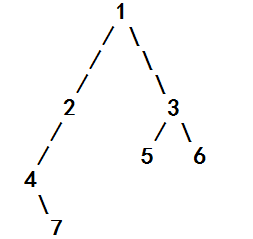
（1）已知一棵二叉树的前序遍历的结果是1247356，中序遍历的结果是4721536，试画出这棵二叉树。

解：



（2）已知一个无向图的邻接表如下图所示，画出这个图。

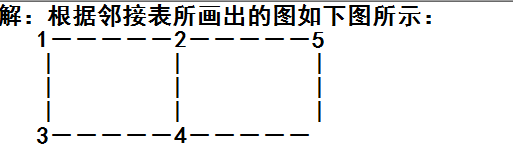
0 [1][-]-→[1][-]-→[2][^]

1 [2][-]-→[0][-]-→[3][-]-→[4][^]

2 [3][-]-→[0][-]-→[3][^]

3 [4][-]-→[1][-]-→[2][-]-→[4][^]

4 [5][-]-→[1][-]-→[3][^]



上图4直接连到5

（3）设待排序的排序码序列为{5.8,4.6,7.2,9.5,8.4,2.5,3.7,5.8,6.3,1.2}，试写出使用希尔排序方法时每趟排序后的结果（增量为5,3,1）。

解：希尔排序（增量为5，2，1）每趟排序后的结果如下所示：

初始序列：5.8, 4.6, 7.2, 9.5, 8.4, 2.5, 3.7, 5.8, 6.3, 1.2

第 1 趟（d=5），排序结果：

2.5, 3.7, 5.8, 6.3, 1.2, 5.8, 4.6, 7.2, 9.5, 8,4

第 2 趟（d=2），排序结果：

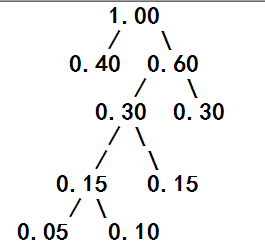
1.2, 3.7, 2.5, 5.8, 4.6, 6.3, 5.8, 7.2, 9.5, 8,4

第 3 趟（d=1），排序结果：

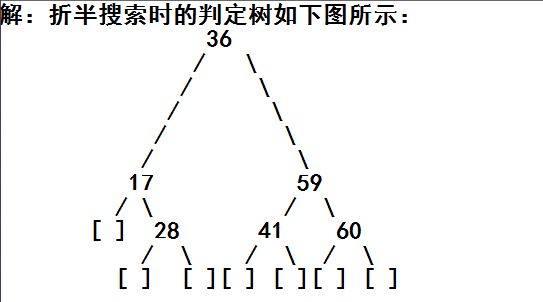
1.2, 2.5, 3.7, 4.6, 5.8, 5.8, 6.3, 7.2, 8,4, 9.5

（4）给定权值集合{0.05,0.15,0.40,0.30,0.10}，构造相应的Huffman树。

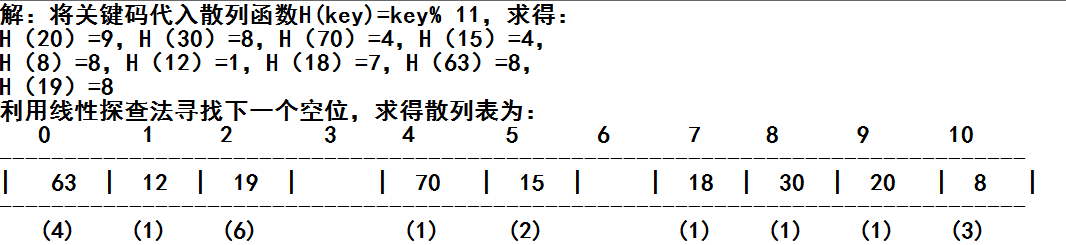
解：



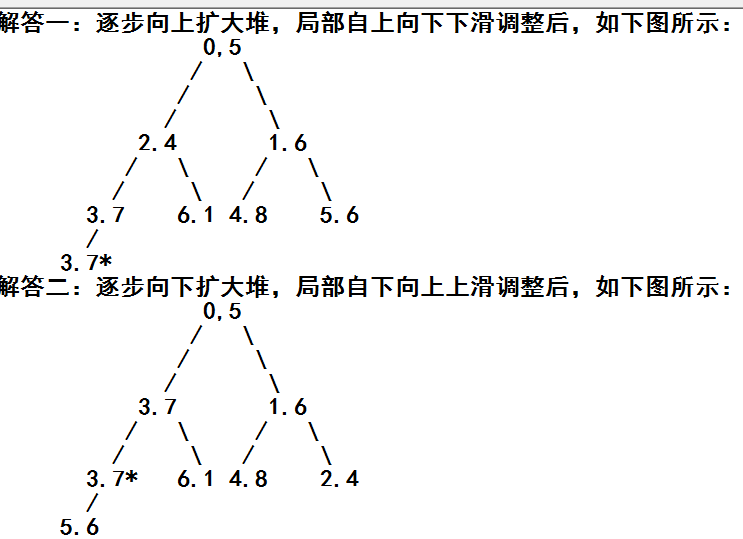
（5）设有序顺序表为{17,28,36,41,59,60}，试画出对其进行折半搜索时的判定树。



（6）设散列表为HT[11]，散列函数为H(key)=key%11。用闭散列法解决冲突，对以下关键码序列{20,30,70,15,8,12,18,63,19}，采用线性探查法寻找下一个空位，画出相应的散列表。

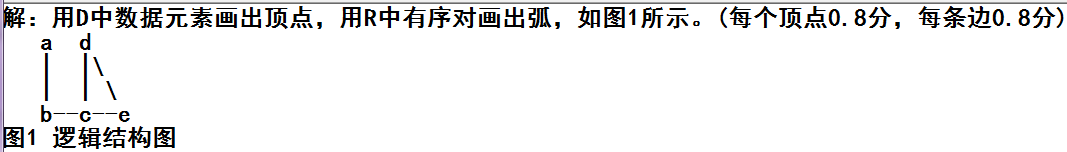


（7）有如下序列{5.6,3.7,4.8,2.4,6.1,0.5,1.6,3.7\*}，将它调整为最小堆。



（8）设数据逻辑结构如下：DS＝{D, S}，其中，D＝{a,b,c,d,e}，S＝{R}，

R＝{(a,b),(b,c),(c,d),(c,e),(d,e)}，试画出DS所对应的逻辑结构图。（本题8分）



（9）已知一个栈S的输入序列为abcd，下面两个序列能否通过栈的Push和Pop操作输出；如果能，请写出操作序列；如果不能，请说明原因。（本题5分）

(1) dbca

(2) cbda

参考答案（本题5分）

（1）答：不能实现（结论1分）。由于最先d出现，要求abcd先入栈，由栈的特点，出栈序列应为dcba（说理1分）。

（2）答：可以实现（结论1分）。操作序列为：Push(a)、Push(b)、Push(c)、Pop(c)、Pop(b)、Push(d)、Pop(d)、Pop(a) 。（评分标准： Push和Pop操作系列占2分。没有写Push和Pop操作系列，此项不得分。从第1个算起，每个正确操作为0.25分，出现错误操作后，后面操作都不得分）

