$a < b^K \rightarrow O(n^K)$

$a = b^K \rightarrow O(n^K \log n)$

$a > b^K \rightarrow O(n^{\log_b a})$

Algorithmics Unit 4 Week 4 Submit Task

1. Use the master theorem to find the time complexity of the following recurrence relations:

(a)

$$T(n) = \begin{cases} 3T\left(\frac{n}{2}\right) + 5n^2 & n > 1 \\ 4 & n = 1 \end{cases}$$

$3 < 2^2$

$O(n^2)$

(b)

$$T(n) = \begin{cases} 8T\left(\frac{n}{2}\right) + n + n^3 & n > 1 \\ 2 & n = 1 \end{cases}$$

$8 = 2^3$

$O(n^3 \log(n))$

(c)

$$T(n) = \begin{cases} 5T\left(\frac{n}{2}\right) + 5n & n > 1 \\ 4 & n = 1 \end{cases}$$

$5 < 2^1$

$O(n^1)$

2. An algorithm takes a matrix of size n x n and divides it into four matrices of size n/2 x n/2. These matrices are then recursively multiplied using the following standard approach:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

Assume that n is a positive power of 2, and that addition/multiplication of integers can be done in constant time.

Use the Master Theorem to show that the time complexity of this algorithm is $O(n^3)$.

$T(n) = 8T\left(\frac{n}{2}\right) + n^2$

$8 > 2^2$

$\therefore O(n^{\log_2 8})$

$O(n^3)$

8 Matrix multiplics → A = 8

Matricies are half as large as prev → B = 2

$4\times4 \rightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} \rightarrow \begin{bmatrix} AE+BG & AF+BH \\ CE+bG & CF+DH \end{bmatrix}$

4 OPS

$n = 4$

$\left(\frac{n}{2}\right)^2$