姓名＿＿＿＿ 学号＿＿＿＿ 年级＿＿＿＿ 专业、班＿＿＿＿ 学院＿＿＿＿

公平竞争、诚实守信、严肃考纪、拒绝作弊

密 封 线

重庆大学《算法分析与设计》课程试卷

◉ A卷 ○ B卷

2015— 2016 学年 第 1 学期

开课学院：计算机学院 课程号：**18016435** 考试日期：**2015.1.4**

考试方式：○ 开卷 ◉ 闭卷 ○ 其他 考试时间：**120** 分钟

题 号	一	二	三	四	五	六	七	八	九	十	总 分
得 分											

考试提示

1.严禁随身携带通讯工具等电子设备参加考试；

2.考试作弊，留校察看，毕业当年不授学位；请人代考、替他人考试、两次及以上作弊等，属严重作弊，开除学籍。

一、（15 分）算法复杂度渐进分析

(1) Prove the following two equations (10 分)

$O(f(n))+O(g(n)) = O(f(n)+g(n))$

$f(n) = \Theta(g(n))$ if and only if $g(n) = \Theta(f(n))$

(2) Prove $T(n)=O(n^2)$ through substitution (替代法) (5 分)

$T(n) = 4T(n/3)+n$, $T(1) = \Theta(1)$.

二、 （20 分）合并排序

(1) What is the time complexity (tight bound Θ) to merge (合并) two sorted (已排好序) sub-arrays of equal length n/2？ (2 分)

(2) What is the time complexity (tight bound Θ) to merge two sorted sub-arrays of lengths n/4 and 3n/4 respectively (长度分别为 n/4 和 3n/4)？Describe the reason briefly. (简要说明其理由) (4 分)

(3) Sort an array of length n by Merging Sort (合并排序). If we recursively divide the array according to the ratio 1:3 (按 1:3 的比例递归地划分数组) into two sub-arrays and merge them, then

A) write the recurrence(递推函数) of the time complexity T(n); (3 分)

B) draw out the corresponding recursion tree (递归树); (3 分)

C) prove the tight bound (Θ) of T(n) with substitution. (8 分)

三、 （15 分）动态规划（0-1 背包问题）

Given n objects and a "knapsack". Item i has weighs $w_i > 0$ kilograms and has value $v_i > 0$; Knapsack has capacity of W kilograms. Goal: fill knapsack so as to maximize total value (总价值最大化).

(1) Define OPT(i, W) = max profit (最大价值) for subset of items 1, …, i with weight limit W, complete the following recursive function. (5 分)

$$OPT(i,W) = \begin{cases} 0 & i = 0 \\ & w_i > W \\ & \text{otherwise} \end{cases}$$

(2) Given a knapsack with capacity of 11 kilograms, and 5 objects which been described below. Work out OPT(5,11) with table (用表格计算). (10 分)

Item	Value	Weight
1	1	1
2	18	5
3	6	2
4	20	6
5	28	7

命题人：陈波、张敏 组题人：夏云霓 审题人：李佳 命题时间：2015.12.30 教务处制

四、 （10 分）贪心算法（简化 0-1 背包问题）

Given n objects and a knapsack with capacity W kilograms. Assume all objects have <u>different weights</u> (重量不同) but <u>an identical value</u> (价值一样). Then, the OPT(n, W) can be worked out more easily through <u>greedy choices</u> (贪心选择). Describe the algorithm briefly and give its time complexity (6 分) and <u>prove the first choice satisfying greedy property</u> (证明第一次选择满足贪心原则) . (4 分)

五、 （20 分）动态规划（整数相加合并问题）

正整数的序列：$a_1,a_2,\dots,a_i,\dots,a_n$ ($1\leqslant i\leqslant n$)。相邻两个整数可以相加合并成一个整数，通过有限次合并，最终合并成一个整数。比如1, 2, 3，先合并1和2得到序列3, 3；再合并3和3，最终得到6。但合并整数s和t, 需要付出s+t的代价。在上述1, 2, 3的合并过程中共付出3+6=9的代价，但如果先合并2和3再与1合并的话，需要支出5+6=11的代价。用动态规划设计最佳的合并顺序使总代价最小。

(1) 设F(i,j)等于合并序列$a_i,a_{i+1},\dots,a_j$的总代价的最小值. 给出F(i,j)的递推方程式和边界条件 (5分)，并简单描述算法 (3分)。

(2) 求序列 5,7,10,12,8,9 的最小总代价，用表格记录F(i,j)的值 (8分)。

(4) 分析算法的计算复杂度. (4分)

六、 （20 分）最大流

Let G=(V, E) be a flow network and $|f|$ be the value of a flow f on G, i.e., $|f| = f(s,V)$ with s being source of G.

(1) Prove $f(V, V)=0$. (3 分)

(2) Work out the maximum flow of the following flow network by Edmond-Karp algorithm, where the positive integers denote the <u>capacities</u> (容量) of each edge respectively. During each iteration, you should <u>draw the residue network</u>(画出剩余流量图) and <u>find out an shortest augmenting path</u> (找出最短增广路径) (if exists) using <u>Breadth-First Search</u> (广度优先遍历) . (17 分)