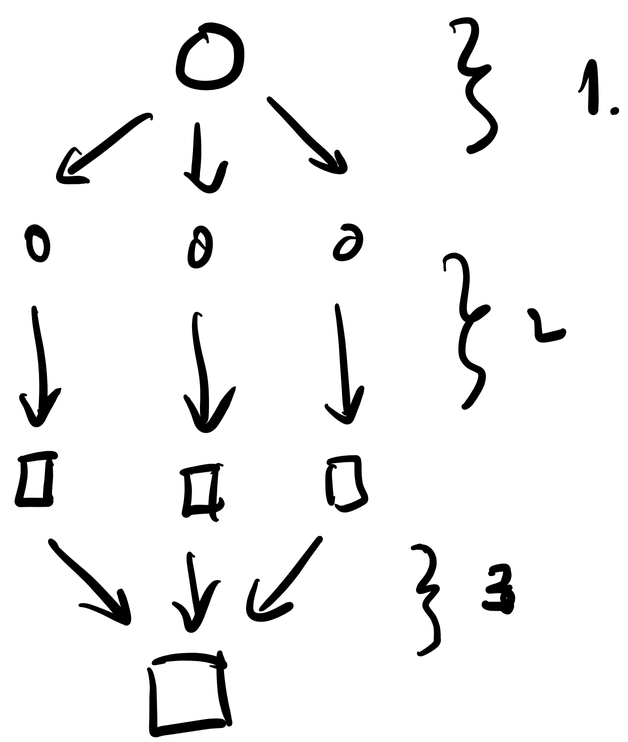
LECTURE 6 : DIVIDE & CONQUER

"Divide et impera."

— Philip II of Macedonia
382 - 336 BC

Divide & conquer is a strategy with 3 steps:

1. Divide a problem into subproblems
2. Turn subproblems into subsolutions
3. Conquer subsolutions into a solution.



```
> msort :: [Int] -> [Int]
> msort []  = []
> msort [x] = [x]
> msort xs  =
>       merge (msort us) (msort vs)
>     where
>       (us, vs) = splitAt (n `div` 2) xs
>       n        = length xs
```

conquer

divide

We need to define merge:

