《数据结构》参考答案

一、选择题**（每小题2分，共20分）**

1-5 ADBDB

6-10 ADCCB

二、填空题**（每小题2分，共20分）**

1、 确定性 输入

2、10

3、213 321

4、减少不必要的存储空间

5、31

6、n+1

7、n(n-1)/2 n(n-1)

8、9, 4, 6, 7, 8

9、堆排序 归并排序

10、12，22，34，17，35，56，75

三、名词解释题**（每小题5分，共15分）**

1. 单链表：每一个数据元素，都需用两部分来存储：一部分用于存放数据元素值，称为数据域；另一部分用于存放直接后继结点的地址（指针），称为指针域，元素的存储空间可以连续，也可以是不连续的。而数据元素之间的逻辑关系由指针域来确定。

双向链表：线性表采用链式存储时，每个结点除一个数据域外，包含两个指针域，一个指向该结点的直接后继，一个指向该结点的直接前驱，这种方式构成的链表，即为双向链表。

2. 树：树是n（n>=0）个结点的有限集。在任意一棵非空树中

（1）有且仅有一个特殊的称为根的结点

（2）当n>1时，其余结点可分成m（m>0）个互不相交的有限集T1，T2，，Tm，其中每一个集合本身又是一棵树，并且称为根的子树。

孩子结点和双亲结点：树中某个结点的子树的根结点称为该结点的孩子结点。相反，称该结点为孩子结点的双亲结点。

二叉树：二叉树是每个结点至多有两个孩子结点的一种树。其中两个孩子结点分别被称为左孩子结点和右孩子结点。

3. 结点的度：树的某个结点的分支（子树）个数叫做该结点的度。

树的度：树的度是树中所有结点的最大度数

平衡因子：结点的左子树深度与右子树深度之差。

四、应用题**（每小题6分，共30分）**

**1.** H(19)=19 Mod 11=8 H(14)=14 Mod 11=3 H(23) =23 Mod 11=1

H(10) =10 Mod 11=10 H(68)=68 Mod 11=2 H(20) =20 Mod 11=9

H(84) =84 Mod 11=7

H(24) =24 Mod 11=2（冲突）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | 23 | 68 | 14 | 24 |  |  | 84 | 19 | 20 | 10 |
|  | 1 | 1 | 1 | 3 |  |  | 1 | 1 |  | 1 |

所以平均查找长度是：ASL=（1+1+1+3+1+1+1+1）/8=10/8

**2.**  kruskal算法的最小生成树

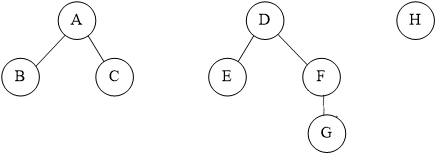


3. （1）哈夫曼树构造过程



（2） 带权路径长度：WPL=30\*1+16\*2+9\*3+5\*4+(1+3)\*5=30+32+27+20+20=129

1. 二叉树所表示的森林：



5. 初始：54，23，89，48，64，50，25，90，34

1：（23，54），89，48，64，50，25，90，34

2：（23，54，89），48，64，50，25，90，34

3：（23，48，54，89），64，50，25，90，34

4：（23，48，54，64，89），50，25，90，34

5：（23，48，50，54，64，89），25，90，34

6：（23，25，48，50，54，64，89），90，34

7：（23，25，48，50，54，64，89，90），34

8：（23，25，34，48，50，54，64，89，90）

**五、设计题（每小题3分，共15分）**

1. void merge(Lnode \*L1, Lnode \*L2)

{Lnode \*p,\*q ;

while(p->next!=L1)

p=p->next;

while(q->next!=L2)

q=q->next;

q->next=L1; p->next =L2;

}

2. void assending(Lnode \*head)

{Lnode \*p,\*q , \*r, \*s;

p=head->next; q=p->next; p->next=NULL;

while(q)

{r=q; q=q->next;

if(r->data<=p->data)

{r->next=p; head->next=r; p=r; }

else

{while(!p && r->data>p->data)

{s=p; p=p->next; }

r->next=p; s->next=r;}

p=head->next; }

}

3. void algo2(BTNode \*bt){

if (bt){

if(bt->data>=’a’&& bt- >data<=’z’)

bt->data-=32;

algo2(bt->lchild);

algo2(bt->rchild);

}

}