CSE 3131: ALGORITHM DESIGN 1

ASSIGNMENT 1:

Submission due date: 30/10/2023

- *Assignment scores/markings depend on neatness and clarity.*
- *Write your answers with enough detail about your approach and concepts used, so that the grader will be able to understand it easily. You should ALWAYS prove the correctness of your algorithms either directly or by referring to a proof in the book.*
- *The marking would be out of 100.*
- *You are allowed to use only those concepts, which are covered in the lecture class till date.*
- *Plagiarized assignments will be given a zero mark.*

---

*CO1: to apply **knowledge of computing and mathematics** to algorithm design;*

(i) to understand computational tractability considering polynomial time as a definition of efficiency of an algorithm;

*(ii) to analyze worst-case **running times of algorithms (both recursive and iterative)** using asymptotic analysis;*

Algorithm Basics:
Asymptotic Notations:
Time and Space Complexity:
Solving Recurrences:

Sl. No.	***Questions***	***PO***	***Level***
1.	**Draw the flowchart for the following algorithms.**	PO1	L2, L3, L4
	a. To find the sum of the following series with n numbers starting from 1: $sum = 1 - 2 + 3 - 4 \pm \cdots \pm n$. b. To print the Fibonacci Series up to nth term. c. To check if a given number is prime or not. d. To count the number of digits in an integer. e. To find the reverse of a given number.		
2.	**Write the pseudocode for the following algorithms:**	PO1	L2, L3, L4
	a. To check if a given number is Palindrome or not. b. To find the sum of the digits of a number. c. Iterative Binary Search on an array of n numbers and a given element. d. To convert a given decimal number to the corresponding binary number. e. To check if a number can be represented as the sum of two prime numbers or not.		
	Asymptotic Notations:		
3.	**For each of the following pairs of functions, determine whether $\boldsymbol{f(n) = O(g(n))}$ or $\boldsymbol{g(n) = O(f(n))}$.** a. $f(n) = n(n-1)/2$ and $g(n) = 6n$	PO1	L1, L2, L3

	b. $f(n) = n + 2\sqrt{n}$ and $g(n) = n^2$ c. $f(n) = n + \log n$ and $g(n) = n\sqrt{n}$ d. $f(n) = n \log n$ and $g(n) = n\sqrt{n}/2$ e. $f(n) = 2(\log n)^2$ and $g(n) = \log n + 1$		
4.	**State TRUE or FALSE justifying your answer with proper reason.** a. $2n^2 + 1 = O(n^2)$ b. $n^2(1+\sqrt{n}) = O(n^2)$ c. $n^2(1+\sqrt{n}) = O(n^2 \log n)$ d. $3n^2 + \sqrt{n} = O(n + n\sqrt{n} + \sqrt{n})$ e. $\sqrt{n} \log n = O(n)$ f. $lg\ n \in O(n)$ g. $n \in O(n\ lg\ n)$ h. $n\ lg\ n \in O(n^2)$ i. $2^n \in \Omega(6^{\ln n})$ j. $lg^3 n \in o(n^{0.5})$	PO1	L1, L2, L3
5.	**For each of the following pair of functions $\boldsymbol{f(n)}$ and $\boldsymbol{g(n)}$, determine whether $\boldsymbol{f(n) = O(g(n))}$ or $\boldsymbol{f(n) = \Theta(g(n))}$ or $\boldsymbol{f(n) = \Omega(g(n))}$.**	PO1	L1, L2, L3
	a. $f(n) = \sqrt{n}, g(n) = \log(n+3)$ b. $f(n) = n\sqrt{n}, g(n) = n^2 - n$ c. $f(n) = 2^n - n^2, g(n) = n^4 + n^2$ d. $f(n) = n^2 + 3n + 4, g(n) = 6n^2$ e. $f(n) = n + n\sqrt{n}, g(n) = 4n \log(n^2 + 1)$		
6.	**Prove the following statements:**	PO1	L1, L2, L3
	a. If $f_1(n) = \Omega(g_1(n))$ and $f_2(n) = \Omega(g_2(n))$ then $f_1(n) + f_2(n) = \Omega(g_1(n) + g_2(n))$		L1, L2, L3
	b. If $f_1(n) = \Omega(g_1(n))$ and $f_2(n) = \Omega(g_2(n))$ then $f_1(n) + f_2(n) = \Omega(\min(g_1(n) + g_2(n)))$		L1, L2, L3
	c. If $f_1(n) = O(g_1(n))$ and $f_2(n) = O(g_2(n))$ then $f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$		L1, L2, L3
	d. If $f(n) = O(g(n))$, then $f(n)^k = O(g(n)^k)$.		L1, L2, L3
	e. If $f_1(n) = \theta(g_1(n)), f_2(n) = \theta(g_2(n)), \ldots, f_k(n) = \theta(g_k(n))$ where k $\in I^+$ then prove that $\sum_{i=1}^{k} f_i(n) = \theta\left(\max_{1 \le j \le k}(g_j(n))\right)$.		L1, L2, L3