（10）设广义表L=（（），（e），（a，（b，c，d））），试求GListLength(L)、GListDepgh(L)、GetHead(L)、GetTail(L)的值。（本题4分）

参考答案（每项1分，共4分）

解：

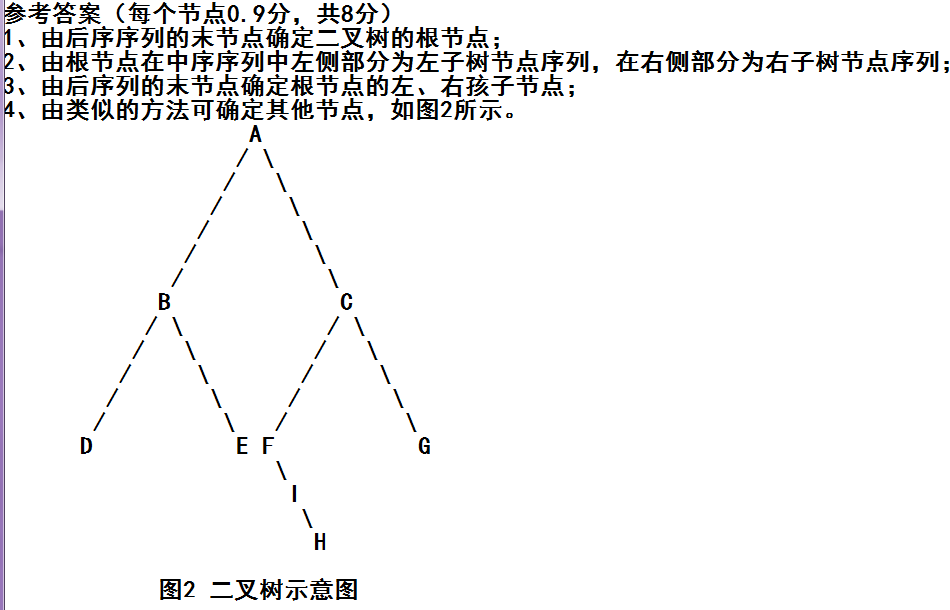
GListLength(L)＝3；

GListDepgh(L)＝3；

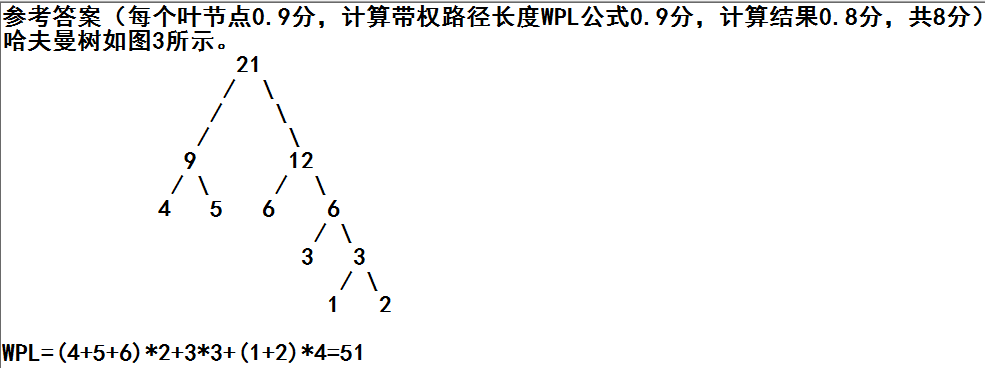
GetHead(L)＝（）；

GetTail(L)＝（（e），（a，（b，c，d））)

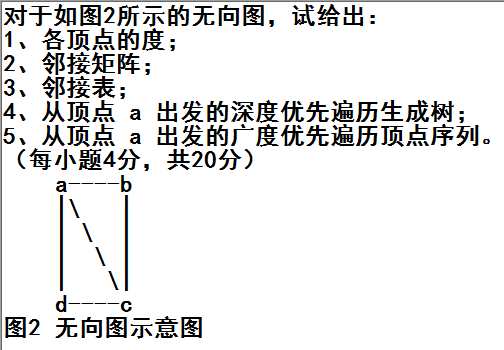
（11）已知一棵二叉树中序序列为：DBEAFIHCG，后序序列为；DEBHIFGCA，请画出此二叉树。（本题8分）

