(第一章)若f(n)=O(nlogn),g(n)=O(2),则f(n)+g(n)=\_\_\_\_\_\_\_\_\_。答案：O(2)

证明等式成立。

证：







又有







故有



（第二章）

1某算法的时间复杂度的表达式为T(n)=an^2+bnlgn+cn+d,其中n为问题规模，a、b、c和d为常数，用O表示其渐进时间复杂度为O(n^2)

2第一个人今年10岁，第二个人比第一个人大2岁，以此类推，给出第n个人年龄的递归算法

int fun(int n){

If(n==1){

return 10;

}else{

Return fun(n-1)+2;

}

}

（第三章）选择排序，插入排序，归并排序算法中，（归并排序）是 分治算法。

1. 数组4,1,6,3,8,2,10，根据最大堆的特点，说明这个数组按递减的顺序进行堆排序的步骤。

解：

10

6

4

8

1

10

2

3

10

2

6

6

2

10

8

6

8

1

3

8

3

1

2

1

3

4

堆的建立已经完成，下面进行堆排序算法

输出10，之后数组变为8,4,6,3,1,2

输出8,10，之后数组变为6,4,2,3,1

输出6,8,10，之后数组变为4,3,2,1

输出4,6,8,10，之后数组变为3,1,2

输出3,4,6,8,10，之后数组变为2,1

输出2,3,4,6,8,10，之后数组变为1

输出1,2,3,4,6,8,10。

（第四章）【解答题】

有如下两个多项式：p(x)=1+x-x^2+2x^3,q(x)=1-x+2x^2-3x^3,用分治法计算这两个多项式的乘积。

解：n=4, n/2=2

划分多项式得：

P(x)=(1+x)+(-1+2x)x^2

q(x)=(1-x)+(2-3x)x^2

p(x)q(x)=r0(x)+(r2(x)-r0(x)-r1(x))x^2+r1(x)x^4

r0(x)=(1+x)(1-x)=1-x^2

r1(x)=(-1+2x)(2-3x)=-2+7x-6x^2

r2(x)=(1+x-1+2x)(1-x+2-3x)=9x-12x^2

p(x)q(x)=1-x^2+(9x-12x^2-1+x^2+2-7x+6x^2)x^2+(-2+7x-6x^2)x^4

=1-x^2+(1+2x-5x^3)x^2-2x^4+7x^5-6x^7

=1+2x^3-7x^4+7x^5-6x^6

【填空题】

实现快速排序算法如下(A)：

Private static void quickSort(int p,int r)

{ if(p<r)

{

Int q=partition(q,r);

(A );

qicksort(q+1,r);

}

}

1. quickSort(p,q-1)
2. quickSort(p+1,q-1)
3. quickSort(p,q+1)
4. quickSort(p,q-2)

递归算法不能适用以下场合（D）

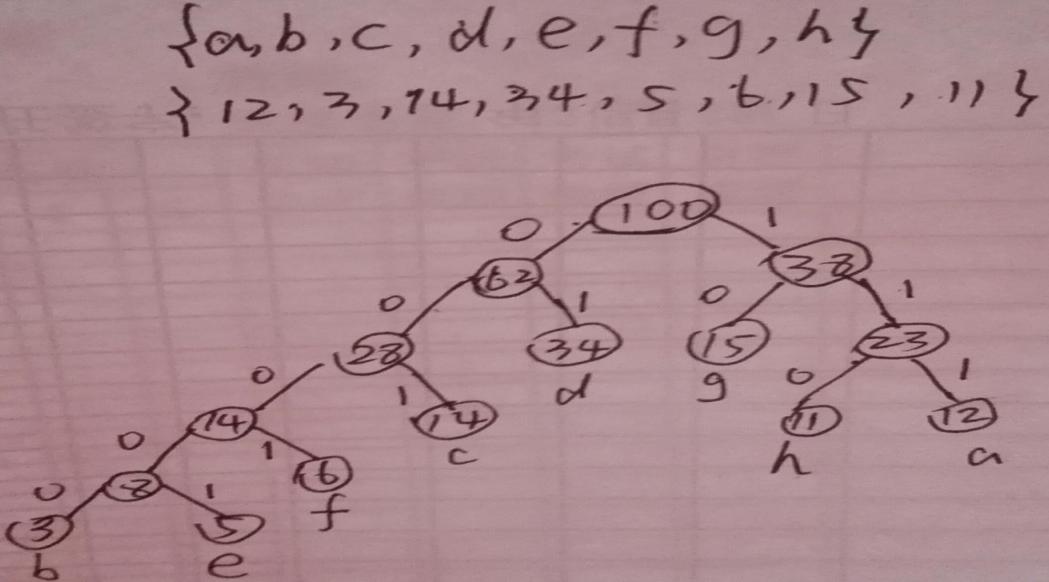
1. 数据的定义形式按递归定义
2. 数据之间的关系（即数据结构）按递归定义
3. 问题解法按递归算法实现
4. 概率问题

