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) 5n2 - 2n + 1024 ∈ O(n)

(2) 5n2 - 2n + 1024 ∈ Ω(n)

(3) 5n2 - 2n + 1024 ∈ Θ(n2)

(4) The array A

87 85 72 84 79 75 70 55 68

forms a max-heap.

(5) If a sorting algorithm is NOT stable, then the output of the algorithm may be NOT in correctly sorted order.

(6) In Dynamic Programming 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 0-1 knapsack problem by choosing the most value/ weight per unit in descending order.

(8) 3-COLOR problem is an NP-Complete problem.

2．Single Choice (2*10 points)

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

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

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

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

(3) The worst-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 Quick Sort ( )

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

C. Greedy **D. Brute Force**

(7) Which designing strategy is used in Assembly-Line Scheduling problem ( )

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

C. Greedy **D. Brute Force**

(8) Which designing strategy is used in Activity Selection problem ( )

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) $T(n) = 5T(n/2) + n^2$

(2) $T(n) = 4T(n/2) + n^2$

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

装 订 线

任课教师：

装 订 线

学号：

装 订 线

姓名：

装 订 线

班级：

4. Divide and Conquer Strategy (12 points)

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

(2) Given the following array A to be sorted using Quick Sort as following:

10 9 8 5 4 11 7 6

using the last element (6) as pivot, give the result of the total array after the first partition.

(3) Mark the two sub-problems remained in question (2), describe the following steps in Quick Sort for this instance.

5. Design Strategies(16 points)

Answer the following questions briefly.

(1) To calculate the optimal solution for Fractional Knapsack problem and 0-1 Knapsack problem, which problem can solved using Greedy strategy? Which problem can be solved using Dynamic Programming strategy? Describe the main idea of the two corresponding algorithms.

(2) When solving the Single-Source Shortest Path problem, there may be negative edge(s) or not, in which case Greedy strategy (Dijkstra) can be used to get the optimal solution? Describe the main idea of Dijkstra algorithm. How to calculate the length of the shortest paths in the other case?

6. Maximum Subarray Problem (16 points)

Maximum Subarray Problem means to find the subarray indexed from *i* to *j* which maximize the sum of $A_i \sim A_j$. You should notice that there may be negative ones but not all in the input array A[1..n].

(1) Give an algorithm to calculate the maximum subarray in $O(n^2)$ time, using Brute-Force or Divide and Conquer strategy as you like.

(2) How to solve the problem in O(n) time using Dynamic Programming strategy? Give and describe the recursive equation used in the optimal substructure of the DP algorithm.

(3) Given the following input array as following, used the algorithm in step (2) to calculate the maximum subarray.

8 -5 -4 10 -1 7 -3 12 -20 18

装 订 线 任课教师：_____ 装 订 线 学号：_____ 装 订 线 姓名：_____ 装 订 线 班级：_____

7. Algorithm Design(8 points)

Design an algorithm to find the longest palindrome(回文) subsequence for a given string S. A subsequence is a sequence that can be derived from a string by deleting some chars without changing the order of the remaining ones. A palindrome is a symmetrical string, that is, a string read identically from left to right as well as from right to left.

For example, one longest palindrome subsequence of string "DABBEAF" is "ABBA", whose length is 4.

(NOTICE: You're NOT allowed to use the algorithm for Longest Common Subsequence problem!)