Lezione 2

2.1 Esempio: Analisi di Merge-Sort

```
MERGE-SORT(A, p, r)
    if p ≥ r                              // zero or one element?
        return
    q = ⌊(p + r)/2⌋                       // midpoint of A[p : r]
    MERGE-SORT(A, p, q)                   // recursively sort A[p : q]
    MERGE-SORT(A, q + 1, r)               // recursively sort A[q + 1 : r]
    // Merge A[p : q] and A[q + 1 : r] into A[p : r].
    MERGE(A, p, q, r)
```

DIVIDE $\rightsquigarrow D(n) = \Theta(1)$

CONQUER $\rightsquigarrow 2T\left(\frac{n}{2}\right)$

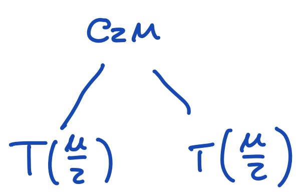
COMBINE $\rightsquigarrow C(n) = \Theta(n)$

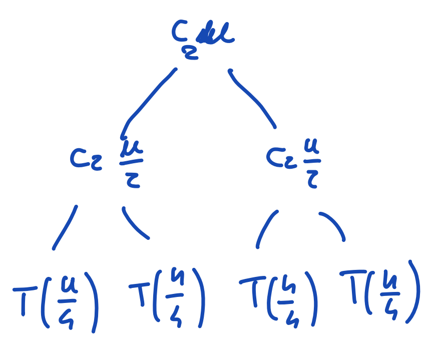
$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n) & n > 1 \end{cases}$$

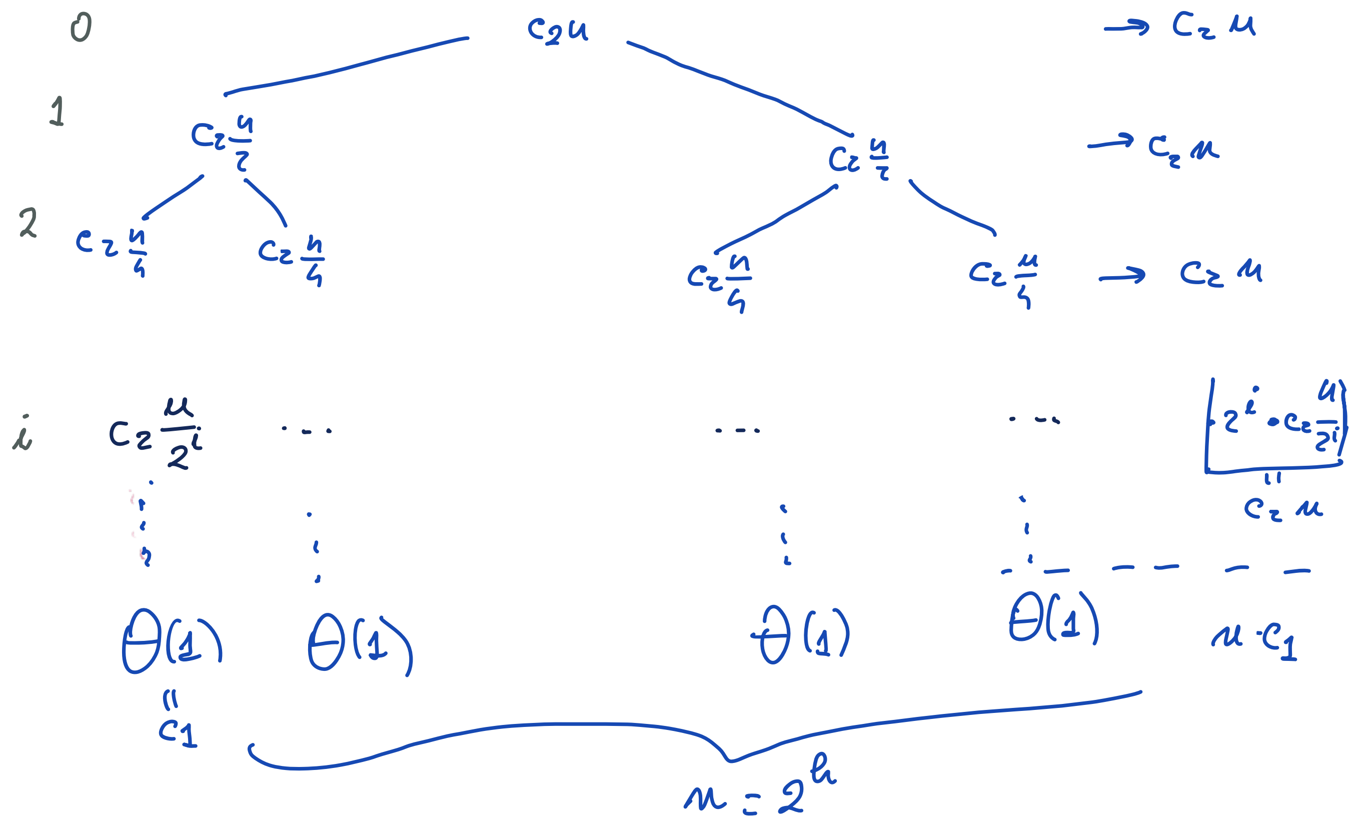
$$T(n) = \begin{cases} c_1 & n = 1 \\ 2T\left(\frac{n}{2}\right) + c_2 n & n > 1 \end{cases}$$

