Complexity Analysis

Data Structures and Algorithms in Java

Adam Drozdek

Computational and Asymptotic Complexity

- To evaluate an algorithm’s efficiency, real-time units such as microseconds and nanoseconds should not be used.
- Rather, it should be logical units between the size of n of a file or an array and the amount of time t required to process the data.

Asymptotic Complexity

- A measure of efficiency
- Used when disregarding certain terms of a function to express the efficiency of an algorithm or when calculating a function is difficult or impossible and only approximations can be found.

Big - O Notation

- Introduced in 1894 by Paul Bachman.
- Given two positive – valued functions f and g, consider the following definitions:
  – $f(n)$ is $O(g(n))$ if there exist positive numbers c and N_o such that $f(n) \le cg(n)$ for all $n \ge N_o$

Growth Rate of $f(n) = n^2 + 100n + \log_{10}n + 1{,}000$

n	f(n)	n^2	100n	log10n	1000
1	1,101	1	100	0	1000
10	2,101	100	4.76	1	1000
100	21,002	10,000	10,000	2	1000
1000	1,101,003	1,000,000	100,000	3	1,000
10,000	101,001,004	100,000,000	1,000,000	4	1,000
100,000	10,010,001,005	10,000,000,000	10,000,000,000	5	1,000

Logarithmic Form

Logarithmic Form	Exponential Form
$\text{Log}_2 16 = 4$	$4^2 = 16$
$\text{Log}_7 1 = 0$	$7^0 = 1$
$\text{Log}_5 5 = 1$	$5^1 = 5$
$\text{Log}_4 ¼ = -1$	$4^{-1} = ¼$
$\text{Log}_{10} 0.01 = \log 10\ 1/100 = -2$	$10^{-2} = 0.01$

Properties of Big – O Notation

- Fact 1:
  - If $f(n)$ is $O(g(n))$ and $g(n)$ is $O(h(n))$, then $f(n)$ is $O(h(n))$
- Fact 2:
  - If $f(n)$ is $O(h(n))$ and $g(n)$ is $O(h(n))$, then $f(n) + g(n)$ is $O(h(n))$

Properties of Big – O Notation

- Fact 3:
  - The function an^k is $O(n^k)$
- Fact 4:
  - The function n^k is $O(n^{k+j})$ for any positive j.
- Fact 5:
  - If $f(n) = cg(n)$, then $f(n)$ is $O(g(n))$
- Fact 6:
  - The function $\log_a n$ is $O(\log_b n)$ for any positive numbers a and $b \neq 1$

Properties of Big – O Notation

- Fact 7:
  - $\text{Log}_a n$ is $O(1_g n)$ for any positive $a \neq 1$, where $1_g n = \log_2 n$

Ω (big omega)and θ(theta) Notation

- The function f(n) is Ω(g(n) if there exist positive numbers c and N such that $f(n) \geq cg(n)$ for al $n \geq N$.
- F(n) is θ(g(n)) if there exist positive numbers c_1,c_2 and N such that $c_1g(n) \leq f(n) \leq c_2g(n)$ for all $n \geq N$

Prove:

- $4n^3 + 4n^2 + 6n + 4$ is $O(n^3)$
- $6n + 5$ is $O(n)$
- $2n + 3$ is $O(n^2)$
- $3n^2 + 6n + 1$ is $O(n)$

Programming Assignment: