**计算机2002-123数据结构试题参考答案**

**一、单项选择题（在每个小题四个备选答案中选出一个正确答案，填在题末的括号中）（本大题共10小题，每小题2分，总计20分）**

1、D 2、B 3、B 4、B 5、C 6、B 7、C 8、D 9、A 10、D

**二、二、**填空**（本大题共10小题，每小题1分，总计10分）**

1、连通分量 2、结点个数/2 3、6 1 4、（n-1）/2

5、EF-G\*ABC-/+D\* 6、1140 7、2K-1 2K-1

8、堆栈 9、动态查找表 10、先序遍历 中序遍历

**三、解答下列问题（本大题共6小题，每小题5分，总计30分）**

1、因为：H（19）=19 MOD 13=6

H（14）=14 MOD 13=1

H（23）=23 MOD 13=10

H（1）=1 MOD 13=1 冲突

所以 （H（1）+1）MOD 16=2

H（68）=68 MOD 13=3

H（20）=20 MOD 13=7

H（84）=84 MOD 13=6 冲突

所以 （H（84）+1）MOD 16=7 冲突

所以 （H（84）+2）MOD 16=8

H（27）=27 MOD 13=1 冲突

所以 （H（27）+1）MOD 16=2 冲突

所以 （H（27）+2）MOD 16=3 冲突

所以 （H（27）+3）MOD 16=4

H（55）=27 MOD 13=3 冲突

所以 （H（55）+1）MOD 16=4 冲突

所以 （H（55）+2）MOD 16=5 冲突

H（11）=11 MOD 13=11

H（10）=10 MOD 13=10 冲突

所以 （H（10）+1）MOD 16=11 冲突

所以 （H（10）+2）MOD 16=12 冲突

H（79）=79 MOD 13=1 冲突

所以 （H（79）+1）MOD 16=2 冲突

所以 （H（79）+2）MOD 16=3 冲突

所以 （H（79）+3）MOD 16=4 冲突

所以 （H（79）+4）MOD 16=5 冲突

所以 （H（79）+5）MOD 16=6 冲突

所以 （H（79）+6）MOD 16=7 冲突

所以 （H（79）+7）MOD 16=8 冲突

所以 （H（79）+8）MOD 16=9 冲突

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 14 | 1 | 68 | 27 | 55 | 19 | 20 | 84 | 79 | 23 | 11 | 10 |  |  |  |

1 2 1 4 3 1 1 3 9 1 1 3

（2）查找成功的平均查找长度为：

1/12\*（1+2+1+4+3+1+1+3+9+1+1+3）=2.5

2、

3、

0 1

0 1 0 1

0 1 0 1

0 1 0 1

编码为：7为：0000

8为：0001

18为：001

32为：01

3为：1000

5为：1001

12为：101

26为：11

4、

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i  closedg | 2 | 3 | 4 | 5 | **6** | **U** | **V-U** | K |
| Adjvex  lowcost | **1**  **16** | 1  ∞ | 1  ∞ | 1  19 | 1  21 | {1} | {2，3，4，5，6} | 2 |
| Adjvex  lowcost | **0** | **2**  **5** | **1**  **6** | 1  19 | 2  11 | {1，2} | {3，4，5，6} | 3 |
| Adjvex  lowcost | **0** | 0 | **2**  **6** | 1  19 | 2  11 | {1，2，3} | {4，5，6} | 4 |
| Adjvex  lowcost | **0** | **0** | **0** | 4  18 | **2**  **11** | {1，2，3，4} | {5，6} | 6 |
| Adjvex  lowcost | **0** | **0** | **0** | **4**  **18** | **0** | {1，2，3，4，6} | {5 } | 5 |
| Adjvex  lowcost | **0** | **0** | **0** | **0** | **0** | {1，2，3，4，6，5} | { } |  |

所以最小生成树为：

16 16

5 5

11 6 11

18 18 6

5、

第一趟： [68 11 69 23 18] 70 [93 73]

第二趟：[18 11 23 ] 68 [ 69 ] 70 [93 73 ]

第三趟：11 18 23 68 69 70 [ 93 73]

第四趟：11 18 23 68 69 70 73 93

**四、算法设计题（本大题共2小题，总计15 分）**

1、（7分）

#define size 100

int correct(char exp[size],int len)

{ char s[size];

int top=0,I=0,tag=1;

while(I<len && tag!=0)

{ if(exp[I]==’(‘ || exp[I]==’[‘ || exp[I]==’{‘}

{ top++;

s[top]=exp[I];}

if(exp[I]==’)’){ if(s[top]==’(‘) top--;

else tag=0;}

if(exp[I]==’)’){if(s[top]==’[‘ ) top--;

else tag=0;}

if(exp[I]==’)’){if(s[top]==’{‘ } top--;

else tag=0;}

I++;}

if(top>0)top=0;

return tag;}

2、（8分）

#include "iostream.h"

#define NULL 0

#define N 10

typedef struct node

{ ELEMTP data;

struct node \*lchild,\*rchild;

}TNode;

//求给定结点的所有祖先和祖先数

int count\_zx(TNode \*t,ELEMTP x)

{ struct {

TNode \*pp; int tag; }s[N],ss;

int top,n;

TNode \*p; top=0; n=0; p=t;

while( p || top>0)

{ while(p)

{ top++;

s[top].pp=p;s[top].tag=0;

p=p->lchild; }

ss=s[top]; top--;

if(ss.tag==0)

{ ss.tag=1; top++;

s[top]=ss; p=ss.pp->rchild;

}

else {

if(ss.pp->data==x)

break; p=NULL;

} }

cout<<"the zx wei:"<<endl;

for(n=1;n<=top;n++) {

p=s[n].pp;

cout<<p->data<<" "; }

return top;

}

**五．（10分**）已知二叉树每个非终端节点都有左孩子和右孩子，试回答下列问题：

解：（1）由已知：因为二叉树的每个非终端结点都有左右孩子，可知此二叉树的所有非终端结点的度都为2，无度为1的结点，又因为此树有n个叶结点，根据二叉树的性质3知度都为2的结点数为n-1个，所以此二叉树共有n+n-1=2n-1个结点。

1. 证明 ：对n用数学归纳法证明

当n=1时，由题意知该叶结点只能为根结点，所以其所在的层数为l1=1,带入左端得：1/2l1-1=1/20=1 由此知等式成立。

假设当n=k时 等式成立 即： 1/2l1-1+1/2l2-1+……+1/2lj-1+…+1/2lk-1=1

下面证明当n=k+1时也成立。

因为该二叉树有k+1个叶结点，又由于该二叉树无度为1的非终端节点，所以有k+1个叶结点的二叉树一定是从有k个叶结点的二叉树中的某一个叶结点给它增加两个子结点，使该叶结点变为非终端节点而形成的，不妨设该叶结点原来所在层数为lj(1<= j <=k)

，则 新增添的两个结点的层数必为 lj+1

又因为 1/2lj+1-1+1/2lj+1-1=1/2lj-1

∴ 1/2l1-1+1/2l2-1+…+1/2lj+1-1+1/2lj+1-1+…+1/2lk-1= 1/2l1-1+1/2l2-1+……+1/2lj-1+…+1/2lk-1=1

由此知上述等式成立。