Missouri University of Science & Technology	Department of Computer Science
Spring 2024	**CS 2500: Algorithms (Sec: 102)**

Homework 1: Foundations

Instructor: *Sid Nadendla* **Due:** *February 9, 2024*

Problem 1 Karatsuba's Integer Multiplication *2 points*

Problem 1.8 (Ref. Page 26 in the textbook.)

Note: The problem is labeled as Problem 1.6 in the 2017 edition of the same book, as well as on the book's website under Test Cases and Data Sets for Programming Projects.

Statement:

Implement Karatsuba's integer multiplication algorithm in Python. To get the most out of this problem, your program should invoke the language's multiplication operator only on pairs of single-digit numbers.

Test your code by computing the product of the following two 64-digit numbers:

`3141592653589793238462643383279502884197169399375105820974944592`

`2718281828459045235360287471352662497757247093699959574966967627`

Problem 2 Empirical Run-Time Analysis *2 points*

Perform empirical run-time analysis on Karatsuba's algorithm and compare your findings with the theoretical runtime analysis carried out in the class.

Ref. Topic 1.2 (Empirical run-time analysis) under `T1 - Fundamentals` tab in the class notes.

Problem 3 Asymptotic Notation *2 points*

Problem 2.4 (Ref. Page 43 in the textbook.)

Statement:

Arrange the following functions in order of increasing growth rate, with $g(n)$ following $f(n)$ in your list if and only if $f(n) = O\Big(g(n)\Big)$.

(a) $n^2 \log_2 n$ (b) 2^n (c) 2^{2^n} (d) $n^{\log_2 n}$ (e) n^2

Problem 4 Divide and Conquer *2 points*

Problem 3.4 (Ref. Page 69 in the textbook).

Statement:

You are given a sorted (from smallest to largest) array A of n distinct integers which can be positive, negative, or zero. Design the fastest algorithm you can, for deciding whether there is an index i such that $A[i] = i$.

Problem 5 Master Method *2 points*

Problem 4.3 (Ref. Page 88 in the textbook).

Statement:

This question will give you practice with the Master method. Suppose the running time $T(n)$ of an algorithm is bounded by a smallest recurrence with $T(n) \leq 7 \cdot T(\frac{n}{3}) + O(n^2)$. Which of the following is the smallest correct upper bound on the asymptotic running time of the algorithm?

(a) $O(n \log n)$ (b) $O(n^2)$ (c) $O(n^2 \log n)$ (d) $O(n^{2.81})$

Problem 6 Extra credit *1 point*

Problem 4.7 (Ref. Page 89 in the textbook.)

Statement:

Suppose the running time $T(n)$ of an algorithm is bounded by the (non-standard!) recurrence with $T(1) = 1$ and $T(n) \leq T(\lfloor \sqrt{n} \rfloor) + 1$ for $n > 1$. Which of the following is the smallest correct upper bound on the asymptotic running time of the algorithm? (Note that the Master method does not apply! Instead, trying using recursion trees.)

(a) $O(1)$ (b) $O(\log \log n)$ (c) $O(\log n)$ (d) $O(\sqrt{n})$