CISC 5835 Algorithms for Big Data Fall, 2018

Homework Assignment #2

1 For each of the following situations, indicate whether $f = O(g)$, or $f = \Omega(n)$, or both (i.e., $f = \Theta(g)$). **Briefly explain why, either using the definition of Big-O notations, or using the four rules of thumbs given in Page 8 of DPV textbook, Section 0.3**.

(a) $f(n) = \log 100n$, $g(n) = \log 0.01n$

(b) $f(n) = n^{0.2}$, $g(n) = n^{1/2}$

(c) $f(n) = 10n^2 + 0.00001n$, $g(n) = 10000n$

(d) $f(n) = \log_2 n$, $g(n) = \log_{10} n$.

(e) $f(n) = 3^n$, $g(n) = 1.2^n + n^4$

2 Suppose you are choosing between the following three algorithms:

- Algorithm A solves problems by dividing them into 4 subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.
- Algorithm B solves problems of size n by recursively solving two subproblems of size $n-1$ and then combining the solutions in constant time.
- Algorithm C solves problems of size n by dividing them into 6 subproblems of size $n/3$, recursively solving each subproblem, and then combining the solutions in $O(n^2)$ time. what are the running times of each of these algorithms (in big-O notation), and which would you choose? (Hint: For A and C, you can use the Master's Theorem to find the asymptotic running time. For B, do you best to make a guess.)

3 Solve the following recurrence relations and give a Θ bound for each of the running time function.

(a) $T(n) = 2T(n/3) + 1$

(b) $T(n) = 5T(n/4) + n$

(c) $T(n) = T(n-1) + c^n$, where $c > 1$ is some constant. (Extra Credits)

4 Recall the formula below for summation of geometric sequence

$$1 + c + c^2 + c^3 + ... + c^n = \frac{c^{n+1} - 1}{c - 1}, \text{ for any positive integer, and for } c \neq 1$$

Making use of the formula to show that, if c is a postive real number, then $g(n) = 1 + c + c^2 + ... + c^n$ is:

(a) $\Theta(1)$ if $c < 1$.

(b) $\Theta(n)$ if $c = 1$.

(c) $\Theta(c^n)$ if $c > 1$.

5 How many lines does the following program print? Answer the question with a function of n (in $\Theta(.)$ form).

```
function f(n)
    if n>1:
       printline( "in here");
       f(n/3)
       f(n/3)
```