1.

前序遍历为：18 10 5 73 68 27 25 41 32 51 99

3. 出现的相对频率可以看作是权重，构建霍夫曼树可得

A 0000 B 0001 C 100 D 101

E 001 F 11 G 01

N(a) = 5000 N(b)= 9000 N(c) = 11000 N(d)= 13000

N(e) = 14000 N(f) = 18000 N(g) = 30000

Length = 266000

Avelengrh = 266000/100000 = 2.66

4.

根据题意分析可得

分析过程如下：

前序：G F

中序：F G

根：G 左：F

D

前序：J I

中序：I J

根：J 左：I

前序：C E D G F

中序：C D E F G

根：C 左： 右：D E F G

前序：H C E D G F J I

中序：C D E F G H I J

根：H 左：C D E F G 右：I J

A

前序：E D G F

中序：D E F G

根：E 左：D 右：FG

I

前序：B A H C E D G F J I

中序：A B C D E F G H I J

根：B 左：A 右：C D E F G H I J

5.

class BinNodePtr{

public:

char val; //value

BinNodePtr\* lc; //left child

BinNodePtr\* rc; //right child

}\*root; // root node

BinNodePtr\* elementAtLevel(int theLevel){

queue<BinNodePtr\* >q;

int curlevel = 0;

int cursize = 0;

q.push(root);

while(!q.empty()){

++curlevel;

cursize = q.size();

if(curlevel == theLevel)break;

int temsize = 0;

while(temsize < cursize){

temsize++;

BinNodePtr\* tem = q.front();

q.pop();

if(tem->lc != NULL)q.push(tem->lc);

if(tem->rc != NULL)q.push(tem->rc);

}

}

if(q.empty())return NULL;

else return q.front();

}

2.

It is a max heap.