National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Design and Analysis of Algorithms	Course Code:	CS2009
Degree Programs:	BSCS,BSSE	Semester:	Spring 2023
Due Date:	NA	Total Marks:	NA
Section:	ALL	Page(s):	2
Instrument Type:	Homework		

Instructions: This homework is for your learning; you are NOT supposed to submit it.

Q1) What is the runtime of the following function? Express your answer using the big-*O* notation. Show all working.

```
mystery (n)
     if (n > 1)
          print "hello"
          mystery(2n/3)
          for i=1 to n
                print "world"
          mystery(n/5)
```

Q2) Prove that $T(n) = \Theta(n^4)$ by finding appropriate constants (i.e. $c_1, c_2,$ and n_0,).

$$T(n) = \frac{1}{16}n^4 - 4n^2$$

Q3) Suppose we have a sorting algorithm with the following procedure:

If the sequence length is at most 4, then sort it using bubble sort.

Else:

i. Divide the list into 5 pieces evenly, by scanning the entire list.

ii. (recursively) sort the first 3/5 of the list.

iii. (recursively) sort the last 3/5 of the list.

iv. (recursively) sort the first 3/5 of the list.

For example, on the input sequence 1, 5, 3, 2, 4 The first recursive sort produces 1, 3, 5, 2, 4, the second sort produces 1, 3, 2, 4, 5, and the last produces 1, 2, 3, 4, 5.

(a) Write down a runtime recurrence for this sorting algorithm and analyze its asymptotic running time.

(b) Prove that this algorithm is **not correct** by giving an example sequence (of 5 to 10 elements) where this sorting algorithm does not terminate with the correct answer.

Q4) Use recursion tree method to guess a **good** asymptotic upper bound on following recurrences.

a) $T(n) = 2T(n/4) + \sqrt{n}$

b) $T(n) = T(n - 2) + n^2$

c) $T(n) = 10T(n/2) + \Theta(1)$

e) $T(n) = 2T(n - 1) + \Theta(1)$

f) $T(n) = T(n - 1) + 1/n$

g) $T(n) = 4T(n/2) + n^2\sqrt{n}$

See the Appendix of the textbook for material on harmonic and geometric series.