（第五章）

1、简答题：

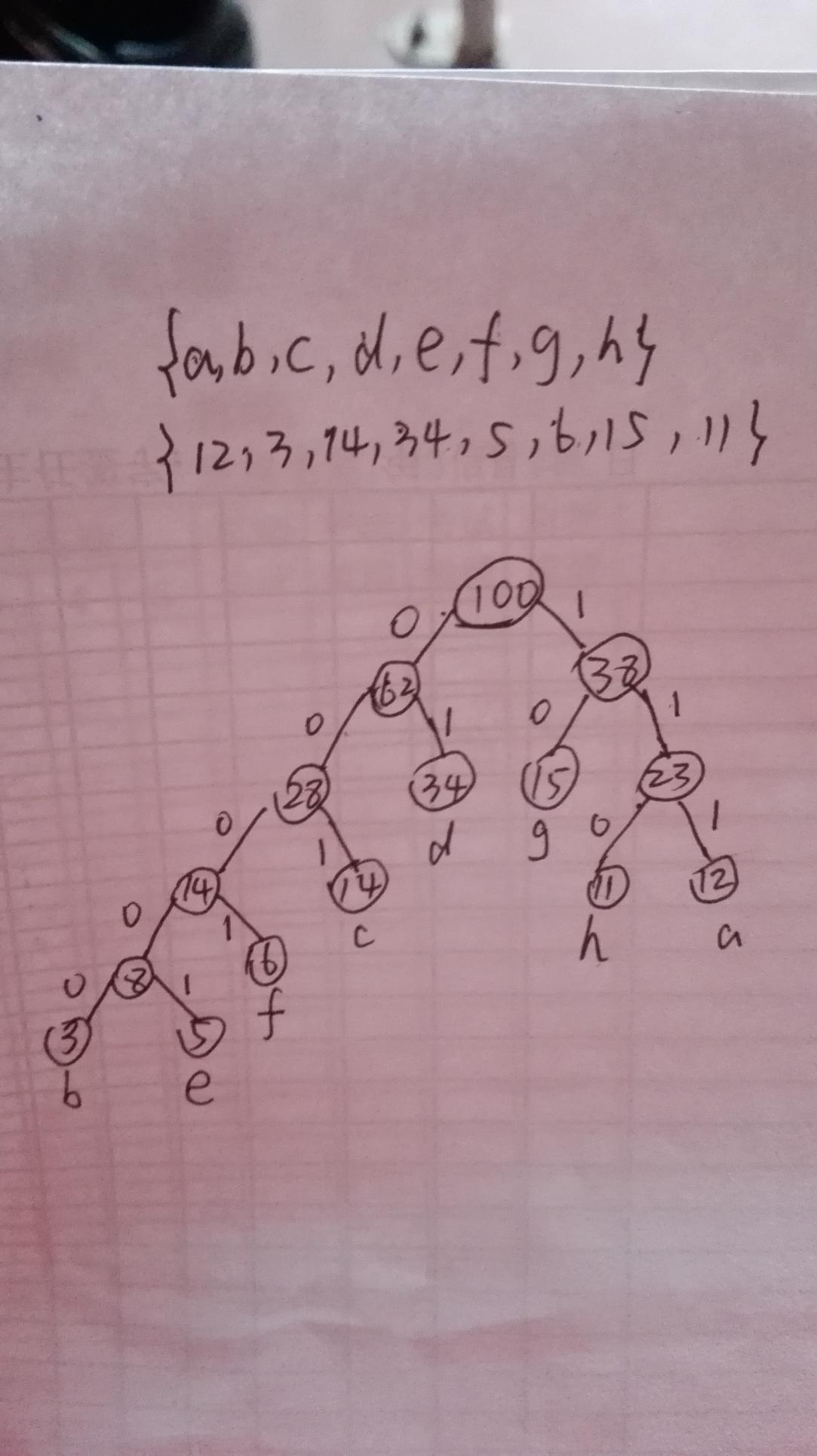
设字符集A={a,b,c,d,e,f,g,h}且各字符在文件中出现频率的百分比分别是12, 3, 14, 34, 5, 6, 15, 11;则字符集的霍夫曼编码？

答案：111,00000,001,01,00001,0001,10,110



2、填空题：

贪婪算法的两个重要性质：贪婪选择性质和最优子结构性质



(第六章)

动态规划算法的基本思想是将待求解问题分成若干\_\_\_\_\_\_\_\_\_\_，先求\_\_\_\_\_\_\_\_\_\_\_\_\_，然后从这些\_\_\_\_\_\_\_\_\_\_\_的解得到原问题的解。

答案：子问题 子问题 子问题

用动态规划方法，求下图从顶点0到顶点6的最短路径。

6

1

4

4 5

2

6

5

3

0

6 8

8 9

5 5 7 9 4

解：求0到6的最短路径是

cost(6)=0;

i=5, cost(5)=C56+cost(6)=4; path(5)=6;

i=4, cost(4)=C46+cost(6)=5; path(4)=6;

i=3, cost(3)=min(C34+cost(4), C35+cost(5), C36+cost(6))=min(8+5,9+4,9)=9; path(3)=6;

i=2, cost(2)=min(C23+cost(3), C25+cost(5))=min(5+9,7+4)=11; path(2)=5;

i=1, cost(1)=min(C13+cost(3), C14+cost(4))=min(6+9,6+5)=11; path(1)=4;

i=0, cost(0)=min(C01+cost(1), C02+cost(2), C03+cost(3))=min(4+11,5+11,8+9)=15; path(0)=1;

route(0)=0;

route(1)=path(route(0))=path(0)=1;

route(2)=path(route(1))=path(1)=4;

route(3)=path(route(2))=path(4)=6;

最短路径为0—1—4—6; cost(0)=15;