MAHARISHI UNIVERSITY OF MANAGEMENT
COMPUTER SCIENCE DEPARTMENT
CS 435: ALGORITHMS

MIDTERM EXAM – NOV-2018

Student Name: _____ **Student Id:** _____

Instructor: Sami Taha

Question #	1 (12)	2 (16)	3 (12)	4 (5)	Total Grade
Grade (45)					

Instructions

1. Allotted exam duration is 2 hours.
2. Closed book/notes.
3. No personal items including electronic devices (cell phones, computers, calculators, PDAs).
4. Cell phones must be turned in to your proctor before beginning exam.
5. No additional papers are allowed. Sufficient blank paper is included in the exam packet.
6. Exams are copyrighted and may not be copied or transferred.
7. Restroom and other personal breaks are not permitted.
8. Total exam including questions and scratch paper must be returned to the proctor.

Information you might need.

1. Master Formula

Suppose T(n) satisfies

$$T(n)=\begin{cases} d & if\ n=1 \\ aT(\left\lceil \frac{n}{b} \right\rceil)+cn^k & otherwise \end{cases}$$

Where k is none negative integer and a, b, c, d are constants with a>0, b>1, c>0, d>=0 then

$$T(n)=\begin{cases} \Theta(n^k) & if\ a<b^k \\ \Theta(n^k \log n) & if\ a=b^k \\ \Theta(n^{\log_b^a}) & if\ a>b^k \end{cases}$$

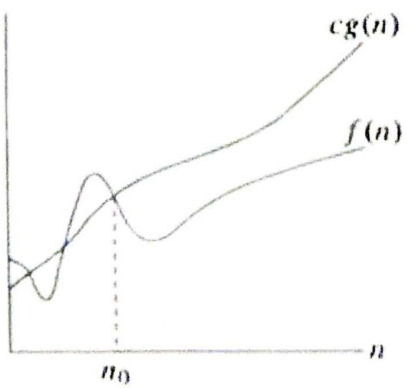
2. $x=b^y \Longrightarrow Log_b x=y$

3. $\sum_{i=0}^{n-1} i=\frac{n(n-1)}{2}$

Q1) (12 points) Complexity Analysis

1. **[12 points] Multiple Choice Question:**

1.	The worst case running time in Big-Oh notation for the following function is :
	`void foo( )`
	`{`
	`  int count=0;`
	`  for(int i=0; i<1000; ++i)`
	`  {`
	`  count += i;`
	`  for(j=0; j<n; ++j)`
	`  {`
	`  System.out.println(j);`
	`  }`
	`  for(k=0; k<i; ++k)`
	`  {`
	`  System.out.println(k);`
	`  }`
	`  }`
	`}`
		a. O(n)
		b. $O(n^2)$ <--This
		c. O(1)
		d. None of the above

2.	What is the tightest asymptotic bound for the following f(n) $f(n) = 2^{100} n^4 + 2^n$	a. $O(n^4)$ b. $O(2^{100})$ c. $O(2^n)$ <---This d. $O(2^{100} n^4)$
3.	What is the tightest asymptotic bound for the following f(n) $f(n) = n^2 \log(n^{10}) + 5n \log(n^5) + n^3$	a. $O(n^2 \log(n^{10}))$ b. $O(n^2 \log n)$ c. $O(n^3)$ <-- This d. $O(5n \log(n^5))$
4.	For the following graph, the asymptotic relationship of f and g functions is  	a. f(n) = Theta (g(n)) b. f(n) = O (g(n)) <--This c. f(n) = Big-Omega (g(n))

5	Given the function $f(n) = 2n + \log n$ and the two statements I. $f(n) = \Theta(n)$ II. $f(n) = \Theta(\log n)$	a. Only statement I is true <--This b. Only statement II is true c. Statement I and II are true d. Both statements are false
6	The order of 2n is	a. O(n) <-- This b. $O(n^2)$ c. $\Theta(n^2)$ d. A and B e. B and C

Lim 2n+log n/n
Lim 2+log n/n=0

Lim n/(2n+log n)=0
Si theta

Lim 2n+log n/log n
2n/log n + 1=infinite
no theta

Q2) (16 points) Analysis

1. [4 points] Give tight asymptotic bound for the following recurrences.
   a. $T(n)=3T(n/9)+2\ n$

a=3
b=9
c=2
k=1

3 < 9^1 Yes => n

   b. $T(n) = 16T(n/4) + 100\ n^2$

a=16
b=4
c=100
k=2

16 < 4^2, no
a = 16, yes => n^2*log n

2. [2 points] Discuss whether the statement "Amortization makes an algorithm more efficient" is true or false. Justify your answer.

No. Amortization is a mechanism to calculate the average cost of single operation in a sequence of n operations, with the formula: O(n)/n

3. [3 points] An algorithm solves a problem by dividing it into 4 sub-problems. Each recursive call divides the problem into one-third of the size of the problem. After solving all sub-problems it combines the solutions in quadratic time.

   a. Give a recurrence formula for this algorithm.

T(n) = 4(T(n/3)) + n^4

   b. Give the tightest asymptotic bound for the proposed recurrence in part a.

a=4,
b=3,
c=1,
k=4

4 < 3^4 => n^4

4. [2 points] Express the complexity of the following algorithm as a recurrence relation procedure

```
void foo(int n)
{
  if(n <=0) return;
  System.out.println(n);
  foo(n/3)+foo(n/2);
}
```

+1
+1
2 + T(n/3) + T(n/2)

T(n) = T(n/3) + T(n/2) + 4

T(4) = 2 + 3 = 5

5. [5 points] Given an array A with n integers and there may be duplicate elements. We want to find which element has the most duplicates in A. Propose an algorithm to solve this problem in O(n) worst-case time or better. Example: [1 2 2 3 1 4 2 2 3] the entry with most duplicate is 2

Q3) (12 points) Sorting algorithms:

1. [5 points] Answer the following questions by True or False. Fill your answers in the following table.
   a. For the binary search algorithm, the successful search requires only O(log n) comparisons, but the unsuccessful search requires n comparisons.
   b. The best time complexity of Bubble Sort is O(n log n).
   c. Selection-Sort makes less swap operations than Merge Sort.
   d. Any comparison based sorting algorithm can be made stable by using position as a criteria when two elements are compared .
   e. Bubble-Sort is stable sorting algorithm.

A	B	C	D	E
T	F	F	T	T

2. [7 points] Considering Quick Sort Algorithm.
   a. [1 point] What is the worst running time for Quick Sort?

      O(n*log n)

   b. [2 points] What can you do to avoid the worst case running time?

      We have to guarantee pickPivot always choose a good pivot.
      Ex: choose the median as pivot not random nor always leftmost or rightmost

c. [4 point] Sort the following array using Quick-Sort Algorithm. Assume that the first element is the pivot in every iteration. Show all your work in each step. You do not need to write the algorithm

48	33	31	60	1	49	30	70

Q4) (5 points) SCI Question:

Write a short essay that connects one of the studied topics to the Science of Creative Intelligence (SCI). You can pick any topic.