CS 3520: Algorithms

Homework 1

**Due Date: Monday, February 11, 2019 at beginning of class**

Type your answers in a word processor, print and submit hardcopy in class.

Show your steps to receive partial credit.

1. (20 points) What is an algorithm? What are the steps in algorithm design and analysis?

An algorithm is a process of rules that must be followed to solve a problem.

* Understand the problem
* Decide on method of solving (exact vs approximate)
* Design it
* Prove correctness
* Analyze it
* Code it

1. (20 points) For each of the following functions, indicate the class Θ (*g*(*n*)) the function belongs to. Use the simplest *g*(*n*) possible in your answers.

a.

Θ(4n)

b.

Θ(nln5(n))

c.

Θ(n5lgn)

d.+

Θ(4n)

1. (20 points) Consider the following algorithm where the input A[0..*n* − 1]) is an array of *n* integers.

**ALGORITHM** ALG1(A[0..*n* − 1])

**for** *i* ←0 **to** *n − 2* **do**

**for** *j* ←0 **to** *n − 2 − i* **do**

**if** A[*j* + 1]<A[*j*] swap A[*j*] and A[*j*+1]

a. What does this algorithm do?

Sorts an array from least to greatest.

b. What are the basic operations of this algorithm?

Less than comparison

c. How many times are the basic operations executed?

(n-2)\*(n-1)

2

d. What is the efficiency class of this algorithm? Express your answer with the most appropriate O, Θ or Ω notation.

O(n2)

1. (20 points) Solve the following recurrence relations and find an explicit formula for .

a. for

3n-1

b. for

5(n-1)

c. for

(1/3) \* (4n-1)

d. for

n2

1. (20 points) Consider the following recursive algorithm where the input A[0..*n* − 1]) is an array of *n* real numbers.

**ALGORITHM** *R*(*A[0..n − 1]*)

**if** *n = 1* **return** *A[0]*

**else** *temp←R(A[0..n − 2])*

**if** *temp* *≥* *A[n − 1]* **return** *temp*

**else** **return** *A[n − 1]*

a. What does this algorithm compute?

Gets maximum of an array

b. What is its basic operation of this algorithm?

Greater than or equal to comparison

c. Set up a recurrence relation for the algorithm’s basic operation count and solve it.

R(1) = A[0] //Initial condition

R(n) = R(n-1) //Recurrence relation