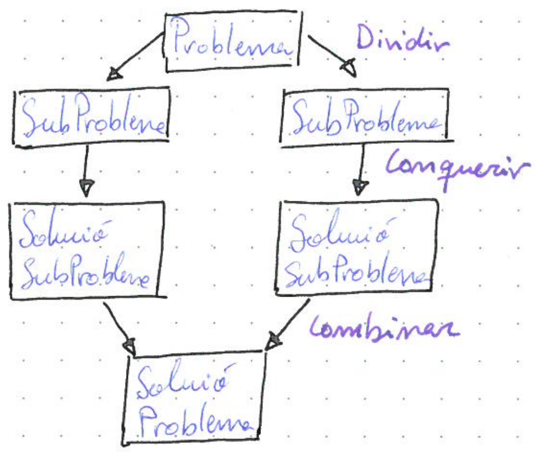
Divide And Conquer



Si $f(n) = \Theta(n^k)$ i $T(n) = \begin{cases} g(n) & n \le n_0 \\ a \cdot T(n/b) + f(n) & n > n_0 \end{cases}$

Llavors tenim que: $[\alpha = \log_b(a)]$

$$\boxed{T(n) = \begin{cases} \Theta(n^k) & \text{si } \alpha < k \text{ o } a < b^k \\ \Theta(n^k \cdot \log n) & \text{si } \alpha = k \text{ o } a = b^k \\ \Theta(n^\alpha) & \text{si } \alpha > k \text{ o } a > b^k \end{cases}}$$

Recordem que:

$f(n)$ Cost de dividir i Combinar

$g(n)$ Cost del Cas Fàcil

Karatsuba Algorithm

$X = \langle a, b \rangle$ on $X = a \times 10^{n/2} + b$ sent n el tamany de X. $1234 = \langle 12, 34 \rangle = 12 \times 10^2 + 34$

$Y = \langle c, d \rangle$ on $Y = c \times 10^{n/2} + d$ sent n el tamany de Y. $9876 = \langle 98, 76 \rangle = 98 \times 10^2 + 76$

$X \cdot Y = (a \times 10^{n/2} + b)(c \times 10^{n/2} + d) = ac \times 10^n + (ad + bc) \times 10^{n/2} + bd$

Veiem que hi ha 4 multiplicacions, però realment podem fer 3 sabent que:

$(a+b)(c+d) - ac - bd = ad + bc$

Llavors fem crida recursiva de:

```
ac_multi = Karatsuba(a, c);
bd_multi = Karatsuba(b, d);
ad_bc_multi = Karatsuba(a+b, c+d) - ac - bd;
```

Aquí estem fent un Divide & Conquer

```
// Base Case
if (x<10 or y<10)
    return x*y;
```

```
def karatsuba(x, y):
  // Base Case
  if x<10 or y<10 return x*y
  // General Case
  n = max(len(x), len(y))
  a = x ÷ 10^(n/2)
  b = x % 10^(n/2)
  c = y ÷ 10^(n/2)
  d = y % 10^(n/2)

  ac = karatsuba(a, c)
  bd = karatsuba(b, d)
  ad_bc = karatsuba(a+b, c+d) - ac - bd
  return ac * 10^n + (ad_bc) x 10^(n/2) + bd
```

Nº crides recursives: $a = 3$

Tamany que dividim: $b = 2$ Dividim dues parts el número

Cost de dividir i Combinar: $f(n) = \Theta(n)$

$\div 10^{n/2}$ sig. agafar els primers dígits $O(1)$

$\% 10^{n/2}$ sig. agafar els últims dígits $O(1)$

$+$, $-$ } Si que tenen cost $\Theta(n)$ pq s'ha de recórrer tot.

Realment en dividir fem $O(1)$ i conquerir fem dues sumes $\Theta(2n) = \Theta(n) = f(n)$

Cost Algoritme: $\boxed{T(n) = O(n^{1'58})}$

$n^k = n^1$ si $k=1$; $\log_b(a) = \alpha = 1'58 \Rightarrow \alpha > k \xRightarrow{\text{Master Theorem}} T(n) = O(n^\alpha)$