Titolo nota 10/03/2016

$T(n) = a\,T(n/b) + f(n)$ Teorema dell'esperto
Teorema fondamentale della ricorrenza

1) se $f(n) = O\left(n^{\log_b a - \varepsilon}\right)$ per $\varepsilon > 0$: $T(n) = \Theta\left(n^{\log_b a}\right)$

2) se $f(n) = \Theta\left(n^{\log_b a}\right)$: $T(n) = \Theta\left(n^{\log_b a} \log n\right)$

$\log n = o\left(n^{\varepsilon}\right)$ per $\varepsilon > 0$

3) se $f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right)$ per $\varepsilon > 0$
& $a f\left(\frac{n}{b}\right) \le c\, f(n)$ $c < 1$ } : $T(n) = \Theta\left(f(n)\right)$

condizione di regolarità

$T(n) = 3\,T\left(\frac{n}{4}\right) + n \log n$

$3 \frac{n}{4} \log \frac{n}{4} \le \frac{3}{4} n \log n$ $\quad c = \frac{3}{4} < 1$

$a = 3$ $\quad b = 4$ $\quad n^{\log_b a} = n^{\log_4 3} \simeq n^{0.79}$

$f(n) = n \log n : n^{0.79}$ $\quad$ per $\varepsilon \le 0.2$ $\quad n \log n = \Omega\left(n^{0.79 + \varepsilon}\right)$

$n \log n$ cresce più di n ma non in modo polin.

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

$$T(n) = 2\,T\left(\frac{n}{2}\right) + n \log n \qquad a = 2 \quad b = 2 \qquad n^{\log_b a} = n$$

$f(n)$: $n^{\log_b a}$

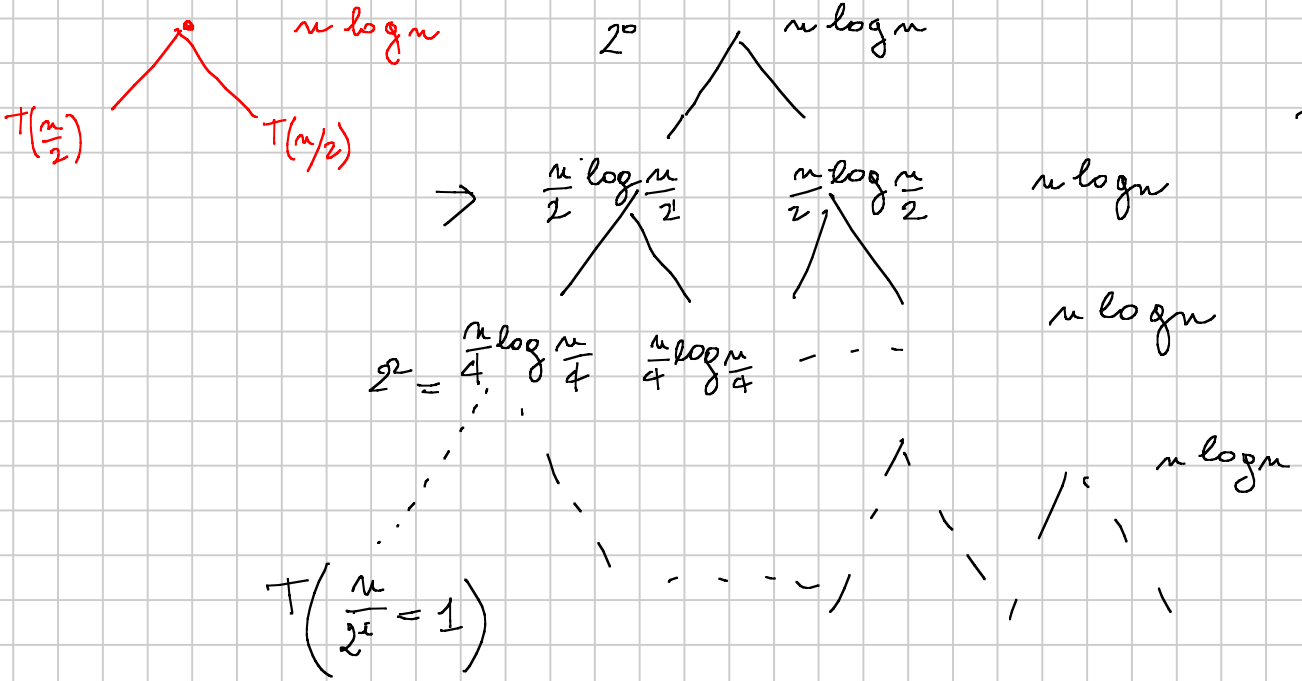
$n \log n$: n ?

