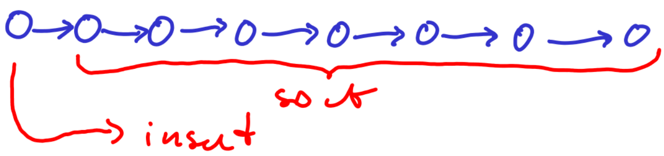
G54 AAD 2015/2016

Lecture 04

DIVIDE & CONQUER

Split problem into subproblems,
solve subproblems recursively,
combine the solutions.

INSERTION SORT

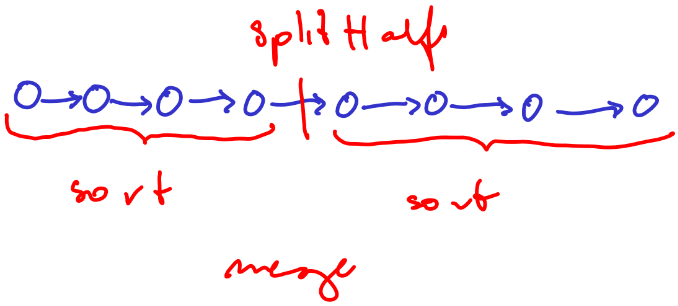


$T(n) = T(n-1) + \Theta(n)$

worst case

$\Rightarrow T(n) = \Theta(n^2)$

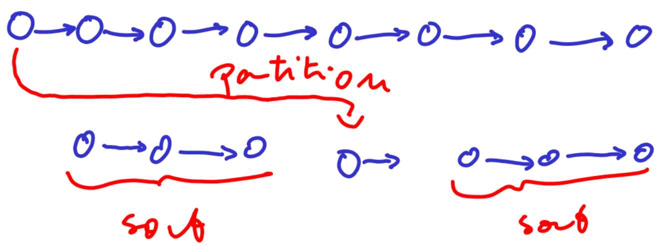
MERGE SORT



$T(n) = 2T(n/2) + \Theta(n)$

$\Rightarrow T(n) = \Theta(n \log n)$

QUICK SORT



$T(n) = T(n_1) + T(n_2) + \Theta(n)$

when $n_1 + n_2 = n - 1$

$\Rightarrow T(n) = O(n \log n)$ best case $n_1 = n_2$

$T(n) = O(n^2)$ worst case $n_1 = 1,\ n_2 = n-2$

```
mergesort xs =
    merge (mergesort left) (mergesort right)
    where (left, right) = split xs
```

Pseudo code
Haskell later

```
merge (x:xs) (y:ys) =
    { x:(merge xs y:ys)   if x ≤ y
    { y:(merge x:xs ys)   otherwise.
```

Excluded during lecture.

Lemma

split and merge have worst-case runtime $f(n) = \Theta(n)$.

Thm:

The worst case runtime of mergesort is $\Theta(n \log n)$.

Proof

The worst-case runtime of mergesort is

$$T(n) = 2T(n/2) + f(n), \quad \text{where}$$

$f(n) = \Theta(n)$ is the worst-case time for merge and split.

We apply the Master Theorem with $a = b = 2$.

Because $f(n) \in \Theta(n^{\log_b a}) = \Theta(n)$ it follows that

$$T(n) = \Theta(n^{\log_b a} \log n) = \Theta(n \log n).$$

$$\boxed{\begin{aligned}&\text{insertsort } (x:xs) = \\ &\quad \text{insert } x \text{ (insertsort } xs)\end{aligned}}$$

Lemma

insert has worst-case runtime $f(n) = \Theta(n)$

Thm

insertionsort has worst-case runtime $\Theta(n^2)$.

Proof

By the iteration method

$$\begin{aligned} T(n) &= T(n-1) + f(n) \\ &= T(n-2) + f(n-1) + f(n) \\ &\;\vdots \\ &= \underbrace{\sum_{i=0}^{n-1} f(n-i) = \Theta\left(\sum_{i=0}^{n-1} n-i\right)}_{\substack{g(n) = \Theta(h_1(n)) + \Theta(h_2(n)) \\ \Rightarrow g(n) = \Theta(h_1(n) + h_2(n))}} = \Theta(n^2) \end{aligned}$$

```
quicksort (x : xs) =
  (quicksort left) ++ [x] ++ (quicksort right)
  where (left, right) = partition x xs
```

Lemma

partition has worst-case runtime $f(n) = \Theta(n)$.

Theorem

quicksort has worst-case runtime $\Theta(n^2)$ and best-case runtime $\Theta(n \log n)$.

Proof

$$T(n) = T(n_1) + T(n_2) + f(n)$$

Worst case: $n_1 = 0$ $n_2 = n - 1$ in every recursion

$$\begin{aligned} T(n) &= T(n-1) + T(0) + f(n) \\ &\vdots \\ &= (n-1)T(0) + \sum_{i=0}^{n-1} f(n-i) \\ &= \Theta\left(\sum_{i=0}^{n-1} n - i\right) = \Theta(n^2) \end{aligned}$$

Best case: $n_1 = \frac{n}{2}$ and $n_2 = \frac{n}{2}$ in every recursion

$$T(n) = 2T(n/2) + f(n)$$

$$\Rightarrow T(n) = \Theta(n \log n).$$