CSCE 340 STUDY GUIDE 1

DR. GODFREY MUGANDA

1. Important Concepts

1. How do you define a *computational problem*?

2. What is an algorithm?

3. What is the concept of a *program invariant*? Why are program invariants important in the design of a program or algorithm? In the context of program invariants and a piece of code C, what do we mean when we write something like

$$[P] \quad C \quad [Q]$$

and say that this means that P is a *pre-condition* and Q is a *post-condition* for C?

4. What is a *basic step* in the context of algorithm analysis?

5. Define the notions of

(1) *worst case complexity function*
(2) *average case complexity function*

for an algorithm.

6. State the formal definition for the concept of $f(n) \in \mathrm{O}(g(n))$ when f and g are complexity functions. Do the same for $f(n) \in \Omega(g(n))$ and $f(n) \in \Theta(g(n))$.

2. Practice Problems

1. Consider the following computational problem

Input:	An integer array $a[1..n]$ and an integer s.
Output:	The boolean value **true** if s is the sum of two distinct entries of the array, and **false** otherwise.

For example, if the array $a = [20, 5, 12, 5]$ and $s = 10$ then the output should be **true**, but if $s = 20$ or $s = 13$ then the output should be **false**.

(a) Write an algorithm for solving this problem.

(b) Determine the worst case complexity function for your algorithm.

2. Find the closed form sum for the finite series

$$S = \sum_{k=1}^{n} f(k)$$

if

(a) $f(k) = 1$
(b) $f(k) = k$
(c) $f(k) = 2k$
(d) $f(k) = 2k - 1$
(e) $f(k) = \left(\frac{1}{2}\right)^{k-1}$

3. Consider the problem of searching an integer array $a[1..n]$ for an integer X. The probability of finding X at position k, for $1 \leq k \leq n$ is $\frac{1}{n+1}$, and, the probability that X is not found in the array at all is also $\frac{1}{n+1}$.

Consider the obvious algorithm that searches the array from one end to another and stops as soon as X is found, and determine its average case complexity function.

3. More Concepts

1. Express the exponential equation $b^x = y$ in terms of logarithms.

2. Express the logarithmic equation $z = \log_w y$ in terms of bases and exponents.

3. What is $\log_4 16$?

4. What is $\log_4 2$?

5. If I ask you to express 9 as a power of 3, you would write $9 = 3^2$. Express 9 as a power of 7 using only logarithms, and without using a calculator.

4. Solving Recurrence Relations

1. Solve the following recurrence relations and express the answer in closed form.

(a)

$$T(n) = \begin{cases} 2 + T(\frac{n}{3}) & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

(b)

$$T(n) = \begin{cases} 2 + T(n-1) & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

(c)

$$T(n) = \begin{cases} n + T(\frac{n}{2}) & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

(d)

$$T(n) = \begin{cases} n + T(n-1) & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

(e)

$$T(n) = \begin{cases} 3T(\frac{n}{5}) + 1 & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

(f)

$$T(n) = \begin{cases} 5T(\frac{n}{5}) + 1 & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

5. Algorithms

1. Consider a divide and conquer algorithm that works by splitting a problem of size n into 3 sub-problems of the same type, solving two of the sub-problems recursively, and then combining the solutions of the sub-problems to form the solution to the original problem. The algorithm requires constant time to split the problem, and quadratic time to combine the solutions of the sub-problems. Write a recurrence relation for the complexity function of this algorithm.

2. Make sure you understand how to write simple sorting algorithms, Mergesort, and Binary Search.