$n \log n$ è $\Omega(n^{1+\varepsilon})$?

n

non si può risolvere col Teorema principale

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



$$T(n) = i \cdot n\log n$$

$$T(n) = \log n \cdot n\log n = n\log^2 n$$

$$\frac{n}{2^i} = 1 \qquad n = 2^i$$

$$i = \log_2 n$$

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2}{3}n\right) + cn$$

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{n}{3}\right) + cn = 2T\left(\frac{n}{3}\right) + cn \quad \text{Th.}$$

$$T'(n) \leq 2T\left(\frac{2}{3}n\right) + cn$$

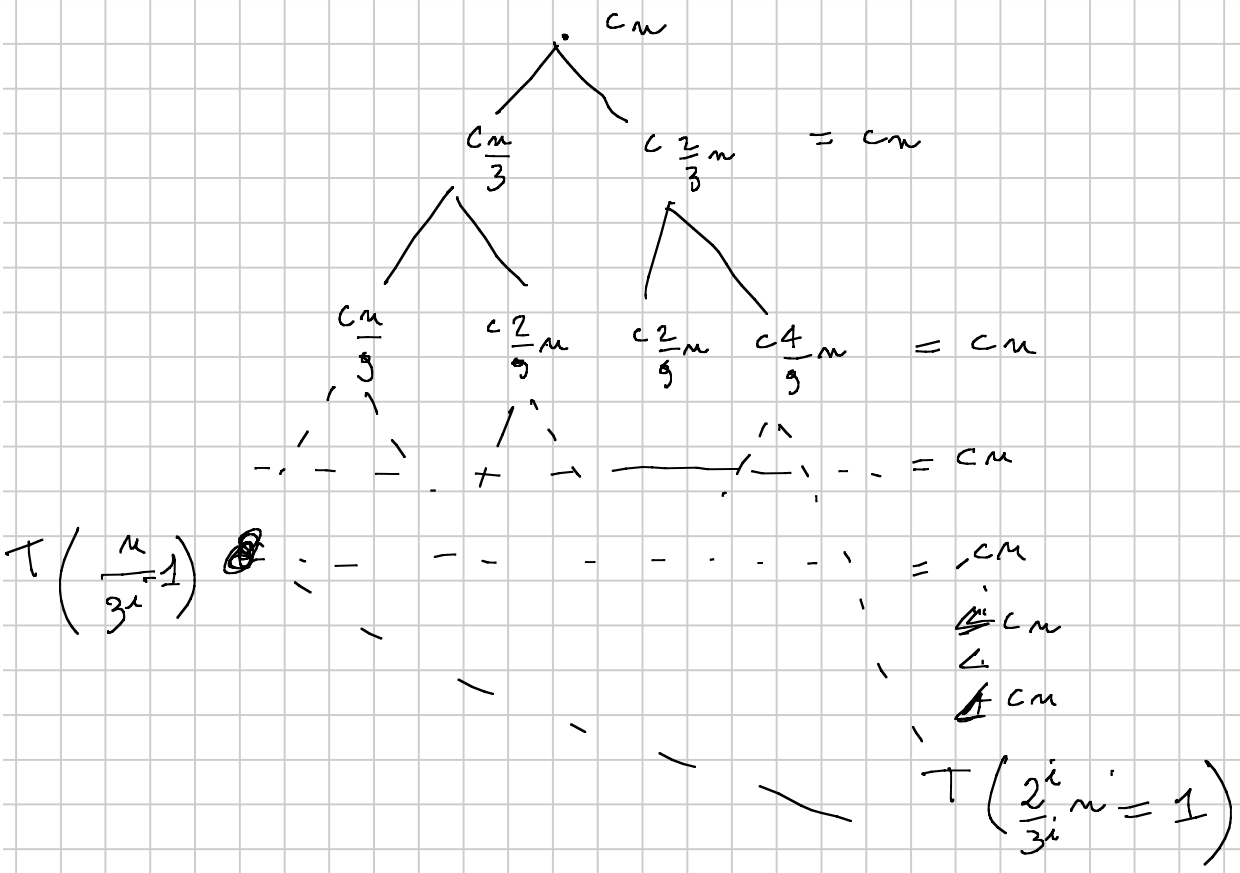
$a = 2 \qquad b = \frac{3}{2} \qquad cn$

