《数据结构》参考答案

一、选择题**（每小题2分，共20分）**

1-5 BAAAD

6-10 DBBCB

二、填空题**（每小题2分，共20分）**

1、顺序存储结构、链式存储结构

2、9，501

3、5

4、O(n2)，O(nlog2n)

5、0

6、e=d

7、(1，4，3，2)

8、7

9、O(1)

10、(5，16，71，23，72，94，73)

三、名词解释题**（每小题5分，共15分）**

1. 最大堆：如果一棵完全二叉树的任意一个非终端结点的元素都不小于其左儿子结点和右儿子结点（如果有的话）的元素，则称此完全二叉树为最大堆。

最小堆：如果一棵完全二叉树的任意一个非终端结点的元素都不大于其左儿子结点和右儿子结点（如果有的话）的元素，则称此完全二叉树为最小堆。

2. 二叉排序树： 二叉排序树或者是一棵空树，或者是具有下列性质的二叉树：

1、若它的左子树非空，则左子树上的所有结点的值均小于它的根结点的值。

2、若它的右子树非空，则右子树上的所有结点的值均大于或等于它的根结点的值。

3、它的左右子树分别为二叉排序树

3. 连通图：在无向图G中，如果从顶点v到顶点v1有路径，则称v和v1是连通的。如果对于图中任意两个顶点vi、vj∈V，vi和vj都是连通的，则称G是连通图。

连通分量：指的是无向图中的极大连通子图。

四、应用题**（每小题6分，共30分）**

1、H(32)=3\*32 Mod 11=8 H(13)=3\*13 Mod 11=6 H(49)=3\*49 Mod 11=4

H(24)=3\*24 Mod 11=6（冲突） H(38)=3\*38 Mod 11=4（冲突）

H(21)=3\*21 Mod 11=8（冲突） H(4)=3\*4 Mod 11=1 H(12)=3\*12 Mod 11=3

（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 | 10 |
| 关键字 |  | 4 |  | 12 | 49 | 38 | 13 | 24 | 32 | 21 |  |
| 比较  次数 |  | 1 |  | 1 | 1 | 2 | 1 | 2 | 1 | 2 |  |

（2）查找成功时ASL： （2分）

ASL成功=（1+1+1+2+1+2+1+2）/8=11/8。

2. 拓扑排序如下：

v1, v2, v4, v6, v5, v3, v7, v8 v1, v2, v4, v6, v5, v7, v3, v8

v1, v2, v6, v4, v5, v3, v7, v8 v1, v2, v6, v4, v5, v7, v3, v8

v1, v6, v2, v4, v5, v3, v7, v8 v1, v6, v2, v4, v5, v7, v3, v8

3.

WPL=3\*4+5\*4+11\*3+23\*2+29\*2+14\*3+7\*4+8\*4=271

4.



5. 答案：

第一次：[24 40 38] 46 [56 80 95 79]

第二次： 24 [38 40] 46 [56 80 95 79]

第三次： 24 38 40 46 [56 80 95 79]

第四次： 24 38 40 46 56 [80 95 79]

第五次： 24 38 40 46 56 79 [80 95]

第六次: 24 38 40 46 56 79 80 95

**五、设计题（每小题3分，共15分）**

1. void headInPrint(LinkList\* L){

LinkList \*p;

p = L->next;

L->next = NULL;

while(p!=NULL){

LinkList\* temp = (LinkList\*)malloc(sizeof(LinkList\*));//保持不断链

temp = p->next;

//头插法

p->next = L->next; //p节点指向node2

L->next = p;

p = temp;

}

printList(L);

}

2. LinkList\* del\_min(LinkList\* L){

LinkList\* pre;//前驱节点

LinkList\* p;//工作节点

LinkList\* minpre;//最小值前驱节点

LinkList\* min;//最小值节点

pre = L,minpre = pre;

p = L->next,min = p;

if(p==NULL)

return 0;

while(p!=NULL){

if(p->data<min->data){ //如果当前节点小于最小值，min,minpre移动到当前位置

minpre = pre;

min = p;

}else{

p = p->next;

pre = pre->next;

}

}

//删除min节点

minpre->next = min->next;

free(min);

return L;

}

3. //假设二叉树结构体如下

struct binTree

{

int data;

binTree \*lchild;

binTree \*rchild;

}\*BiTree;

//函数如下

BiTree find(BiTree node, int x)

{

if(node)

{

if(node->data==x) delete(node);

else

{

find(node->lchild);

find(node->rchild);

}

}

}

BiTree delete(BiTree tree)

{

if(node)

{

delete(node->lchild);

delete(node->rchild);

free(node);

node=NULL;

}

}