**湖南科技大学考试试题卷（A卷）**

**（2015-2016学年度第二学期）**

**课程名称 算法设计与分析 开课单位 数学学院**

**命题人 信计四班第一命题小组**

**授课对象学院 数学学院 年级 2013 班级 信息与计算科学1-4班**

**考试时量100 分钟 审核人 彭叶辉 审核时间 2016年4月22日**

**考核方式（考试）考试方式（闭卷）**

**一、填空题**

1. **一个算法就是一个有穷规则的集合，其中之规则规定了解决某一特殊类型问题的一系列运算，此外，算法还应具有以下五个重要特性:\_\_\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_\_\_。**
2. **若f(n)=nlogn g(n)=logn，用O、Ω和θ表示函数f与g之间的关系 。**

**3. 算法的复杂性有\_\_\_\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_\_之分，衡量一个算法好坏的标准是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。**

**4. 合并排序算法是利用 实现的算法。**

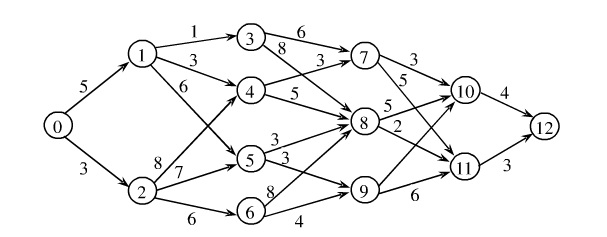
**5. 贪心算法从初始阶段开始，每一个阶段总是做一个使 的贪心选择。**

**6. 在字符集{a,b,c,d,e,f,g,h}中，8个字符在文件中出现频率的百分比分别是3,4,5,8,9,15,20,36，求该字符集的霍夫曼编码 。**

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

**8. 用动态规划方法，求下图从顶点0到顶点12的最短路径为**

**。**



**二、 解答题**

**1．对于下列各组函数f(n)和g(n)，确定f(n)=O(g(n))或f(n)=Ω(g(n))或f(n)=θ(g(n))，并简述理由。**

**(1)**



**(2)**



**(3)**



**2． 编写合并排序程序，将两个已经排序的数组合并成一个数组，其中一个数组能容下两个数组的所有元素;**

**3．设多项式**

**p(x) = 1+ x - x2 + 2x3**

**q(x) = 1- x + 2x2 – 3x3**

**用分治法计算这两个多项式的乘积。**

**4．设计动态规划算法的主要步骤有哪些？**

**5. 用动态规划算法解0-1背包问题:**

**n=5, w=[2,9,4,6,7], p =[6,10,12,8，13], c = 15**。

**答案**

**一、填空**

**1．确定性 有穷性 可行性 0个或多个输入 一个或多个输出**

**2. f(n)=Ω(g(n))**

**3. 时间复杂性 空间复杂性 时间复杂度高低**

**4. 分治策略**

**5. 局部最优**

**6.字符集{a,b,c,d,e,f,g,h}中字符的霍夫曼编码分别为01000,01001,0101,100,101,011,11,00。(答案不唯一)**

**7.子问题 子问题 子问题**

**8. 0→1→3→8→11→12**

**二、 解答题**

**1、解：**

**（1）**

**因为**

**所以**



**（2）**

**因为**

**所以；**



**（3）**

**因为**

**所以；**



**2.**

|  |
| --- |
| **void** MergeArray(**int** a[],**int** alen,**int** b[],**int** blen)  {  **int** len=alen+blen-1;      alen--;      blen--;  **while** (alen>=0 && blen>=0)      {  **if** (a[alen]>b[blen])          {              a[len--]=a[alen--];          }**else**{              a[len--]=b[blen--];          }      }    **while** (alen>=0)      {          a[len--]=a[alen--];      }  **while** (blen>=0)      {          a[len--]=b[blen--];      }  }    **void** MergeArrayTest()  {  **int** a[]={2,4,6,8,10,0,0,0,0,0};  **int** b[]={1,3,5,7,9};      MergeArray(a,5,b,5);  **for** (**int** i=0;i<**sizeof**(a)/**sizeof**(a[0]);i++)      {          cout<<a[i]<<" ";      }  } |

3.

解：n=4,n/2=2,划分多项式得：

p(x) = (1+ x) +(-1 + 2x)x2

q(x) = (1- x) +(2 – 3x)x2



4.

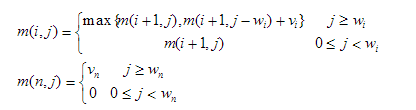
**（1）找出最优解的性质，并刻画其结构特征。**

**（2）递归地定义最优值。**

**（3）以自底向上的方式计算出最优值。**

**（4）根据计算最优值时得到的信息，构造最优解。**

5



M(1,15)=max{m(2,15),m(2,13)+6}={25,25+6}=31

M(2,15)=max{m(3,15),m(3,6)+10}={25,12+10}=25;m(2,13) =max{m(3,13),m(3,4)+10}={25,12+10}=25; m(3,15) =max{m(4,15),m(4,11)+12}={21,13+12}=25; m(3,13) =max{m(4,13),m(4,9)+12}={21,13+12}=25 m(3,6) =max{m(4,6),m(4,2)+12}=12; m(3,4) =max{m(4,4),m(4,0)+12}=12 ；

m(4,15) =max{m(5,15),m(5,9)+8}=max{13,13+8}=21; m(4,13) =

max{m(5,13),m(5,7)+8};={13,13+8}=21 m(4,11) =max{m(5,11),m(5,4)+8}=max{13, 8}=13;

m(4,9) =max{m(5,9),m(5,2)+8}=max{13,0+8}=13 m(4,6) =max{m(5,6),m(5,0)+8}=max{0,0+8}=8

m(4,4) =m(5,4)=0;m(4,2)=m(5,2)=0;m(4,0)=0

最优解为（1,0,1,0,1），最优值为31.