UWI

Design and Analysis of Algorithms

Tutorial 2

Asymptotic Notations

Understanding asymptotic notations and their properties is essential when performing time efficiency analysis.

1. Explain the formal definitions of the following notations.
   (a) O (Big Oh)
   (b) Ω (Big Omega)
   (c) Θ (Big Theta)
2. Explain what is meant by the following asymptotic notation properties.
   (a) $f(n) \in O(f(n))$
   (b) $f(n) \in O(g(n)) \iff g(n) \in \Omega(f(n))$
   (c) If $f(n) \in O(g(n))$ and $g(n) \in O(h(n))$, then $f(n) \in O(h(n))$
   (d) If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$, then $t_1(n) + t_2(n) \in O(max\{g_1(n), g_2(n)\})$

Comparing Orders of Growth

We can compare orders of growth by using limits to derive whether two functions grow at the same, lower or higher rates.

1. State the implications if the below values result from the following limit:

$$\lim_{n\to\infty} \frac{t(n)}{g(n)} \tag{1}$$

   (a) 0 (Zero)
   (b) c (Some Positive Constant)
   (c) ∞ (Infinity)
2. State L'Hopital's rule and discuss when it should be used.

$$\lim_{n\to\infty} \frac{t(n)}{g(n)} = \lim_{n\to\infty} \frac{t'(n)}{g'(n)} \tag{2}$$

3. Compare the orders of growth of $\frac{n^2(n-4)}{4}$ and n^3 using limits.

Time Efficiency Calculations

Calculating the time efficiency of algorithms can involve the use of summation rules and formulas. It is important we learn to calculate the time efficiencies for simple brute force algorithms.

1. Analyse the following algorithms and calculate their time efficiencies using summation rules.
   (a) Algorithm 1: Sequential Search

```
for i ← 0 to n–1 do
    if A[i] = K then
        return i
    end if
end for
return –1
```

   (b) Algorithm 2: Matrix Multiplication

```
for i ← 0 to n–1 do
    for j ← 0 to n–1 do
        C[i, j] ← 0.0
        for k ← 0 to n–1 do
            C[i, j] ← C[i, j] + A[i, k] × B[k, j]
        end for
    end for
end for
return C
```