CMPUT 204 — Problem Set 2

(Partially provided by *Yang Wang*)

Topics covered in Part I are recurrence relation deriving and solving by iterative substitution method (which can then be proven by mathematical induction); in Part II are recursion-tree method, the master theorem, and the introduction to heaps. Using the master theorem is required while its proof (by iterative substitution) is not required.

It is highly recommended that you read pages **62–75**, **85–90**, and **127–135** very carefully and do **all** the exercises. The following are some of them that you are **REQUIRED** to practice on.

Quiz questions are mostly based on this list, with some minor modifications necessary. Consult your instructor and TAs if you have any problem with this list.

Part I

1. Assuming the following uniform RAM operation times, for each of the pseudocode fragments, give a recurrence relation for the running time, in number of cycles.

   - arithmetic, assignment — 1 cycle
   - return, branch test — 1 cycle
   - procedure call — 1 cycle/parameter

   Procedure QZ(n)
   ```
   if n > 1 then
       a ← n × n + 97
       b ← a × QZ(n/2)
       return QZ(n/2) × QZ(n/2) + n
   else
       return n × n
   ```

   Procedure Z(n)
   ```
   if n > 1 then
       a ← n + Z(n/3)
       b ← a + Z(n/4) + Z(n/3)
       return b
   else
       return 1
   ```

   Hints:

   For procedure QZ(n),

   - Line 1 contains 1 comparison and 1 branch test;
   - Line 2 contains 1 multiplication, 1 addition, and 1 assignment;
   - Line 3 contains 1 division, 1 procedure call, 1 multiplication, and 1 assignment;
   - Line 4 contains 2 divisions, 2 procedure calls, 1 multiplication, 1 addition, and 1 return;
   - Line 6 contains 1 multiplication and 1 return.

   Therefore, the recurrence relation for $T(n)$ is

   $$T(n) = \begin{cases} 4, & \text{if } n = 1, \\ 3T(\frac{n}{2}) + 16, & \text{if } n > 1. \end{cases}$$

   For procedure Z(n),

   $$T(n) = \begin{cases} 3, & \text{if } n = 1, \\ 2T(\frac{n}{3}) + T(\frac{n}{4}) + 14, & \text{if } n > 1. \end{cases}$$

2. Solve the following recurrence relations. You may assume $T(1) = 1$, the recurrence is for $n > 1$, and c is some positive constant. (The first step is to derive a closed form for the answer, using repeated substitutions; The second step is to prove the closed from is correct, by induction.)

   (a) $T(n) = 2T(n/3) + 5$.
   Hint:
   $T(n) = 6 \times n^{\log_3 2} - 5$.

   (b) $T(n) = 2T(n/3) + cn$.
   Hint:
   $T(n) = 3cn + (1 - 3c) \times n^{\log_3 2}$.

(c) $T(n) = 2T(n/4) + 7$.
Hint:
$T(n) = 8 \times n^{\frac{1}{2}} - 7$.

(d) $T(n) = 2T(n/4) + cn$.
Hint:
$T(n) = (1 - 2c) \times n^{\frac{1}{2}} + 2cn$.

(e) $T(n) = 3T(n/2) + 6$.
Hint:
$T(n) = 4 \times n^{\lg 3}$ - 3.

(f) $T(n) = 3T(n/2) + cn$.
Hint:
$T(n) = (2c + 1) \times n^{\lg 3} - 2cn$.

(g) $T(n) = 4T(n/2) + 3$.
Hint:
$T(n) = 2n^2 - 1$.

(h) $T(n) = 4T(n/2) + cn$.
Hint:
$T(n) = (c + 1) \times n^2 - cn$.

(i) $T(n) = 4T(n/2) + n^2$.
Hint:
$T(n) = (1 + \lg n) \times n^2$.

(j) $T(n) = 4T(n/2) + n^3$.
Hint:
$T(n) = 2n^3 - n^2$.

3. Write two recurrence relations for

   (a) the number of *additions* and *subtractions*,

   (b) the number of *multiplications* and *divisions*

   performed by the following function. DO NOT SOLVE THE RECURRENCES.

   Procedure $\mathtt{W}(n)$, n is an integer
   if $n < 4$ **then**
   return 1
   else
   return $2 \times \mathtt{W}(n-4) + \mathtt{W}(n-2) + \mathtt{W}(n/2)$

   Hint:

$$T_1(n) = \begin{cases} 0, & \text{if } n < 4, \\ T_1(n-4) + T_1(n-2) + T_1(\frac{n}{2}) + 4, & \text{if } n \geq 4. \end{cases}$$

$$T_2(n) = \begin{cases} 0, & \text{if } n < 4, \\ T_2(n-4) + T_2(n-2) + T_2(\frac{n}{2}) + 2, & \text{if } n \geq 4. \end{cases}$$

4. Give an exact solution to the following recurrence.

$$T(n) = \begin{cases} 0, & n = 1 \\ 6T\left(\frac{n}{3}\right) + 2n, & n \geq 2 \end{cases}$$

   You may assume that $n = 3^k$ for some non-negative integer k.

   Hint:
   $T(n) = 2n^{1+\log_3 2} - 2n$.

5. P67, Ex 4.1-1.

   Hint:

   - Without loss of generality, we assume $T(1) = 1$ and $T(n)$ is defined on all positive integers.
   - Secondly, observe that the ceiling implies that
     $T(3) = T(4)$
     $T(5) = T(6) = T(7) = T(8)$
     ...
     $T(2^{k-1} + 1) = T(2k - 1 + 2) = \ldots = T(2^k)$
     ...
   - Therefore, we want to get the closed form for $T(n)$ when $n = 2^k$, $k \geq 0$, which by using iterated substitution can be guessed as $T(2^k) = k+1$ and thus $T(n) = \lg n + 1$.
   - Fourthly, we want to prove that

$$T(n)(\leq \lceil \lg n \rceil + 1) \leq 2 \times \lg n + 1$$

     by induction.

6. P67, Ex 4.1-5.

   Hint: similar to 4.1-1, guess the constant factor(s) and prove by induction.