$T(n)$

step 0







$$h : \quad \frac{n}{2^h} = 1 \Rightarrow n = 2^h \Rightarrow h = \log n$$

$$T(n) = \underbrace{\log n}_{\text{livelli prima delle foglie}} \cdot c_2 n + n c_1 = c_2 n \log n + c_1 n = O(n \log n)$$

$$\Theta(n \log n)$$

2.2 EQUAZIONI DI RICORRENZA

DEF.: Un'eq. di ricorrenza è una formula che descrive il valore di una funzione in corrispondenza di un argomento generico x, in termini del valore della stessa funzione su argomenti di dimensione generalmente inferiore.

⇓

caratterizzano il tempo computazionale degli algoritmi ricorsivi

$$T(n) = \underset{\text{divide}}{D(n)} + \underset{\text{combina}}{C(n)} + aT\left(\frac{n}{b}\right) \qquad \forall n \geq n_0$$

$$\exists\, n_0 > 0 : \quad \forall n < n_0 \qquad T(n) = \Theta(1)$$

la ricorsione produce a sottoproblemi di dimensione $\frac{n}{b}$

$$T(n) = \begin{cases} \Theta(1) & n < n_0 \quad \text{(caso base)} \\ D(n) + aT\left(\frac{n}{b}\right) + C(n) & \forall n \geq n_0 \quad \text{(caso ricorsivo)} \end{cases}$$

2.3 IL METODO DI SOSTITUZIONE

1. Guess della soluzione usando costanti simboliche
2. Si usa l'induzione matematica per verificare la soluzione e trovare i valori delle costanti.

Esempio (a): $T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \Theta(n)$ (#)

guess: $T(n) = O(n \log n)$

Ipotesi induttiva: $T(n) \le c\, n \log n$ $\exists n_0 > 0, \exists c > 0$ $\forall n \ge n_0$

$\frac{n}{2} \ge n_0 \Rightarrow n \ge 2n_0$ $T\left(\frac{n}{2}\right) \le c\, \frac{n}{2} \log\left(\frac{n}{2}\right)$

$T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \Theta(n) \le 2c \left\lfloor \frac{n}{2} \right\rfloor \log\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \Theta(n) \le$

$\cancel{2}\, c\, \frac{n}{\cancel{2}} \log\left(\frac{n}{2}\right) + \Theta(n) \le c\, n \log(n) - \underbrace{c\, n \log 2 + \Theta(n)}_{\text{se prendo } c \text{ in modo che } cn > dn \text{ (d costante nascosta in } \Theta(n))} \le$

$\le c\, n \log(n)$

Base: Se $n_0 \le n < 2n_0$

$\log n_0 > 0 \rightsquigarrow n_0 = 2$

$2 \le n < 4$ $n = 3$

$c = \max\{T(2), T(3)\}$

$$\left.\begin{aligned} T(2) &\le c < 2\log 2 \cdot c \\ T(3) &\le c < 3\log 3 \cdot c \end{aligned}\right\} \Rightarrow T(n) \le c\, n \log n \quad \forall n \ge 2$$

$T(n) = O(n \log n)$ è sol di #.

