《数据结构》参考答案

一、选择题**（每小题2分，共20分）**

1-5 BDCBB

6-10 BACBD

二、填空题**（每小题2分，共20分）**

1、 (F+1) % m

2、O(n)，O(n)

3、2n，n+1

4、61

5、n, 2e

6、15

7、CBA

8、4，16

9、L->prior==L->next==L

10、1305

三、名词解释题**（每小题5分，共15分）**

1. 关键路径：在 AOE网中，由于有些活动可以并行，所以完成工程的最短时间是从源点到汇点的最大路径长度。因此，把从源点到汇点具有最大长度的路径称为关键路径。

2. 拓扑排序：给定一个无环路有向图 G=(V,E),各结点的编号为v=(1,2,i,j,n)。要求对每一个结点i重新进行编号，使得若i是j的前导，则有Label[ i ]<label[j ]。即拓扑分类是将无环路有向图排成一个线性序列，使当从结点i到结点j存在一条边，则在线性序列中，将i排在j的前面。

3. AOE网：在带权的有向图中，用结点表示事件，用边表示活动，边上权表示活动的开销（如持续时间），则称此有向图为边表示活动的网络, 简称 AOE网。

四、应用题**（每小题6分，共30分）**

**1.** H(32)=32 Mod 11=10 H(13)=13 Mod 11=2 H(49)=49 Mod 11=5

H(38)=38 Mod 11=5（冲突），H1=（5+12） Mod 11=6

H(21)=21 Mod 11=10（冲突），H1=（10+12） Mod 11=0

H(60)=60 Mod 11=5（冲突），H1=（5+12） Mod 11=6，H2=（5-12） Mod 11=4

H(12)=12 Mod 11=1

（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 | 21 | 12 | 13 |  | 60 | 49 | 38 |  |  |  | 32 |
| 比较  次数 | 2 | 1 | 1 |  | 3 | 1 | 2 |  |  |  | 1 |

（2）查找成功时ASL： （2分）

ASL成功=（2+1+1+3+1+2+1）/7=11/7。

**2.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 终点 | 从v0到各终点的d值和最短路径的求解过程 | | | |
| i=1 | i=2 | i=3 | i=4 |
| v1 | 12 (v0,v1) | 12 (v0,v1) | 7 (v0,v4,v1) |  |
| v2 | 4 (v0,v2) |  |  |  |
| v3 | 9 (v0,v3) | 8 (v0,v2,v3) | 7 (v0,v4,v3) | 7 (v0,v4,v3) |
| v4 | 5 (v0,v4) | 5 (v0,v4) |  |  |
| vj | v2 | v4 | v1 | v3 |
| s | {v0,v2} | {v0,v4} | {v0,v4,v1} | {v0,v4,v3} |

3.



4.

WPL=1\*4+2\*4+3\*3+8\*2+4\*3+5\*3+11\*2=86

5.

初始序列16 15 18 16 17 18 20 13

第1次排序15 16 16 17 18 18 13 [20]

第2次排序15 16 16 17 18 13 [18 20]

第3次排序15 16 16 17 13 [18 18 20]

第 4 次排序 15 16 16 13 [17 18 18 20]

第 5 次排序 15 16 13 [16 17 18 18 20]

第 6 次排序 15 13 [16 16 17 18 18 20]

第 7 次排序 13 [15 16 16 17 18 18 20]

**五、设计题（每小题3分，共15分）**

1.

void f2(SeqList \*L){

int i,j,k;

k=0;

for(i=0;i<L->length;i++){

for(j=0;j<k && L->data[i]!=L->data[j];j++);

if(j==k){

if(k!=i) L->data[k]=L->data[i];

k++;

}

}

L->length=k;

}

2. void mergelklist(lklist \*ha,lklist \*hb,lklist \*&hc)

{

lklist \*s=hc=0;

while(ha!=0 && hb!=0)

if(ha->data<hb->data){if(s==0) hc=s=ha; else {s->next=ha; s=ha;};ha=ha->next;}

else {if(s==0) hc=s=hb; else {s->next=hb; s=hb;};hb=hb->next;}

if(ha==0) s->next=hb; else s->next=ha;

}

3. typedef struct node {datatype data; struct node \*lchild,\*rchild;} bitree;

int judgebitree(bitree \*bt1,bitree \*bt2)

{

if (bt1==0 && bt2==0) return(1);

else if (bt1==0 || bt2==0 ||bt1->data!=bt2->data) return(0);

else return(judgebitree(bt1->lchild,bt2->lchild)\*judgebitree(bt1->rchild,bt2->rchild));

}