ALGORITHMIC ANALYSIS

Big O: $f(n) \in O(g(n)) \longleftrightarrow \exists c \in \mathbb{R}, n_0 \in \mathbb{Z}^{\geq 0}$ s.t. $\forall n \geq n_0, f(n) \leq c \cdot g(n)$

Big Ω: $f(n) \in \Omega(g(n)) \longleftrightarrow \exists c \in \mathbb{R}, n_0 \in \mathbb{Z}^{\geq 0}$ s.t. $\forall n \geq n_0, f(n) \geq c \cdot g(n)$

Big Θ: $f(n) \in \Theta(g(n)) \longleftrightarrow f(n) \in O(g(n)) \wedge f(n) \in \Omega(g(n))$

$\longleftrightarrow \exists c_1, c_2 \in \mathbb{R}, n_0 \in \mathbb{Z}^{\geq 0}$ s.t. $\forall n \geq n_0, c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

Growth Rates: $1 \rightarrow \log(n) \rightarrow n \rightarrow n \log(n) \rightarrow n^2 \rightarrow n^3 \rightarrow c^n \rightarrow n!$

RECURSION ANALYSIS

Case 1: work per call follows pattern - $T(n) = T(n-1) + O(1)$

Case 2: work per level is the same - $T(n) = 2T(\frac{n}{2}) + O(n)$

Case 3: work per call is the same - $T(n) = 2T(n-1) + O(1)$

SORTING ALGORITHMS

Sort	Worst	Best	Expected
Bubble $A[i] \overset{swap}{\leftrightarrow} A[i+1]$	$O(n^2)$	$O(n)$ *pre-sorted*	$O(n^2)$
Insertion *by next unsorted*	$O(n^2)$	$O(n)$ *pre-sorted*	$O(n^2)$
Selection *by smallest*	$O(n^2)$	$O(n)$ *pre-sorted*	$O(n^2)$
Merge	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Quick *3-partition*	$O(n^2)$	$O(n)$ *uniform list*	$O(n \log n)$
Radix	$O(d(n+N))$	$O(d(n+N))$	$O(d(n+N))$

DATA STRUCTURES

Amortisation: $T(n) \div n$, where $n =$ no. of operations

test