RUNTIME ANALYSIS OF FUNCTIONS

ANALOGY TO REAL NUMBERS

$f(n) = O(g(n))$	$a \leq b$
$f(n) = \Omega(g(n))$	$a \geq b$
$f(n) = \Theta(g(n))$	$a = b$
$f(n) = o(g(n))$	$a < b$
$f(n) = \omega(g(n))$	$a > b$

EXAMPLES

$logn$	$\boxed{}$	$O(log^2n)$	log^2n	$\boxed{}$	$O(log^2n)$	$nlogn$	$\boxed{}$	$O(log^2n)$
$logn$	$\boxed{}$	$\Theta(log^2n)$	log^2n	$\boxed{}$	$\Theta(log^2n)$	$nlogn$	$\boxed{}$	$\Theta(log^2n)$
$logn$	$\boxed{}$	$\Omega(log^2n)$	log^2n	$\boxed{}$	$\Omega(log^2n)$	$nlogn$	$\boxed{}$	$\Omega(log^2n)$
$logn$	$\boxed{}$	$o(log^2n)$	log^2n	$\boxed{}$	$o(log^2n)$	$nlogn$	$\boxed{}$	$o(log^2n)$
$logn$	$\boxed{}$	$\omega(log^2n)$	log^2n	$\boxed{}$	$\omega(log^2n)$	$nlogn$	$\boxed{}$	$\omega(log^2n)$

DIVIDE AND CONQUER

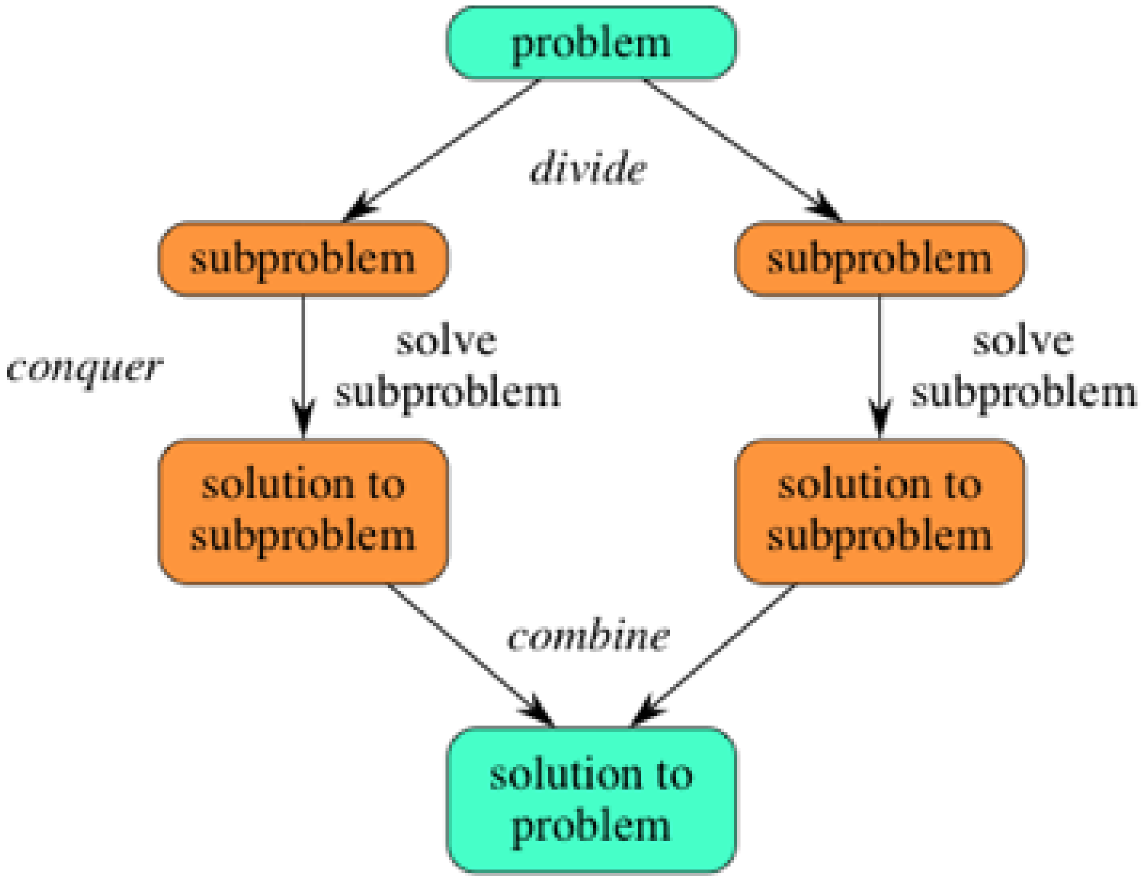

Figure 1: Taken from `https://www.khanacademy.org/computing/computer-science/algorithms/merge-sort/a/divide-and-conquer-algorithms`

```
void mergesort(int *A, int n) {
  if (n <= 1) return; else {          ← Divide
    mergesort(A, n/2);
    mergesort(A+n/2, n-n/2);          ← Conquer
    A = merge(A, n/2, A+n/2, n-n/2);  }}  ← Combine
```

Figure 2: Example from class: pseudocode of merge sort

EXERCISES FROM THE BOOK

3-2 Relative asymptotic growths

Indicate, for each pair of expressions (A, B) in the table below, whether A is O, o, Ω, ω, or Θ of B. Assume that $k \geq 1$, $\epsilon > 0$, and $c > 1$ are constants. Your answer should be in the form of the table with "yes" or "no" written in each box.

	A	B	O	o	Ω	ω	Θ
a.	$\lg^k n$	n^ϵ					
b.	n^k	c^n					
c.	$\sqrt{n}$	$n^{\sin n}$					
d.	2^n	$2^{n/2}$					
e.	$n^{\lg c}$	$c^{\lg n}$					
f.	$\lg(n!)$	$\lg(n^n)$					

4.4-2

Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = T(n/2) + n^2$. Use the substitution method to verify your answer.

4.4-4

Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 2T(n-1) + 1$. Use the substitution method to verify your answer.