ANALYSIS OF MERGESORT

WORST CASE
$T(n) = O(n \log n)$

Let n be the number of elements to be sorted.
Mergesort breaks up the list of n elements to two parts.
Part 1: elements 1 to floor(n/2)
Part 2: elements ceil(n/2) to n
We have the recurrence

$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + (n-1)$ with $T(1) = 0$

$$\equiv T(n_f) + T(n_c) + (n-1)$$

Let n be a power of 2 say $n = 2^k$

This gives $T(n) = 2T(n/2) + (n-1)$ with $T(1) = 0$

$T(n) = 2T(n/2) + (n-1)$	
	$T(n/2) = 2T(n/2^2) + (n/2 - 1)$
$2T(n/2) = 2^2T(n/2^2) + (n - 1)$	
	$T(n/2^2) = 2T(n/2^3) + (n/2^2 - 1)$
$2^2T(n/2^2) = 2^3T(n/2^3) + (n - 1)$	
	$T(n/2^3) = 2T(n/2^4) + (n/2^3 - 1)$
$2^3T(n/2^3) = 2^4T(n/2^4) + (n - 1)$	
	$T(n/2^4) = 2T(n/2^5) + (n/2^4 - 1)$
$2^4T(n/2^4) = 2^5T(n/2^5) + (n - 1)$	

$2^{k-3}T(n/2^{k-3}) = 2^{k-2}T(n/2^{k-2}) + (n - 1)$	
	$T(n/2^{k-2}) = 2T(n/2^{k-1}) + (n/2^{k-2} - 1)$
$2^{k-2}T(n/2^{k-2}) = 2^{k-1}T(n/2^{k-1}) + (n - 1)$	
	$T(n/2^{k-1}) = 2T(n/2^k) + (n/2^{k-1} - 1)$
$2^{k-1}T(n/2^{k-1}) = 2^kT(2^k/2^k) + (n - 1)$ $= 2^kT(1) + (n - 1)$	

$$\Sigma_{lhs} = T(n) + [\, 2T(n/2) + 2^2T(n/2^2) + 2^3T(n/2^3) + \text{----} + 2^{k-2}T(n/2^{k-2}) + 2^{k-1}T(n/2^{k-1})]$$

$$\Sigma_{rhs} = [\, 2T(n/2) + 2^2T(n/2^2) + 2^3T(n/2^3) + \text{----} + 2^{k-2}T(n/2^{k-2}) + 2^{k-1}T(n/2^{k-1})]$$

$+2^{k}T(1)$

$+[\ (n-1) + (n-1 + (n-1) + \text{-----} k \text{ times}\ ]$

Equating both sides

$T(n) = (n-1)\log_2(n-1) = O(n\log n)$