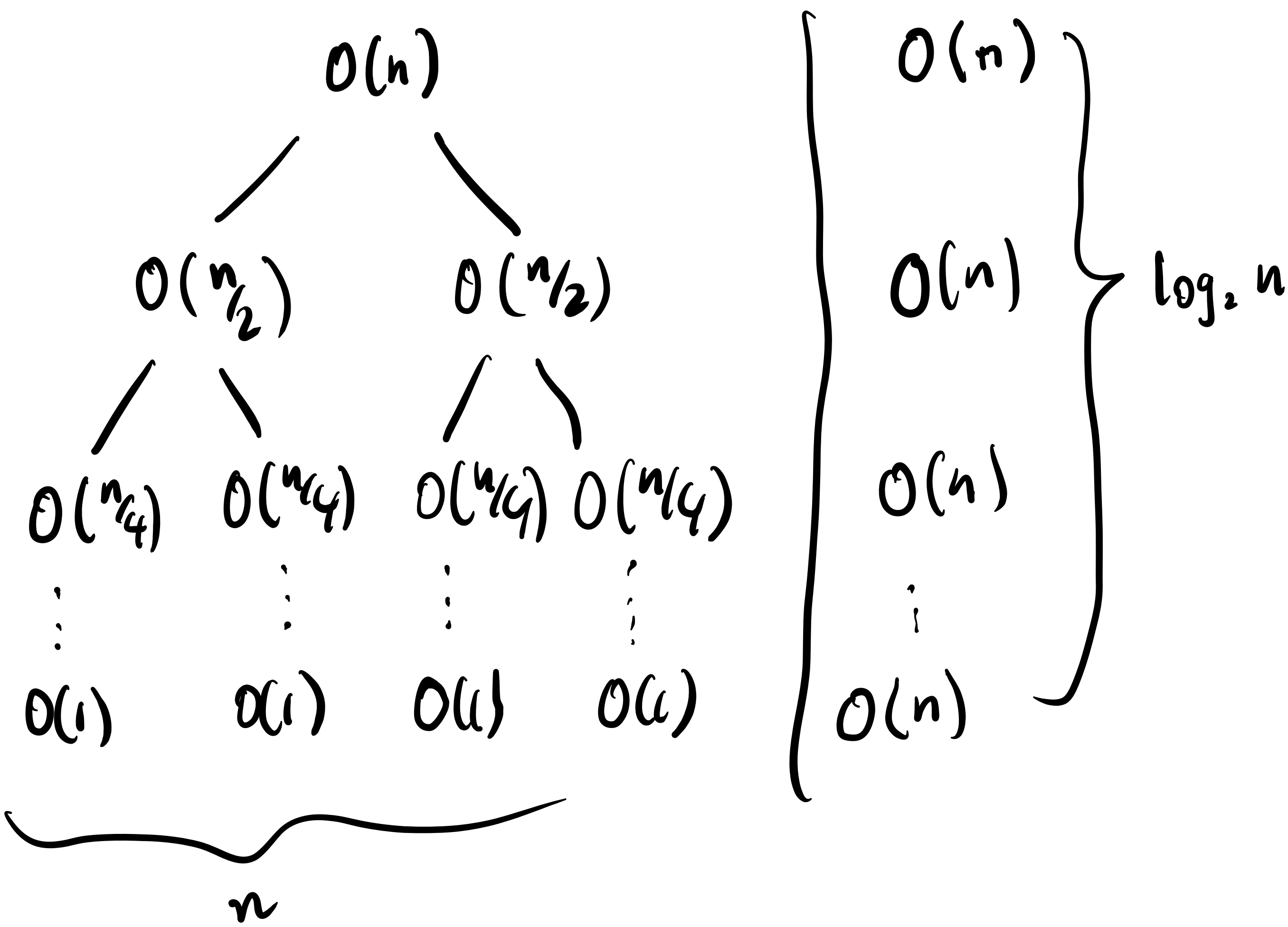
```
> merge :: [Int] -> [Int] -> [Int]
> merge [] ys = ys
> merge xs [] = xs
> merge xxs@(x:xs) yys@(y:ys)
>     | x <= y    = x : merge xs yys
>     | otherwise = y : merge xxs ys
```

$$T(\text{msort})\ 0 = 1$$

$$T(\text{msort})\ 1 = 1$$

$$T(\text{msort})\ n = T(\text{length})\ n + T(\text{splitAt})\left(\frac{n}{2}\right) + 2 \cdot T(\text{msort})\ (n/2)$$

$+ T(\text{merge})(n/2)(n/2)$



$$T(\text{msort})(n) \in \Theta(n \cdot \log n)$$

Quicksort.

Quicksort is also a D&C algorithm

```
> qsort :: [Int] -> [Int]
> qsort []  = []
> qsort [x] = [x]
> qsort (x:xs)
```

conquer.

```
>      qsort us ++ [x] ++ qsort vs
>        where
```
(x : qsort vs)

```
>             (us, vs) = partition (≤ x) xs
```
divide

```
> partition :: (a -> Bool) -> [a] -> ([a], [a])
> partition p xs =
>     ( filter p xs, filter (not . p) xs )
```

Assuming the lists are split into two equal parts:

$$
\begin{aligned}
T(qsort)(0) &= 1 \\
T(qsort)(1) &= 1 \\
T(qsort)(n) &= T(partition)(n) \\
&\quad + T(++)(n/2) \\
&\quad + 2 \cdot T(qsort)(n/2)
\end{aligned}
$$

In the worst case, partition splits xs into lists of length 0 and n-1:

$$
\begin{aligned}
T(qsort)(0) &= 1 \\
T(qsort)(1) &= 1 \\
T(qsort)(n) &= T(partition)(n) \\
&\quad + T(++)(n-1) \\
&\quad + T(qsort)(0) \\
&\quad + T(qsort)(n-1)
\end{aligned}
$$

$$= O(n) + T_{qsort}(n-1)$$

$$= O(n) + \left(O(n-1) + T_{qsort}(n-2)\right)$$

$$= \underbrace{O(n) + \cdots + O(n)}_{n}$$

$$= O(n^2)$$