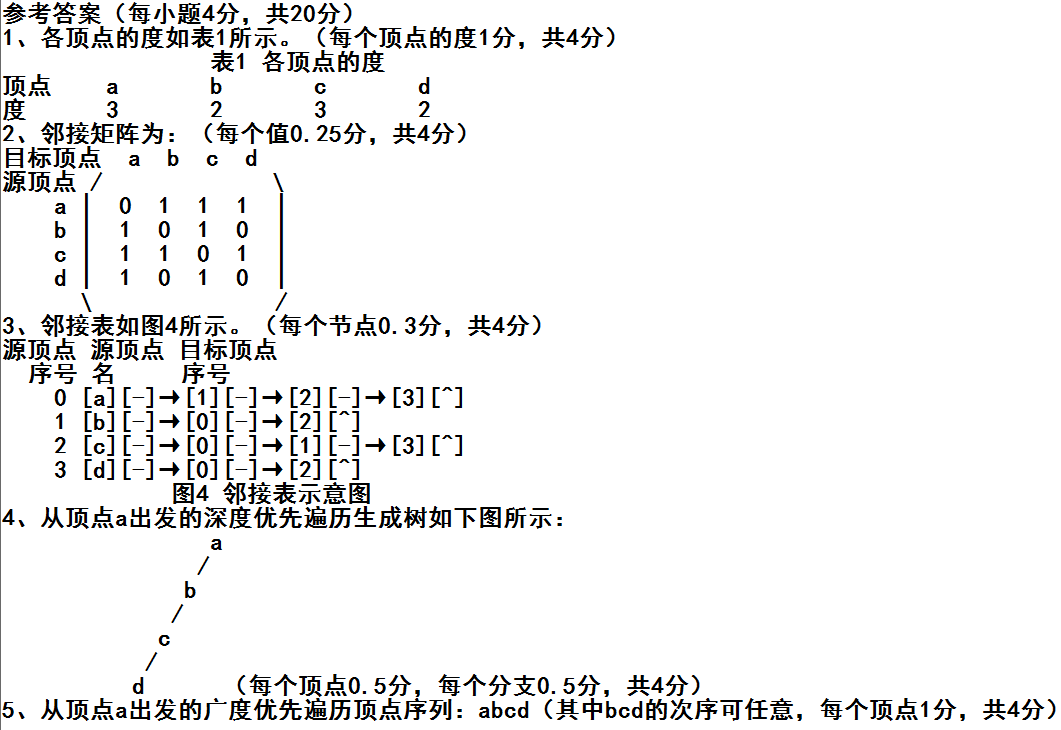


（12）给定一组权值{1，2，3，4，5，6}，试画出相应的哈夫曼树，并计算带权路径长度WPL的值。（本题8分）

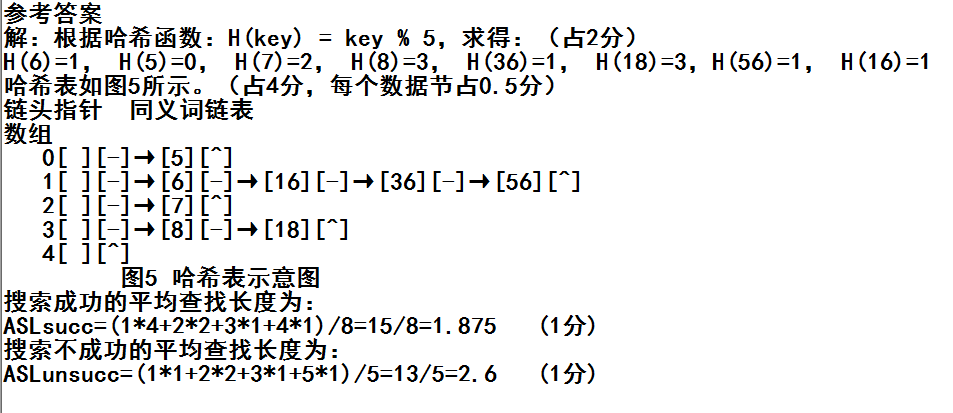


（13）

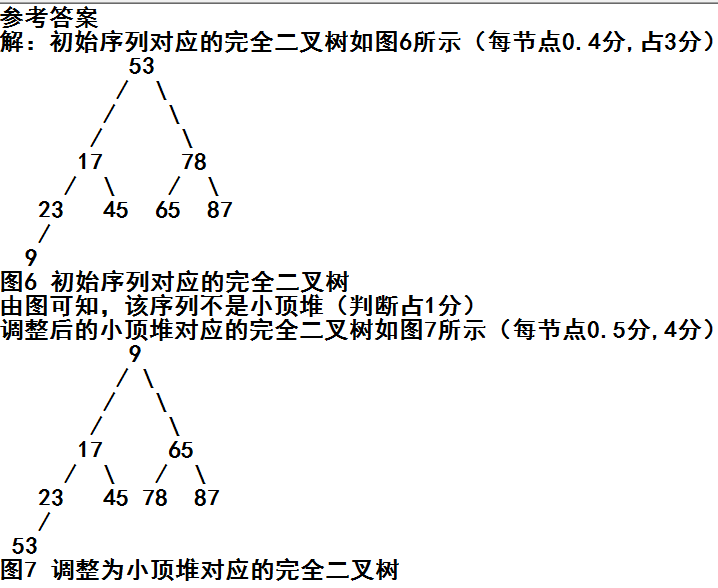




（14）已知一组关键字为（6，5，7，8，36，18，56，16），哈希函数：H(key) = key % 5，采用链地址法处理冲突，试构造哈希表，并求平均查找长度ASL。（本题8分）



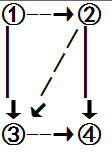
（15）判别序列（53，17，78，23，45，65，87，9）是否为小顶堆，如果不是，则将它调整为小顶堆。要求画出初始序列对应的完全二叉树和调整为小顶堆对应的完全二叉树。（本题8分）



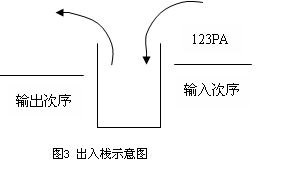
（16）有如下数据元素的形式定义，试画出此结构的图形表示。（本题8分）

DS＝{D, S}

其中，D＝{1,2,3,4}，S＝{R}，R＝{<1,2>,<1,3>,<2,3>,<2,4>,<3,4>}



（17）如图3所示，设输入元素为1、2、3、P和A，输入次序为123PA，元素经过栈后到达输出序列，当所有元素均到达输出序列后，有哪些序列可以作为高级语言的变量名？（本题5分）



解：

1、P为第1个输出字符的情形：PA321、P3A21、P32A1、P321A

2、A为第1个输出字符的情形：AP321

评分标准：每个正确完整的5个字符变量名得1分，每个不足5个字符的变量名扣0.5分，每个错误的变量名扣1分，负分以0分计算。

（18）设广义表L=（a，（a，b），c，d，((i，j)，k)），试求GetHead(L)、GetTail(L)、GListLength(L)、GListDepgh(L)的值。（本题4分）

