Homework Assignment #3

Due: Wednesday, Feb. 9, 2022 (11:59pm)

(Total 40 marks)

CMPUT 204
Department of Computing Science
University of Alberta

In all problems, we assume $T(n) = O(1)$ for small n.

Problem 1. (20 marks)

- Use iterated substitution to guess a good solution (a tight asymptotic upper bound) to the recurrence $T(n) = T(n-10)+n$ and prove your guess is indeed a solution. What about $T(n) = T(n-100)+100n$? No justification is required (hope you can answer this question immediately with confidence).
- Show the solution to $T(n) = 2T(\lfloor n/2 \rfloor + 8) + n$ is $O(n \log n)$.
- Given the recurrence $T(n) = T(n/3) + T(2n/3) + 3n$, use the recurrence tree method to guess a tight upper bound and a tight lower bound and prove that your guesses are correct (to simplify the question, for this latter part, you only need to present a proof either for the upper bound or for the lower bound).
- Solve the recurrence $T(n) = 2T(\sqrt{n}) + \log \log n$. Hint: You may consult any solutions you can find online for CLRS 4.3-9.
- Solve the recurrence $T(n) = 2T(n-2) + 1$. You can use any method to guess a tight solution (you only need to sketch how you guessed) and then prove your guess is correct.

Problem 2. (8 marks) Apply the master method to solve the following recurrences.

a) $T(n) = 8T(n/4) + n^2$.

b) $T(n) = 5T(n/9) + \sqrt{n}$.

c) $T(n) = 2T(n/4) + 1$.

d) $T(n) = 9T(n/8) + n^2 + n$.

Problem 3. (12 marks) Consider the following very simple and elegant(?) sorting algorithm:

SomeSort (A, b, e)
if $e = b + 1$ **then**
    if $A[b] > A[e]$ **then**
        exchange $A[b]$ and $A[e]$
    end if
else if $e > b + 1$ **then**
    $p \longleftarrow \lfloor \frac{e-b+1}{3} \rfloor$
    SomeSort $(A, b, e - p)$
    SomeSort $(A, b + p, e)$
    SomeSort $(A, b, e - p)$
end if

a. Does SomeSort correctly sort its input array A (assuming that $n = e - b + 1$ is the length of the array)? Justify your answer.

b. Consider the input array A: 8 1 4 9 7 3 2 6 5. List five valid states of the array during the execution of the algorithm.

c. Find a recurrence for the worst-case running time of SomeSort. Give a tight (i.e. Θ) asymptotic bound for the worse-case running time of SomeSort (Hint: consider $n = 3^k$ for some k).

d. By comparing SomeSort with insertion sort and merge sort, argue if this simple algorithm is efficient.