COMP2119 Introduction to algorithms and data structure

Victor Chui

October 6, 2022

1 Chapter 1: Notations

1.1 Big Oh

For a given function g(n), the definition is as below.

$$f(n) = O(g(n)) \text{ iff } \exists c > 0, \text{ s.t. } f(n) \geq c * g(n), \ \forall n \geq n_0$$

which c represents the constant multiplier for $g(n)$, and n_0 represent the minimum n for $g(n)$ to return an output.

P.S. iff means if and only if.

1.2 Big Omega

For a given function g(n), the definition is as below.

$$f(n) = \Omega(g(n)) \text{ iff } \exists c > 0, \text{ s.t. } f(n) \leq c * g(n), \ \forall n \geq n_0$$

which c represents the constant multiplier for $g(n)$, and n_0 represent the minimum n for $g(n)$ to return an output.

1.3 Big Theta

For a given function g(n), the definition is as below.

$f(n) = \Theta(g(n))$ iff $\exists c_1 > 0, c_2 > 0$ s.t. $0 \leq c_1 * g(n) \leq f(n) \leq c_2 * g(n), \ \forall n \geq n_0$

which c represents the constant multiplier for $g(n)$, and n_0 represent the minimum n for $g(n)$ to return an output.

2 Chapter 2: Recursion

2.1 Fundamental Elements of Recursion

2.2 Recurrence Equation

3 Chapter 3: Data Structure

3.1 Stack

3.2 Queue

3.3 Linked List

3.3.1 Singly Linked List

3.3.2 Doubly Linked List