$cn : \; n^{\log_{3/2} 2} \qquad n^1$

$f(n) = O\left(n^{1.5+\varepsilon}\right) \qquad \text{con } 1$

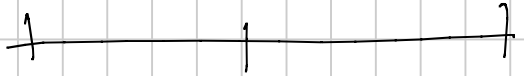
$$T(n) = O\left(n^{\log_{3/2} 2}\right)$$

$$T'(n) = \Theta\left(n^{\log_{3/2} 2}\right)$$



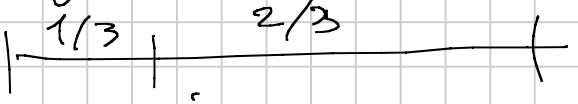
$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2}{3}n\right) + cn$$

$$T(n) = O\left(n \log_{3/2} n\right)$$



MergeSort su sottosezioni di n/2

MergeSort



$$\left(\frac{2}{3}\right)^i n = 1 \qquad n = \left(\frac{3}{2}\right)^i$$

$$i = \log_{3/2} n$$

Titolo nota 10/03/2016

$$T(n) = 4T\left(\frac{n}{2}\right)$$

✔

Merge Sort

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

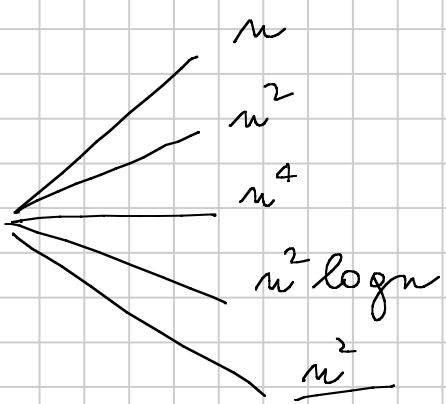
$a = 2 \quad b = 2 \quad n^{\log_b a} = n$

$f(n) = n : n \Rightarrow f(n) = \Theta(n^{\log_b a})$

caso 2: $T(n) = \Theta(n^{\log_b a} \cdot \log n) = \Theta(n \log n)$

if $p < r$
:
MERGESORT
MERGESORT
MERGE $\Theta(n)$

$T(n) = 4\,T\left(\frac{n}{2}\right) + f(n)$



$T(n) = n$

$T(n) = n^2 \log n$

$a = 4 \quad b = 2 \quad n^{\log_b a} = n^2$

$f(n) = n$

$n : n^2 ?$

$n = O(n^{2-\varepsilon}) \qquad 0 < \varepsilon \leq 1$

caso 1: $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$

$f(n) = n^2$

$n^2 : n^2 ?$

$f(n) = \Theta(n^{\log_b a})$ caso 2 $T(n) = \Theta(n^2 \log n)$

$f(n) = n^4$ $\quad n^2 : n^4$? $\quad n^4 = \Omega(n^{2+\varepsilon}) \quad 0 < \varepsilon \leq 2$

$T(n) = \Theta(n^4)$

se $\quad a f\left(\frac{n}{b}\right) \leq c f(n) \quad c < 1$

$4 f\left(\frac{n}{2}\right) \leq c f(n)$

$4 \left(\frac{n}{2}\right)^4 \leq c n^4$

$\frac{1}{4} n^4 \leq c n^4 \qquad c \geq \frac{1}{4}$ vero

Titolo nota 10/03/2016

$f(n) = n^2 \log n$

$f(n) = n^2 \log n \; ; n^2$

NON SI APPLICA Teo

caso 3? no

$f(n) = \dfrac{n^2}{\log n}$

$\dfrac{n^2}{\log n} \; ; n^2$

caso 1? no

Titolo nota 10/03/2016

Es. 4

Progettare algoritmo Divide et Impera per calcolare a^n con $O(\log n)$ moltiplicazioni.