参考答案（每项1分，共4分）

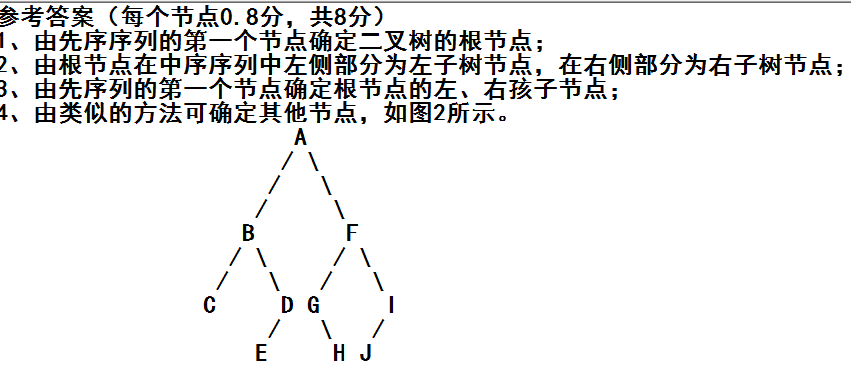
GetHead(L)＝a；

GetTail(L)＝（（a，b），c，d，((i，j)，k)）；

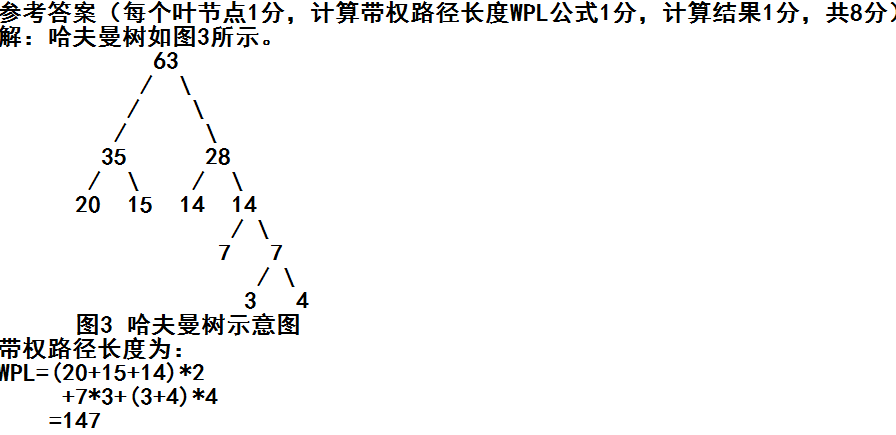
GListLength(L)＝5；

GListDepgh(L)＝3

（19）已知一棵二叉树先序序列为：ABCDEFGHIJ，中序序列为；CBEDAGHFJI，请画出此二叉树。（本题8分）



（20）给定一组权值{3，4，7，14，15，20}，试画出相应的哈夫曼树，并计算带权路径长度WPL的值。（本题8分）



（21）对于如图4所示的无向图，试给出：

1、各顶点的度；

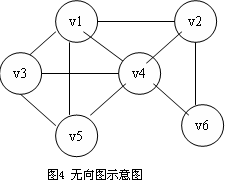
2、邻接矩阵；

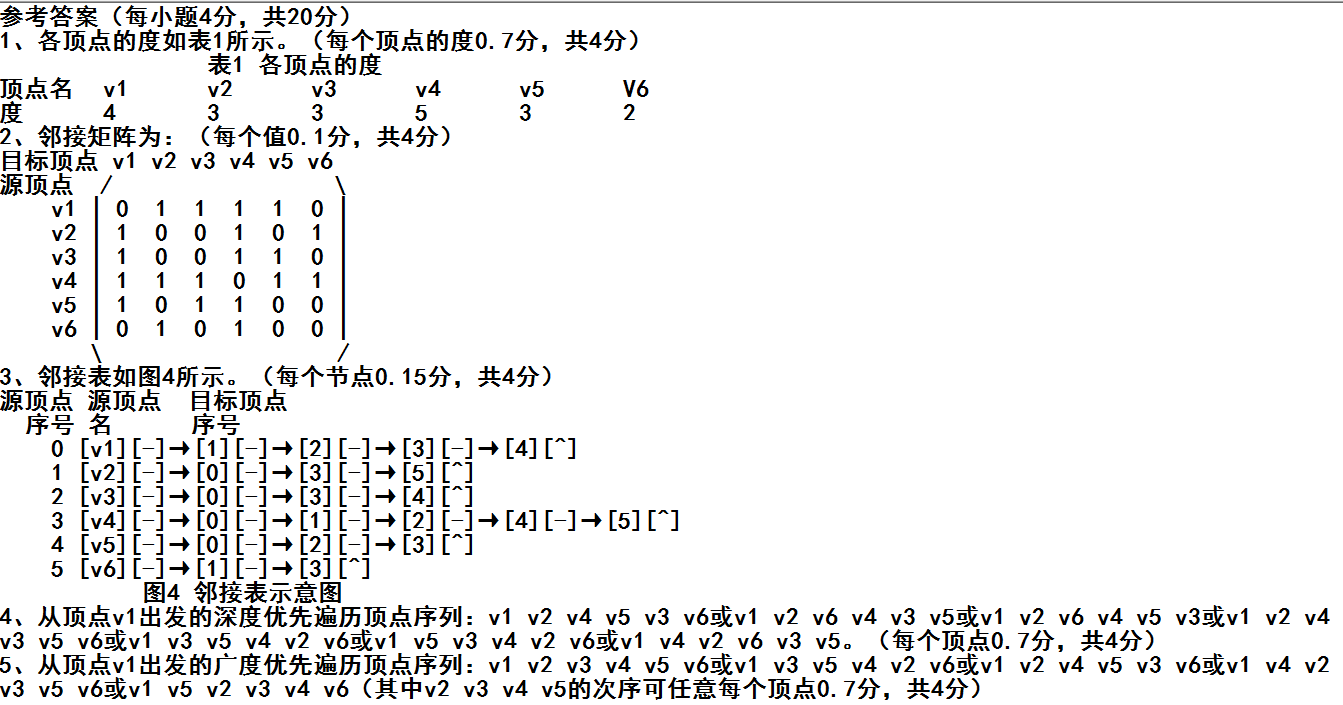
3、邻接表；

4、从顶点v1出发的深度优先遍历顶点序列；

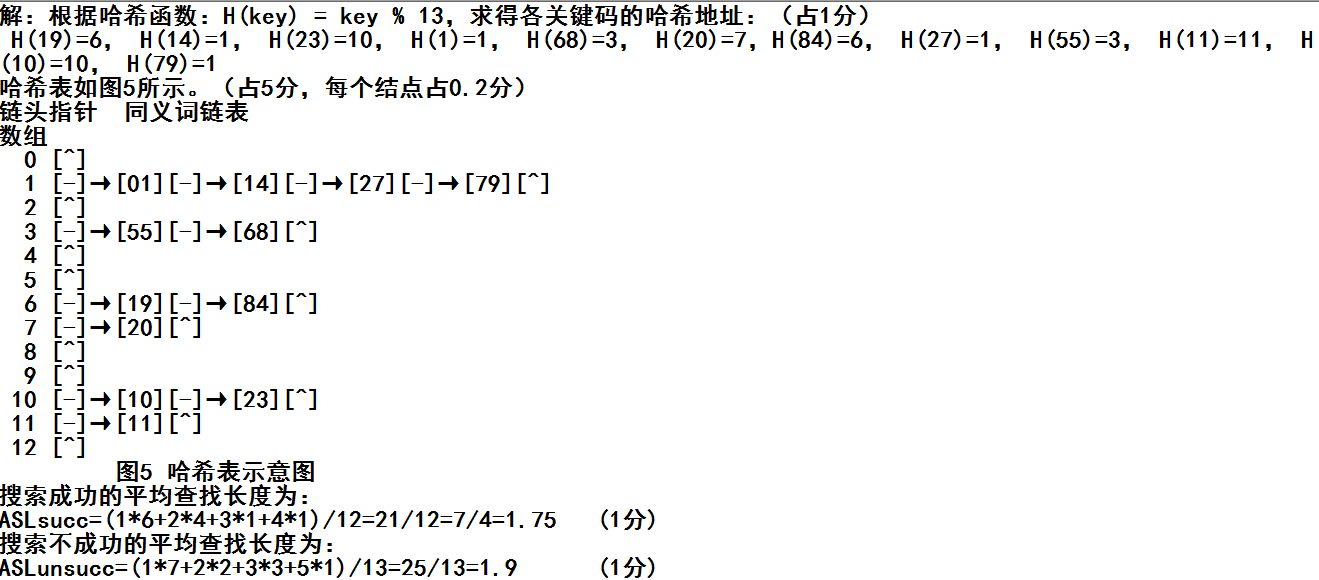
5、从顶点v1出发的广度优先遍历顶点序列。

（每小题4分，共20分）

