Data Structures and Algorithms Theory

Ritik Singla

Index

1 Time Complexity

1.1 Master's Theorem

1.1.1 Master's Theorem for Dividing Functions

$T(n) = aT(\frac{n}{b}) + f(n)$ a $\geq 1, b > 1$

$f(n) = O(n^k log^p n)$

Case 1: If $log_b(a) > k$ Then TC $= O(n^{log_b(a)})$

Case 2: If $log_b(a) = k$

If $(p > -1)$ then TC $= O(n^k log^{p+1} n)$

If $(p = -1)$ then TC $= O(n^k log(log(n)))$

If $(p < -1)$ then TC $= O(n^k)$

Case 3: If $log_b(a) < k$

If $p \geq 0$ then TC $= O(n^k log^p n)$

else TC $= O(n^k)$

Example: Merge Sort $T(n) = T(\frac{n}{2}) + O(n)$

1.1.2 Master's Theorem for Decreasing Functions

$T(n) = aT(n-b) + f(n)$ **a** $>$ **0, b** $>$ **0**

f(n) $= O(n^k)$ **k** ≥ 0

Case 1: If a $=$ **1 then TC** $= O(nf(n))$

Case 2: If a $>$ **1 then TC** $= O(f(n)a^{\frac{n}{b}})$

Case 3: If a $<$ **1 then TC** $= O(f(n))$

Example: Quick Sort $T(n) = T(k) + T(n-k-1) + O(n)$

1.2 Time Complexities of Sorting Algorithms

Algorithm	**Best**	**Average**	**Worst**	**Stable**
Selection Sort	$\Omega(n^2)$	$\theta(n^2)$	$O(n^2)$	**No**
Bubble Sort	$\Omega(n)$	$\theta(n^2)$	$O(n^2)$	**Yes**
Insertion Sort	$\Omega(n)$	$\theta(n^2)$	$O(n^2)$	**Yes**
Heap Sort	$\Omega(nlog(n))$	$\theta(nlog(n))$	$O(nlog(n))$	**No**
Quick Sort	$\Omega(nlog(n))$	$\theta(nlog(n))$	$O(n^2)$	**No**
Merge Sort	$\Omega(nlog(n))$	$\theta(nlog(n))$	$O(nlog(n))$	**Yes**