Hint: $a^n = (a^{n/2})^2$ se n è pari oppure $(a^{n/2})^2 * a$ se n è dispari

$n \geq 1$

```
Potenza (a, n):
if (n == 1) return a;
  else{
  if (n%2) == 0 return Potenza(a, n/2) * Potenza(a, n/2)
          else return Potenza(a, n/2) . a;
  }
```

$$M(n) = \begin{cases} 0 & n = 1 \\ 2M\left(\frac{n}{2}\right) + 1 \end{cases}$$

Teorema

$a = 2$, $b = 2$ $\quad n^{\log_b a} = n \quad f(n) = \Theta(1)$

$\Theta(1) = f(n) = O\left(n^{1-\varepsilon}\right)$ per $\varepsilon \leq 1$

caso 1) $\quad T(n) = \Theta(n)$

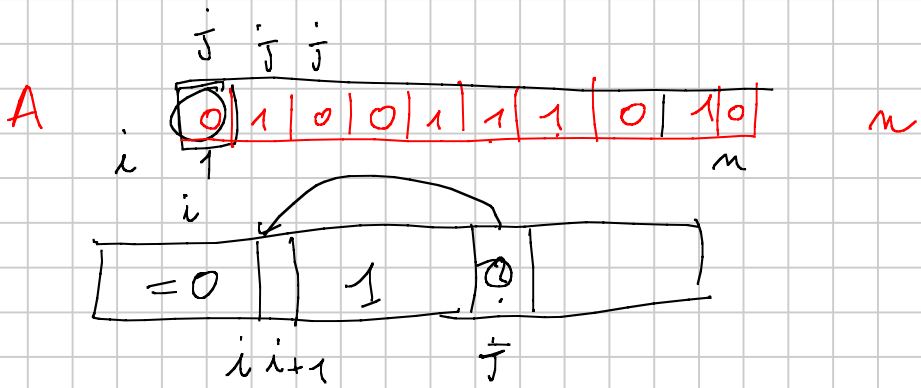
$$\begin{cases} T(n) = T\left(\frac{n}{2}\right) + \Theta(1) & \text{Ricerca Binaria} \quad \text{Sol: } T(n) = \Theta(\log n) \\ T(n) = 2T\left(\frac{n}{2}\right) + \Theta(1) & \text{Potenza} \quad \text{sol: } T(n) = \Theta(n) \\ T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n) & \text{Merge Sort} \quad \text{sol: } T(n) = \Theta(n \log n) \end{cases}$$

```
Power (a, n)
if (n == 1) return a
else {
p = power (a, n/2);
if (n%2 == 0)   return p*p;
        else    return p*p*a;
}
```

$$M(n) = M(n/2) + \Theta(1)$$

$$M(n) = \Theta(\log n)$$

Es: 5



ORDINA (A)

$i = 0$;

for ($j = 1$; $j \leq n$; $j++$) {

  if (A[j] == 0) {

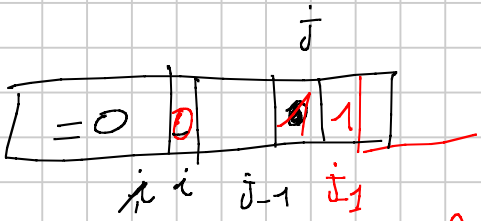
    $i++$; "Scambio" A[i] e A[j];

  }

}

$\Theta(1)$

all'inizio $i = 0$ $j = 1$



$T(n) = \Theta(n)$ tempo

$S(n) = \Theta(n)$ scambi