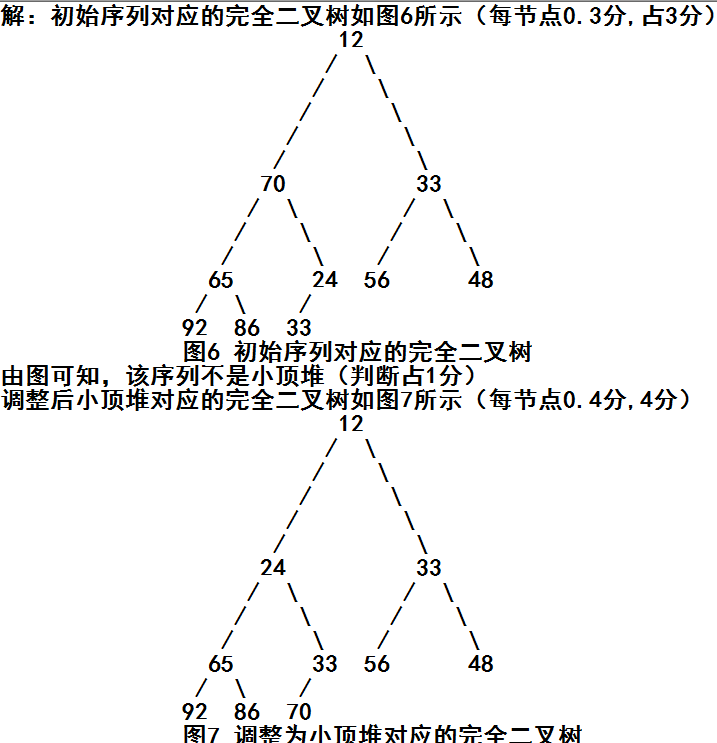




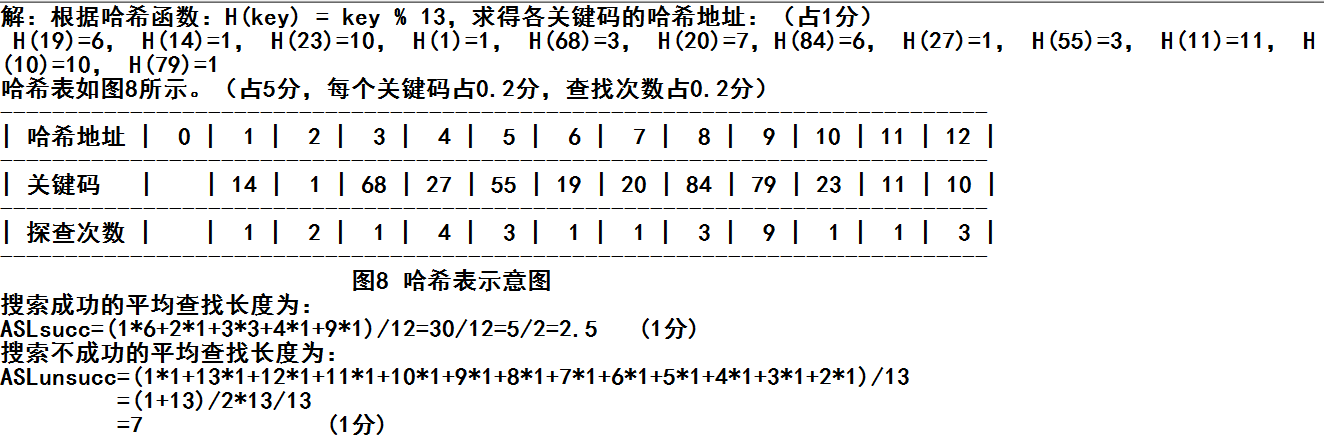
（22）已知一组关键字为（19，14，23，1，68，20，84，27，55，11，10，79），哈希函数：H(key) = key % 13，哈希地址空间为0~12,请构造用链地址法处理冲突的哈希表，并求平均查找长度ASL。（本题8分）



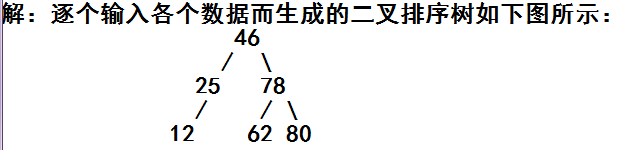
（23）判别序列（12，70，33，65，24，56，48，92，86，33）是否为小顶堆，如果不是，则将它调整为小顶堆。要求画出初始序列对应的完全二叉树和调整为小顶堆对应的完全二叉树。（本题8分）



（24）已知一组关键字为（19，14，23，1，68，20，84，27，55，11，10，79），哈希函数：H(key) = key % 13，哈希地址空间为0~12,请构造用线性探查法处理冲突的哈希表，并求平均查找长度ASL。（本题8分）



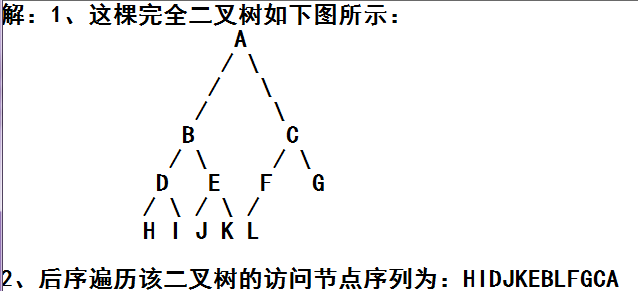
（25）设有一个输入数据的序列是{46,25,78,62,12,80}，试画出从空树起，逐个输入各个数据而生成的二叉排序树。