7.	Given $f(n) = (n + a)^b$ for any real constants a and b, where $b > 0$. Prove that $f(n) \in \theta(n^b)$.	PO1	L1, L2, L3
8.	Prove that $lg\ n = O(\sqrt{n})$, however $\sqrt{n} \neq O(lg\ n)$.	PO1	L1, L2, L3
	Time and Space Complexity:		
9.	Let *A* [1 ... 60] = {10, 11, ... , 70}. How many comparisons are performed by algorithm BinarySearch(A) when searching for the elements 10, 40, 50, 70 respectively? Explain each cases with detailed execution. Draw the recursion tree for each of the cases and compare the stack space used.	PO1, PO2	L1, L2, L3
10.	Suppose you are given with a problem P with input size *n* and three algorithms A, B and C to choose from, in order to solve P. The algorithms operate as follows: • Algorithm *A* solves P by dividing it into five sub-problems of half the size, recursively solving each sub-problem and then combining the solution in linear time. • Algorithm *B* solves P by recursively solving two sub-problems of size *n*-1 and then combining the solutions in constant time. • Algorithm *C* solves P by dividing it into nine sub-problems each of size *n*/3, recursively solving each sub-problem and then combining the solutions in O(n^2) times. Compare the running times of these algorithm in big-O notation. Which one would you choose?	PO1, PO2	L1, L2, L3

11.	**function(int *n*)**
	{ if(*n* = =1)
	    return 1;
	  else
	    function(*n*/3); function(*n*/3); function(*n*/3);
	    for(*i* = 1; *i* <= *n*; *i*++)
	      x = *x* + 1;
	}
	Find the time and space complexity of the given algorithm.
		PO1	L1, L2, L3

12.	**void function(int *n*){**
	    Temp = 1;
	    Repeat
	      for *i* =1 to n
	        temp = *temp* + 1
	        n=*n*/2;
	    Until *n* <= 1
	    }
	Find the time and space complexity of the given algorithm.
		PO1	L1, L2, L3

13.	**int function(int *n*){**
	if(n <= 2)
	return 1;
	else
	return(function(floor(sqrt(n))) + 1);
	}
	Find the time and space complexity of the given algorithm.
		PO1	L1, L2, L3

14.	**Void function(int n)** {
	if(n = = 1)
	return 1;
	else
	for(i = 1; i <= 8; i++)
	function(n/2);
	for(i = 1; i<= n^3; i ++)
	count=count+1;
	}
	Find the time and space complexity of the given algorithm.
		PO1	L1, L2, L3

15.	The following pseudocode performs linear search on an array of size n to find the presence of an element *el*. LinearSearch(A, n, el) 1. for i = 1 to n do 2. If $A[i]$ = *el* then 3. return i 4. return *NIL* Write the recursive version of the LinearSearch algorithm, formulate the recurrence relation for its time complexity function and compare the time and space complexities with the iterative version.	PO1	L1, L2, L3
	Solving Recurrences:		
16.	**Solve the following recurrence using any of the suitable methods. If no solution is possible, justify using proper reasoning.** a. $T(n) = \begin{cases} 1 & if\ n = 2 \\ 1 & if\ n = 4 \\ T\left(\frac{n}{2}\right) + 2T\left(\frac{n}{4}\right) + \theta(n^2) & if\ n > 4 \end{cases}$ Where n is assumed to be a power of 2	PO1	L1, L2, L3

b. $T(n) = n^{\frac{1}{3}} T\left(n^{\frac{2}{3}}\right) + \theta(n)$

c. $T(n) = \sqrt{n}\, T(\sqrt{n}) + \log n$

d. $T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{4n}{5}\right) + \theta(n)$

e. $T(n) = 3T\left({}^{n}/_{4}\right) + cn^2$

f. $T(n) = \begin{cases} 8 & for\ n = 1 \\ 3T(n-1) - 15 & for\ n > 1 \end{cases}$

g. $T(n) = \begin{cases} 5 & for\ n = 1 \\ 2T(n-1) + 3n + 1 & for\ n > 1 \end{cases}$

h. $T(n) = \frac{n}{n-5}\, T(n-1) + 1$

i. $T(n) = T(\log n) + \log n$

j. $T(n) = T\left(n^{\frac{1}{4}}\right) + 1$

k. $T(n) = n + 7\sqrt{n} \cdot T(\sqrt{n})$

l. $T(n) = T\left(\frac{3n}{4}\right) + \frac{1}{\sqrt{n}}$

m. $T(n) = \begin{cases} 1 & for\ n = 1 \\ 3T\left(\frac{n}{3} + 5\right) + \frac{n}{2} & for\ n > 1 \end{cases}$

n. $T(n) = \begin{cases} 1 & for\ n = 1 \\ 2T\left(\frac{n}{2}\right) + \frac{n}{\log\log n} & for\ n > 1 \end{cases}$

o. $(n) = \begin{cases} 1 & for\ n = 0 \\ T(n-2) + 2\log n & for\ n > 0 \end{cases}$

p. $T(n) = 2T\left(\frac{n}{2}\right) + n^2(1 + \sin n)$

Submission and Grading:

Submit the hard copy of your assignment by the due date, i.e. 30.10.2023.

Part of your assignment grade comes from its "external correctness." This is based on correct output on various sample inputs.

The rest of your assignment's score comes from "internal correctness." Internal correctness includes:
1. Use of methods to minimize the number of steps.
2. Appropriate use of rules, axioms, and suitable diagrams to enhance readability of your responses.

Send a zip folder (name of the zip folder must be your registration number_AD1) containing the code and output file/screen-shot of each program implementation mentioned to the official email id of your AD1 class teacher. On the top of each program, you must mention your full name, registration number, title of the program and date.