Esempio (b): $T(n) = 2T\left(\frac{n}{2} + 17\right) + \Theta(n)$

guess: $O(n \log n)$

TRUCCHETTO : sottrarre un termine di grado inferiore

Esempio (c)

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(1)$$

guess: $O(n)$

Ipotesi ind: $T(n) \leq cn$

$$T(n) \leq 2\left(c\frac{n}{2}\right) + \underbrace{\Theta(1)}_{c'} = cn + \underset{\overset{\vee}{0}}{c'} \not\leq cn$$

Ipotesi induttive 2: $T(n) \leq cn - \underset{*}{d}$

$$T(n) \leq 2\left(c\frac{n}{2} - d\right) + \Theta(1) = cn - 2d + \Theta(1) =$$

$$= (cn - d) - \underbrace{(d - \Theta(1))}_{\vee 0} \leq cn - d$$

Es (d):

$$T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \Theta(n)$$

guess: $O(n)$ (che sappiamo essere sbagliato)

ipotesi ind: $T(n) = O(n)$

sost.

$$T(n) = 2\,O\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \Theta(n) = 2\,O(n) + \Theta(n) = O(n)$$

Es (e):

$$T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \Theta(n)$$

guess: $O(n)$ sbagliato

ipotesi ind: $T(n) \leq cn$

sost: $T(n) \leq 2c\left\lfloor \frac{n}{2} \right\rfloor + \Theta(n) \leq cn + \Theta(n) = O(n)$

$$cn + \underbrace{\Theta(n)}_{\Downarrow \ dn} \leq cn$$

$$\underbrace{(c+d)}_{\substack{\vee \\ c}} n \leq cn$$

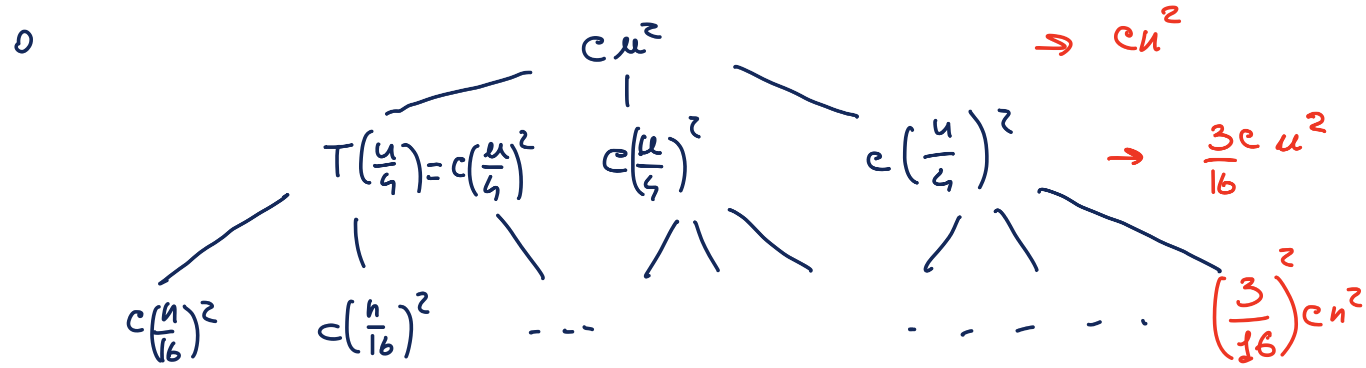
2.4. METODO DELL'ALBERO DI RICORSIONE

In un albero di ricorsione, ogni nodo rappresenta il costo di un singolo sottoproblema che si trova nell'insieme delle chiamate alla funzione ricorsiva.

In genere,
Si sommano i costi ad ogni livello dell'albero per ottenere i costi di ogni livello e si sommano per tutti i livelli, così si ottiene il tempo computazionale

Esempio: $\quad T(n) = 3T\left(\frac{n}{4}\right) + \Theta(n^2)$

$$T(n) = 3T\left(\frac{n}{4}\right) + cn^2 \quad , \quad \exists\, c > 0$$



$i \quad c\left(\frac{n}{4^i}\right)^2$ costo di un nodo , 3^i nodi $\to \left(\frac{3}{16}\right)^i cn^2$

$h: \quad \frac{n}{4^h} = 1 \Rightarrow h = \log_4 n \quad , \quad 3^h = 3^{\log_4 n} = n^{\log_4 3}$

$$T(n) = \underbrace{\sum_{i=0}^{h-1} \left(\frac{3}{16}\right)^i cn^2}_{cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \dots + \left(\frac{3}{16}\right)^{h-1} cn^2} + n^{\log_4 3} \cdot c' =$$

$\log_4 3 < 2$

$$\le \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i cn^2 + c'n^2 \le \frac{1}{1-\frac{3}{16}} cn^2 + \Theta(n^2) = O(n^2)$$

$$\sum_{i=0}^{\infty} q^i = \frac{1}{1-q}$$

$-1 \le q \le 1$

Ex: $\quad T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + \Theta(n)$