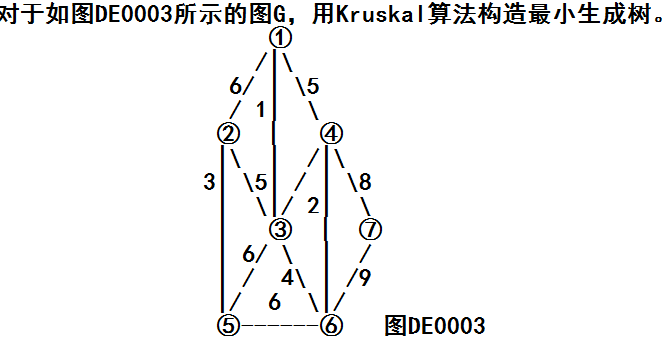
（26）用一维数组存放的一棵完全二叉树：ABCDEFGHIJKL。

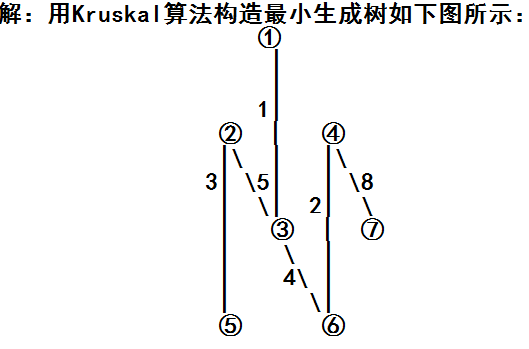
1、画出这棵完全二叉树。

2、写出后序遍历该二叉树的访问节点序列。



（27）





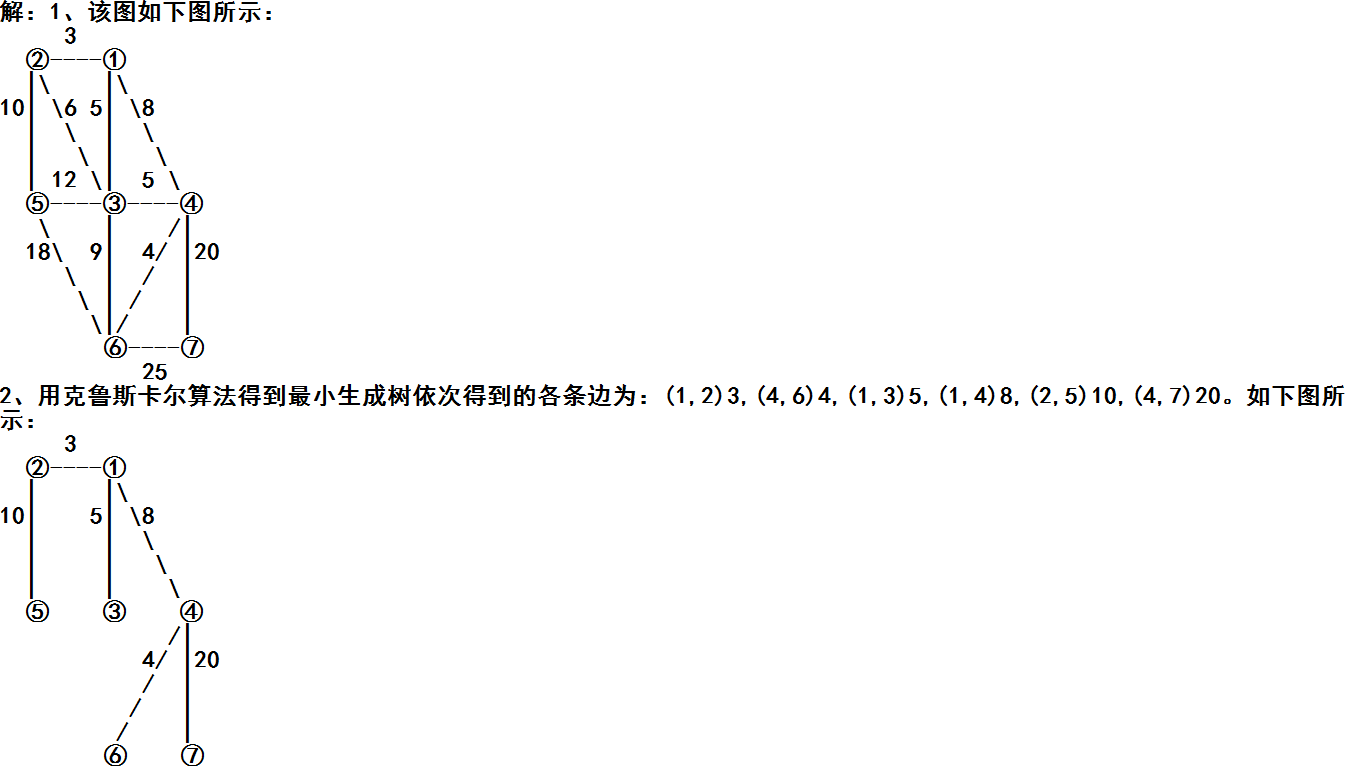
（28）已知一个图的顶点集V和边集E分别为：

V={1,2,3,4,5,6,7}；

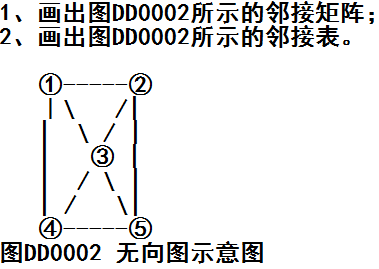
E={(1,2)3,(1,3)5,(1,4)8,(2,5)10,(2,3)6,(3,4)15,(3,5)12,(3,6)9,(4,6)4,(4,7)20,(5,6)18,(6,7)25}；

