=6

H(11)=（11×3） Mod 7=5 H(18)=（18×3） Mod 7=5（冲突）

H(9)=（9×3） Mod 7=6（冲突） H(14)=（14×3） Mod 7=0（冲突）

（1）把关键字依次填入下表，并统计每个关键字查找成功时比较次数（8分）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 关键字 | 7 | 14 |  | 8 |  | 11 | 30 | 18 | 9 |  |
| 比较次数 | 1 | 2 |  | 1 |  | 1 | 1 | 3 | 3 |  |

（2）查找成功时ASL： （2分）

ASL成功=（1+2+1+1+1+1+1+3+3）/7=12/7。

**2.** prim算法求最小生成树如下

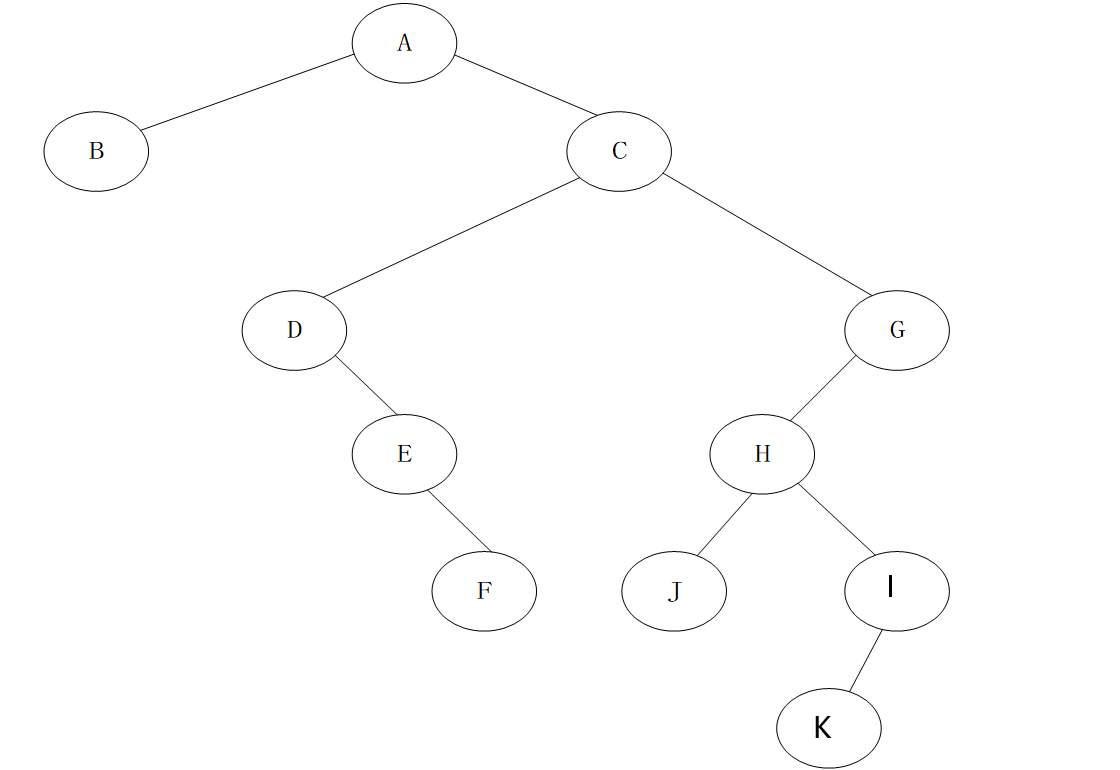


3. 哈夫曼树构造过程：



WPL=(6+7+9)\*2+5\*3+(2+3)\*4=44+15+20=79

4.



5．快速排序的每一趟的过程：

初始序列 66, 13, 51, 76, 81, 26, 57, 69

[57 13 51 26] 66 [81 76 69]

[26 13 51] 57 66 [81 76 69]

[13] 26 [51] 57 66 [81 76 69]

13 26 51 57 66 [69 76] 81

13 26 51 57 66 69 [76] 81

**五、设计题（每小题3分，共15分）**

1. void Insert\_sq(Sqlist va[], ElemType x)

{int i, j, n;

n=length(va[]);

if(x>=va[i])

va[n]=x;

else

{i=0;

while(x>va[i]) i++;

for(j=n-1;j>=I;j--)

va[j+1]=va[j];

va[i]=x; }

n++;

}

2. void quickpass(int r[], int s, int t)

{

int i=s, j=t, x=r[s];

while(i<j){

while (i<j && r[j]>x) j=j-1; if (i<j) {r[i]=r[j];i=i+1;}

while (i<j && r[i]<x) i=i+1; if (i<j) {r[j]=r[i];j=j-1;}

}

r[i]=x;

}

3. int cal\_height(BiTreeNode \* root){

int lheight=0,rheight=0;

if (root==NULL) return 0;

lheight=cal\_height(root->lchild);

rheight=cal\_height(root->rchild);

if (lheight>rheight) return lheight+1;

else return rheight+1;

}