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 II

1. P75, Ex 4.3-1.

   Hints:

   - Since $a = 4, b = 2, c = 1,\ c < \log_b a = 2$, $T(n) \in \Theta(n^2)$
   - $T(n) \in \Theta(n^2 \log n)$
   - $T(n) \in \Theta(n^3)$

2. P75, Ex 4.3-2.

   Hints:

   - $T(n) \in \Theta(n^{\lg 7})$
   - When $a = 16, T'(n) \in \Theta(n^2 \log n)$, which is faster than $T(n)$
   - When $a < 16, T'(n) \in \Theta(n^2)$, which is also faster than $T(n)$
   - When $a > 16, T'(n) \in \Theta(n^{\log_4 a})$, which is faster than $T(n)$ only if $\log_4 a < \lg 7 = \log_4 49$
   - So, when $a < 49$, $T'(n)$ is asymptotically faster than $T(n)$.
   - Solution: 48.

3. P75, Ex 4.3-3.

4. P85, Prob 4-1.

   Hints:

   - Since $a = 2, b = 2, c = 3,\ c > \log_b a = 1$, $T(n) \in \Theta(n^3)$
   - $T(n) \in \Theta(n)$
   - $T(n) \in \Theta(n^2 \log n)$
   - $T(n) \in \Theta(n^2)$
   - $T(n) \in \Theta(n^{\lg 7})$
   - $T(n) \in \Theta(n^{\frac{1}{2}} \log n)$
   - $T(n) \in \Theta(n^2)$
   - $T(n) \in \Theta(\log \log n)$

5. P85, Prob 4-3.

   Hints: Let c be a constant. The general recurrence is (for base cases please figure out yourself)

   - $T(n) = T(\frac{n}{2}) + c \longrightarrow T(n) \in O(\log n)$
   - $T(n) = T(\frac{n}{2}) + c \times N \longrightarrow T(n) \in O(N \log n)$ and thus $T(n) \in O(n \log n)$ (N is the size of the original array)
   - $T(n) = T(\frac{n}{2}) + c \times \frac{n}{2} \longrightarrow T(n) \in O(n)$
   - Mergesort: $T(n) = 2 \times T(\frac{n}{2}) + (n-1) + c \longrightarrow T(n) \in O(n \log n)$
   - Mergesort: $T(n) = 2 \times T(\frac{n}{2}) + (n-1) + c \times N \longrightarrow T(n) \in O(n \log n + Nn)$ and thus $T(n) \in O(n^2)$
   - Mergesort: $T(n) = 2 \times T(\frac{n}{2}) + (n-1) + c \times \frac{n}{2} \longrightarrow T(n) \in O(n \log n)$

6. P130, Ex 6.1-2.

7. P130, Ex 6.1-3.

8. P130, Ex 6.1-7.

   Hints: $A[x]$ is a leaf if and only if $2x > n$.

9. P132, Ex 6.2-1.

10. P132, Ex 6.2-2.
11. P132, Ex 6.2-3.
    Hints: 2 KC, no exchangement.
12. P132, Ex 6.2-4.
    Hints: No action.
13. P132, Ex 6.2-5.
14. P132, Ex 6.2-6.
    Hints: What is the worst case.