Midterm Exam Name___________ ID___________

It's a closed book and notes exam. You have 120 minutes.

Question	Points	
1	20	
2	20	
3	20	
4	20	
5	20	
6	20	
Total:	100	

1. (20 points)

Explain the following terminologies.

a) Optimal substructure (2 points)

b) Asymptotic bound (2points)

Indicate whether each of the following is true or false.

a) If f(n) = Θ(g(n)), then f(n) = O(g(n)) is also true. (2points)

b) T(n) = 2T(n/2) + nlgn can be solved by the Master Theorem. (2points)

c) Both dynamic programming and greedy algorithm are recursive in nature. (2points)

d) Counting sort algorithm is a comparison based sorting algorithm, and it is stable. (2points)

e) A typical randomized algorithm needs to make assumptions on input distributions. (2points)

Binary search can be viewed as a divide and conquer algorithm. Please describe the tasks for divide, conquer, and combine step, respectively. And give the recurrence for the running time. (6 points)

You can choose four problems from problem 2 to problem 6.

2. Solve $T(n) = 3T(n/2) + cn$, $T(1) = d$ using the recursion tree method. (20 points)

3. Find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is <5, 10, 2, 15, 4>. You need to show your work, including the recurrence relationship of your calculation, and intermediate results of m[i,j]. (20 points)

4. Probability Analysis (20 points)

The following program determines the maximum value in an unordered array A[1..n]. Suppose that all numbers in A are randomly drawn from the interval [0,1]. Let *X* be the number of times line 5 is executed. Show that $\mathrm{E}[X] = \Theta(\ln n)$.

Hint: $\sum_{i=1}^{n} \frac{1}{i} = \ln n + O(1)$

```
1   max ← 0
2   for i ← 1 to n
3       do
4         if A[i] > max
5             then max ← A[i].
```

5. Greedy Algorithm (20 points)

Suppose that instead of always selecting the first activity to finish, we instead select the last activity to start that is compatible with all previously selected activities. Describe how this approach is a greedy algorithm, and prove that it yields an optimal solution.

6. Dynamic Programming (20 points)

Consider the coin changing problem with n denominations D[1..n] and total amount m. Write an $O(nm)$ time dynamic programming algorithm that will always determine the fewest number of coins needed to make exact change for the given amount. As an example, if the denominations are 1, 3, and 4, the total amount m=6, then your algorithm should compute the optimal solution uses 2 coins.

Your answer should include:

1) The recurrence relation and a clear justification for it.
2) Pseudo code for the algorithm.
3) Create the table for the instance mentioned above, and fill in the table.

(blank)