装 订 线

任课教师：

装 订 线

学号：

装 订 线

姓名：

装 订 线

班级：

西 安 电 子 科 技 大 学

考试时间 120 分钟

试　　题

题号	一	二	三	四	五	六	七	总分
分数	16	20	12	12	16	16	8	
得分								

1. 考试形式：闭卷☑ 开卷☐

2. 考试日期：　　年　　月　　日(答案直接答在试卷上，不要超出装订线)

1. Answer T/F for the following: (2 * 8 points)

(1) $8n^2 + 2nlgn + 1000 \in O(n)$

(2) $8n^2 + 2nlgn + 1000 \in \Omega(n)$

(3) $8n^2 + 2nlgn + 1000 \in \Theta(n^2)$

(4) The array A

87 85 83 81 79 77 75 73 71

forms a max-heap.

(5) Radix sort runs correctly when using any correct sorting algorithm to sort each digit..

(6) In Greedy strategy, optimal substructure means that an optimal solution to the problem contains within it an optimal solution to subproblems.

(7) Greedy strategy can be used to obtain an optimal solution of the fractional knapsack problem by choosing the most value/ weight per unit in descending order.

(8) TSP problem is an NP-Complete problem.

2．Single Choice (2*10 points)

(1) The average-case running time of Merge Sort is ( )

A. $\Theta(n^2)$　　B. $\Theta(nlgn)$　　C. $\Theta(n)$　　D. $\Theta(n^3)$

(2) The worst-case running time of Insertion Sort is ( )

A. $\Theta(n^2)$　　B. $\Theta(nlgn)$　　C. $\Theta(n)$　　D. $\Theta(n^3)$

(3) The best-case running time of Quick Sort is ( )

A. $\Theta(n^2)$　　B. $\Theta(nlgn)$　　C. $\Theta(n)$　　D. $\Theta(n^3)$

(4) Which of the following sorting algorithm is NOT stable ( )

A. Heap Sort **B. Merge Sort**

C. Insertion Sort **D. Counting Sort**

(5) Which of the following sorting algorithm is NOT in place ( )

A. Quick Sort **B. Bucket Sort**

C. Heap Sort **D. Insertion Sort**

(6) Which designing strategy is used in Merge Sort ( )

A. Divide and conquer **B. Dynamic programming**

C. Greedy **D. Brute Force**

(7) Which designing strategy is used in 0-1 knapsack problem ( )

A. Divide and conquer **B. Dynamic programming**

C. Greedy **D. Brute Force**

(8) Which designing strategy is used in Dijkstra's algorithm ( )

A. Divide and conquer **B. Dynamic programming**

C. Greedy **D. Brute Force**

(9) In the DP recursive equation used for Longest Common Subsequence problem, *c[i,j]* represents the ( ) of *x[1..i]* and *y[1..j]*, it's the ( ) of the problem.

A. longest common subsequence **B. length of longest common subsequence**

C. optimal solution **D. value of optimal solution**

3. Evaluate the following recursions using the Master Method (3 * 4 points)

(1) $T(n) = 9T(n/3) + n^3$

(2) $T(n) = 9T(n/3) + n^2$

(3) $T(n) = 9T(n/3) + n$

4. Divide and Conquer Strategy (12 points)

(1) Describe the 3 steps used in Divide and Conquer strategy to solve a problem.

(2) Illustrate the operation of merge sort on the array <41, 52, 26, 38, 57, 9, 49>.（给出排序过程）

5. Design Strategies(16 points)

Answer the following questions briefly.

(1) To calculate the optimal solution to matrix-chain multiplication problem and activity-selection problem, which problem can be solved using Greedy strategy? Which problem can be solved using Dynamic Programming strategy? Describe the main idea of the two corresponding algorithms.

(2) Describe a Dynamic Programing strategy to solve an all-pairs shortest paths problem and write down the recursive formula.

6. Maximum Subarray Problem (16 points)

There are 5 items that have a value and weight list below, the knapsack can contain at most 100 kg.

Value ($)	**20**	**30**	**65**	**40**	**60**
Weight(kg)	**10**	**20**	**30**	**40**	**50**

(1) Evaluate the most valuable load for this 0-1 knapsack problem.

(2) Evaluate the most valuable load for this fractional knapsack problem.

7. Algorithm Design(8 points)

(Edit distance) Given two strings, s1 and s2, and edit operations:

Insertion – Insert a new character.

Deletion – Delete a character.

Replace – Replace one character with another.

Design an algorithm to find the minimum number of operations required to convert string s1 into s2.

For example, there are two strings, i.e. s1 = "kitten" and s2 = "sitting", and we perform the following three operations:

1.kitten → sitten (substitution of "s" for "k")

2.sitten → sittin (substitution of "i" for "e")

3.sittin → sitting (insertion of "g" at the end)

So, the minimum number of operations is 3.