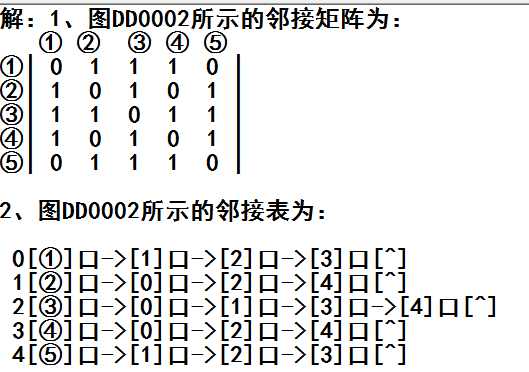
1、画出该图；

2、用克鲁斯卡尔算法得到最小生成树，试写出在最小生成树中依次得到的各条边。要求图示每一步的变化情况。

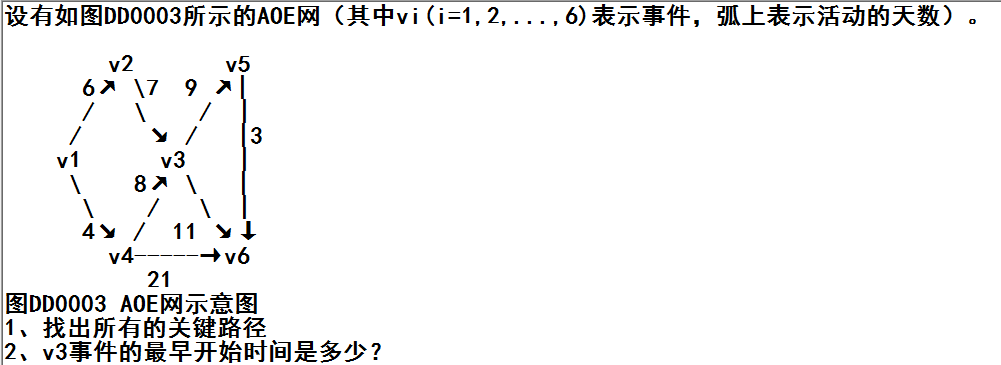


（29）





（30）



解：1、<v1,v4>,<v4,v6>,<v1,v2>,<v2,v3>,<v3,v5>,<v5,v6>是关键路径。

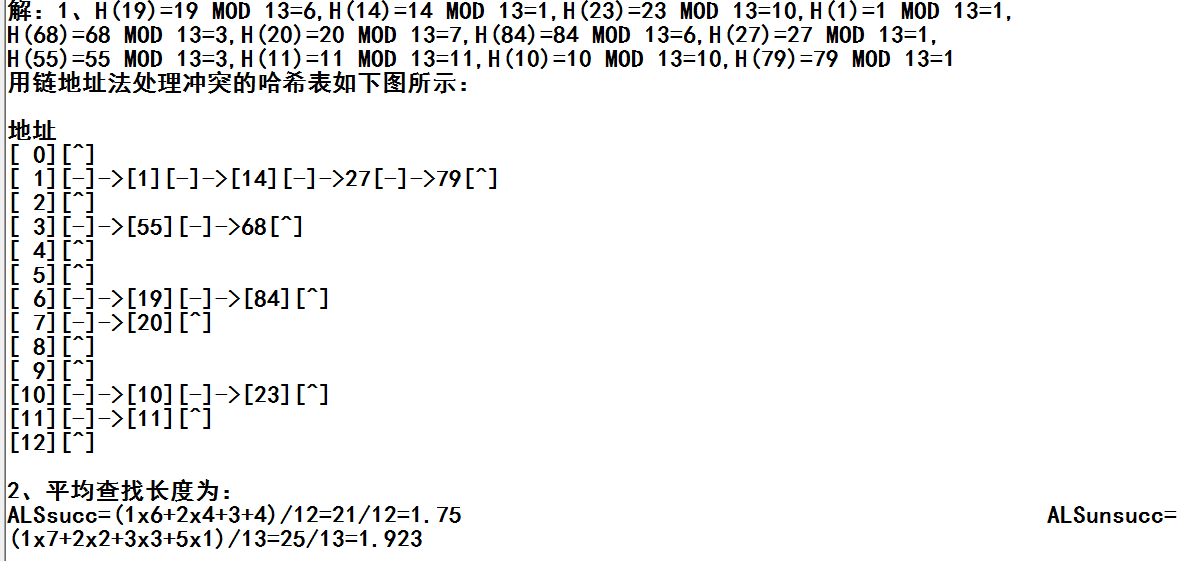
2、v3事件的最早开始时间

=Max{<v1,v2>+<v2+v3>,<v1,v4>+<v4,v3>}=Max{6+7,4+8}=13(天)

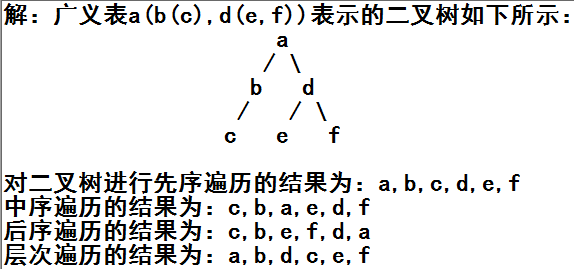
（31）已知一组关键字为(19,14,23,1,68,20,84,27,55,11,10,79)，哈希函数：H(key)=key MOD 13，哈希地址空间为0~12。

1、构造用链地址法处理冲突的哈希表。

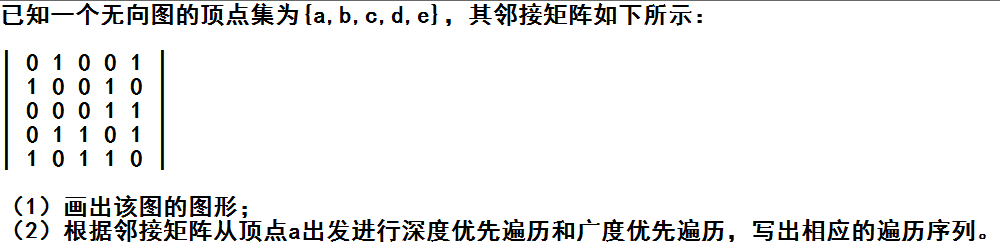
2、求出平均查找长度。



（32）假定一棵二叉树广义表表示为a(b(c),d(e,f))，分别写出对它进行先序、中序、后序、层次遍历的结果。

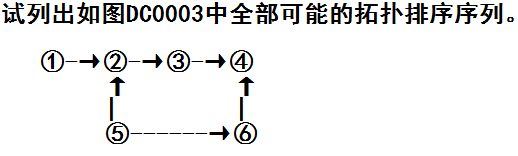


（33）



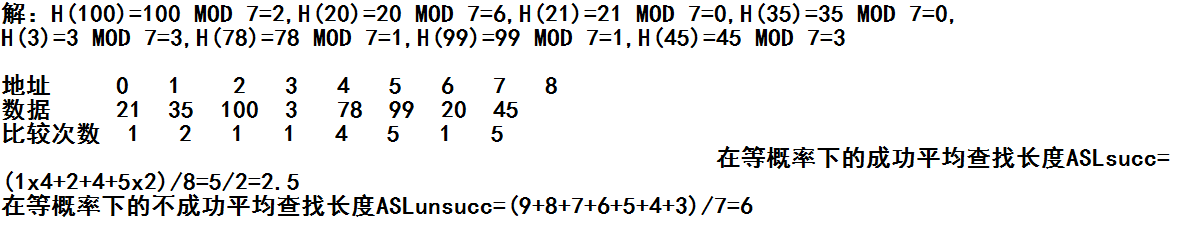


（34）



解：图DC0003全部可能的拓扑排序序列为：①⑤⑥②③④，①⑤②⑥③④，①⑤②③⑥④，⑤⑥①②③④，⑤①⑥②③④，⑤①②⑥③④，⑤①②③⑥④

（35）已知哈希表地址空间为0...8，哈希函数为H(key)=key MOD 7，采用线性探测再散列处理冲突，将数据序列{100,20,21,35,3,78,99,45}依次存入此哈希表中，列出插入时的比较次数，并求出在等概率下的平均查找长度。



（36） 已知一个图的顶点集V和边集E分别为：

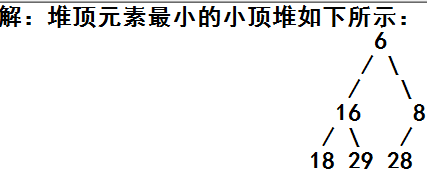
V={1,2,3,4,5,6,7}

E={<2,1>,<3,2>,<3,6>,<4,3>,<4,5>,<4,6>,<5,1>,<5,7>,<6,1>,<6,2>,<6,5>}；

若存储它采用邻接表，并且每个顶点邻接表中的边节点都是按照终点序号从小到大的次序链接的，试给出得到的拓扑排序的序列。

解：拓扑排序的序列为：4 3 6 2 5 1 7 或 4 3 6 2 5 7 1 或 4 3 6 5 7 2 1

（37）对于序列{8,18,6,16,29,28}，试写出堆顶元素最小的小顶堆。



（38）

设有关类已定义和实现，并已对其重载流输出运算符，请写出下面程序执行结果。

#include <iostream>

using namespace std;

#include ".\list.cpp"

int \_tmain()

{

List<int,int> sla(0),slb(0),slc(0);

int i,j;

for(i=0; i<5; i++)

sla.Insert(i,2\*i+1);

cout<<"sla: ";

sla.output();

for(j=0; j<4; j++)

slb.Insert(j,3\*j-2);

cout<<"slb: ";

slb.output();

int n1=sla.Length(), n2=slb.Length(),slaData,slbData;

i=1; j=1;

while(i<=n1 && j<=n2)

{

sla.getData(i,slaData);

slb.getData(j,slbData);

if(slaData<=slbData)

{

slc.Insert(0,slaData);

i++;

}

else

{

slc.Insert(0,slbData);

j++;

}

}

if(i<=n1)

while(i<=n1)

{

slc.Insert(0,slaData);

i++;

}

else

while(j<=n2)

