Name:____________________________ **Grade:**______________

Final Exam

CSCE 411 Design and Analysis of Algorithms
Andreas Klappenecker

- This exam contains 9 problems. You have 90 minutes to earn up to 100 points.
- This exam is closed book.
- You are allowed to use a nonprogrammable calculator, although you will not need one.
- Do not spend too much time on a single problem.
- Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Be neat.
- Recall the Aggie code of honor. Cheating will have severe consequences.

The work shown in this exam is my own.

Signature required: __________________________________

Problem 1 (2 Points)

Neatly print your name on each odd-numbered page.

Problem 2 (13 Points)

(a) Which key property is shared by both greedy algorithms and dynamic programming?

(b) In the matrix chain order problem, an array M is maintained to calculate the cost of multiplying the ith through jth matrix, that is, the cost of the matrix product $A_i \cdots A_j$. The array is initialized in the as follows:

	1	2	3	4
1	0			
2	n/a	0		
3	n/a	n/a	0	
4	n/a	n/a	n/a	0

Explain in which order the remaining entries of the table are generated in the algorithm given in the lecture.

(c) Suppose that the matrix A_i has dimensions $p_{i-1} \times p_i$. What is the dimension of the matrix $U = A_i A_{i+1} \cdots A_k$?

(d) Suppose that the matrix A_i has dimensions $p_{i-1} \times p_i$. Suppose that the optimal way to obtain the product $A_i A_{i+1} \cdots A_j$ is by $(A_i A_{i+1} \cdots A_k)(A_{k+1} \cdots A_j)$. Let $m[i, j]$ denotes the number of scalar multiplications to form the product $A_i A_{i+1} \cdots A_j$. Derive a formula that obtains $m[i, j]$ from $m[i, k]$ and $m[k + 1, j]$ assuming the optimal splitting is at k. [Hint: The previous question helps.]

Problem 3 (15 Points)

Suppose that an array $a[0..n]$ contains all numbers in $\{1, 2, \ldots, n\}$, but it contains one number twice. The array is not sorted. The goal is to determine the number occuring twice in $O(n)$ time. The only way to access the array a is by an operation "fetch the ith bit of $a[j]$", which takes contant time.

(a) How many numbers in the set $\{1, 2, \cdots, n\}$ have least significant bit 0, and how many have least significant bit 1? [Hint: For any integer x, we have $x = \lfloor x/2 \rfloor + \lceil x/2 \rceil$]

(b) How can you determine the least signficant bit of the number occuring twice in $a[0..n]$ in linear time (using the "fetch the ith bit of $a[j]$" operation to access the array content)?

(c) (i) Sketch an $O(n)$-time divide-and-conquer algorithm that determines the number which occurs twice in $a[0..n]$. (ii) Explain why your algorithm takes just linear time.

Problem 4 (15 Points)

```
Bellman-Ford(G,w,s)
// Graph G=(V,E), w:V->R a weight function, s is the source vertex
// uses distance estimates d[v] for v in V
// and parents p[v] for v in V with p[s]=s for the source
1  Initialize(G,s)
2  for i = 1 to |V|-1
3   for each edge (u,v) in E do
4      Relax(u,v,w)
5   end
6  end
7  for each edge (u,v) in E do
8   if d[v] > d[u]+w(u,v) then return false
9  end
10 return true
```

1. Give the code of the procedure Relax(u,v,s)

2. Derive the complexity of Bellman-Ford, assuming that Initialize(G,s) takes $\Theta(V)$ time. (Do not just state the result, but explain what each loop contributes. Use line numbers as references)

3. What does the loop in lines 7–9 do? Explain.

Problem 5 (10 Points)

1. The expected running time of Randomized-Quicksort is given by

$$\mathrm{E}[X] = \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1}.$$

   Show that $\mathrm{E}[X] = O(n \log n)$. Explain each step.

2. In the analysis of Randomized-Quicksort, we denoted by z_i the ith smallest element of the input array. Let $Z_{ij} = \{z_i, z_{i+1}, \ldots, z_j\}$. We were interested in the event "the elements z_i and z_j are compared by the algorithm". The first pivot element selected from the set Z_{ij} determines this event. Why?

Problem 6 (15 Points)

Recall that a **vertex cover** of a graph $G = (V, E)$ is a subset V' of V such that for any edge (u, v) in E, it follows that $u \in V'$ or $v \in V'$ (or both) must hold. The VERTEX COVER problem is to decide whether a graph has a vertex cover of size k.

The CLIQUE problem is to decide whether a graph $G = (V, E)$ has a clique of size k, where a clique C is a subset of V such that any two vertices in C are connected by an edge in G.

1. Show that VERTEX COVER is in NP. [Hint: Use a vertex cover V' as a certificate]

2. Given a graph $G = (V, E)$. Explain what the complementary graph $\overline{G}$ of G is.

3. Suppose that a graph $G = (V, E)$ has a clique C of size k. Show that the complementary graph $\overline{G} = (V, \overline{E})$ must contain a vertex cover of size $|V| - k$.

Problem 7 (10 Points)
Computability

1. A hacker claims that she has written a tool that can read a program P and its input I, and determine whether it will take more than k instructions (for a given positive integer k) on a processor to execute the computation of P(I). Is this possible?

2. Show that the set $\mathbf{N}^{\mathbf{N}} = \{f \mid f : \mathbf{N} \to \mathbf{N}\}$ is uncountable. [Hint: Use Cantor's Theorem that $|S| < |P(S)|$ holds for each set S]

Problem 8 (10 Points)
Multithreading

1. Explain work T_1 and span T_∞ of a multithreaded computation.

2. What is the signficance of the ratio T_1/T_∞?

Problem 9 (10 Points)

Minimum spanning trees are used in many applications. Consider a connected graph G with edges that are labeled by positive weights. Show that if all edges have different weights, then the minimum spanning tree of G is unique.

Complete this proof: Seeking a contradiction, we assume that G has two different minimum spanning trees M and M'.

Total points: 100