Quick Assignment: Master Theorem

Shubham Swami - A02315672

Master Theorem:

$T(n) = aT(n/b) + f(n)$

n = size of input

a = number of subproblems in the recursion.

n/b = size of each subproblem.

$f(n) = cn^k$ = cost of the work done outside the recursive call.

Case 1: $a < b^k$ => $T(n) \sim n^k$

Case 2: $a = b^k$ => $T(n) \sim n^k \log_b n$

Case 3: $a > b^k$ => $T(n) \sim n$ ^ $\log_b a$

Time Complexity Analysis:

1: Binary Search:

Here $a = 1$, $b = 2$, $k = 0$ => $f(n) = O(1)$

$T(n) = T(n/2) + O(1)$

$b^k = 1$ => $a = b^k$ => Case 2

$T(n) = n^k \log_b n$ => $n^0 \log_2 n$

$T(n) = \log_2 n$

$T(n) = \Theta(\log n)$

2: Merge Sort:

$a = 2$, $b = 2$, $k=1$ => $f(n) = O(n)$

$T(n) = 2T(n/2) + O(n)$

$b^k = 2$ => $a = b^k$

$T(n) = n^k \log_b n$ => $n \log_2 n$

$T(n) = n \log_2 n$

$T(n) = \Theta(n\log n)$

3: D&C Polynomial Multiplication:

From the lecture we simplified

$$\mathbf{PQ(x) = P_L Q_L x^0 + (P_H Q_L + P_L Q_H)\, x^{n/2} + P_H Q_H x^n}$$

i.e. 4 subproblems of size n/2 and the D&C polynomial multiply algorithm requires a linear scan through the arrays of size n each recursive call.

$a = 4$ $b = 2$, $k=1$ => $f(n) = O(n)$

$T(n) = 4T(n/2) + O(n)$ ($a = 4$, $b = 2$, $k = 1$)

$b^k = 2 \Rightarrow a > b^k \Rightarrow$ Case 3

$T(n) = n \wedge \log_b a$

$T(n) = n\wedge (\log_2 4)$ and $(\log_2 4 = 2)$

$T(n) = n^2$

$T(n) = \Theta(n^2)$

Summary:

Binary Search	Merge Sort	D&C Polynomial Multiplication
a = 1	a=2	a=4
b = 2	b=2	b=2
k = 0	k=1	k=1
$T(n) = \Theta(\log\ n)$	$T(n) = \Theta(n \log\ n)$	$T(n) = \Theta(n^2)$