{

slc.Insert(0,slbData);

j++;

}

cout<<"slc: ";

slc.output();

return 0;

}

解：

sla: 1,3,5,7,9,

slb:-2,1,4,7,

slc: 9,7,7,5,4,3,1,1,-2，

（39）针对带附加头结点的单链表，编写求最大值函数max：通过一趟遍历在单链表中确定值最大的结点。

解：// 求链表list中的值最大值的结点

template<typename T, typename E>

E max(const List<T,E>& list)

{

E maximum, Element; // 最大值和元素

// 链表中的第1个元素作为最初的最大值

list.getData(1,maximum);

int length=list.Length();

for(int i=2;i<=length;i++)

{

// 取链表中的第i个元素Element

list.getData(i,Element);

if(Element>maximum)

maximum=Element;

}

return maximum;

}

（40）已知函数A定义如下，写出它的递归求解算法。

A(0,n)=n+1 　当n≥0时

A(m,0)=A(m-1,1) 　当m>0时

A(m,n)=A(m-1,A(m,n-1))　　　当m>0, n>0时

解：根据函数A定义，写出它的递归求解算法：

double A(double m, double n)

{

if(m==0 && n>=0)

return n+1;

else if(m>0 && n==0)

return A(m-1,1);

else if(m>0 && n>0)

return A(m-1,A(m,n-1));

}

（40）设有关类已定义和实现，并已对其重载流输出运算符，请写出下面程序执行结果。

#include <iostream>

using namespace std;

#include ".\array.cpp"

#include ".\seqstack.cpp"

int \_tmain()

{

int b[9] = {3, -2, 0, -4, 6, -1, 1, 3, 6};

array<int> t(5);

SeqStack<int> sa(10);

unsigned int j=0;

while(j<t.Size())

cout<<t[j++]<<", ";

cout<<endl;

int k=0,Integer;

for(j=1; j<9; j++)

{

if(b[j]>b[j-1])

{

t[k++]=b[j-1]-1;

sa.Push(t[k-1]);

}

}

j=0;

while(j<t.Size())

cout<<t[j++]<<", ";

cout<<endl;

t.reSize(k-1);

j=0;

while(j<t.Size())

cout<<t[j++]<<", ";

cout<<endl;

while(!sa.IsEmpty())

{

sa.Pop(Integer);

cout<<Integer<<", ";

}

cout<<endl;

return 0;

}

解：

//5个随机数

-3，-5，-2，0, 2

-3，-5，-2，0

2, 0, -2, -5, -3

（41）

设以下的类已定义和实现，请写出下面程序执行的结果。

#include <iostream>

using namespace std;

#include ".\linkedqueue.cpp"

int \_tmain()

{

int Integer;

LinkedQueue<int> ql1,ql2;

ql1.MakeEmpty();

ql2.MakeEmpty();

int c[9] = {5, 6, 8, 3, 4, 9, 10, 1, 7};

cout<<"ql1中为：";

while(!ql1.IsEmpty())

{

ql1.DeQueue(Integer);

cout<<Integer<<", ";

}

cout<<endl;

cout<<"ql2中为：";

while(!ql2.IsEmpty())

{

ql2.DeQueue(Integer);

cout<<Integer<<", ";

}

cout<<endl;

cout<<"c[]中为：";

for(int i=0; i<9; i++)

{

cout<<c[i]<< ", ";

if(c[i]!=(c[i]/2)\*2)

ql1.EnQueue(c[i]);

else

{

while(!ql1.IsEmpty())

{

ql1.DeQueue(Integer);

ql2.EnQueue(Integer);

}

ql1.EnQueue(c[i]);

while(!ql2.IsEmpty())

{

ql2.DeQueue(Integer);

ql1.EnQueue(Integer);

}

}

}

cout<<endl;

cout<<"ql1中为：";

while(!ql1.IsEmpty())

{

ql1.DeQueue(Integer);

cout<<Integer<<", ";

}

cout<<endl;

cout<<"ql2中为：";

while(!ql2.IsEmpty())

{

ql2.DeQueue(Integer);

cout<<Integer<<", ";

}

cout<<endl;

return 0;

}

解：

ql1中为：

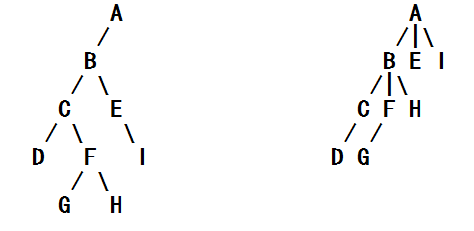
ql2中为：

c[] 中为：5, 6, 8, 3, 4, 9, 10, 1, 7,

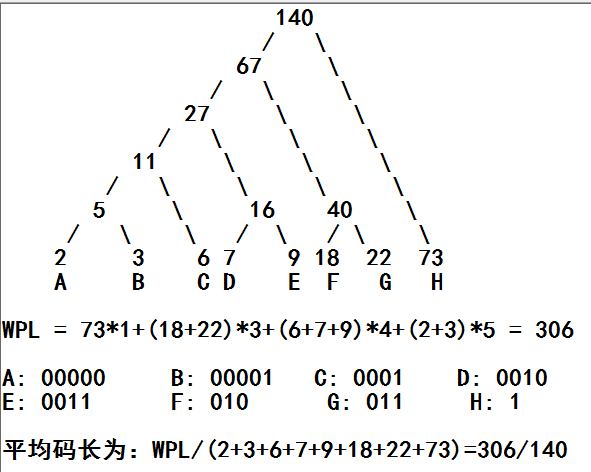
ql1中为：10, 4, 8, 6, 5, 3, 9, 1, 7,

ql2中为：

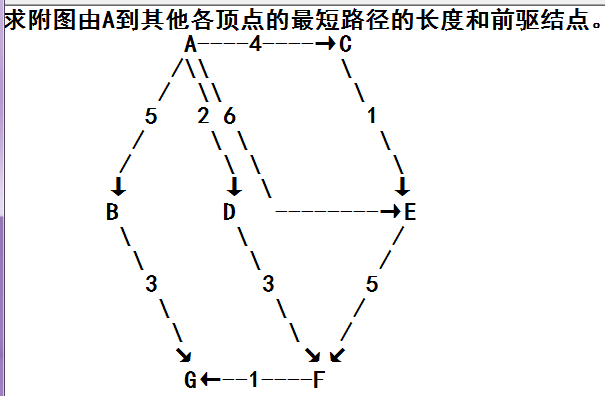
（42）已知一棵树的先根遍历次序是：A,B,C,D,F,G,H,E,I。后根遍历次序D,C,G,F,H,B,E,I,A。请画出这棵树。（提示：先画出这棵树的左子女—右兄弟表示二叉树。）



（43）字符A,B,C,D,E,F,G,H的使用频度依次为2，3，6，7，9，18，22，73，画出相应的哈夫曼树，求出加权路径长度之和WPL，写出各字符的哈夫曼编码，写出平均码长的分数表达式。



（44）



解：

