ESERCITAZIONE   Teorema Principale

$$T(n) = \begin{cases} c & n \leq 1 \\ 9T(n/3) + n & n > 1 \end{cases}$$

$$a^{\log_b n} = n^{\log_b a}$$

$a = 9 \qquad b = 3$

$$n^{\log_b a} = n^{\log_3 9} = n^2$$

$$f(n) = n^1 = O(n^{2-\varepsilon}) \quad \text{vero per } 0 < \varepsilon < 1$$

$1 < 2 - \varepsilon \Downarrow \varepsilon < 1$

I° caso $\qquad T(n) = \Theta(n^2)$

$$T(n) = \begin{cases} O(1) & n \leq 1 \\ 27T(n/3) + \Theta(n^3 \log n) & n > 1 \end{cases}$$

$a = 27, \quad b = 3, \qquad n^{\log_3 27} = n^3$

$f(n) = \Theta(n^3 \log n)$ $\qquad f(n) = \Theta(n^{\log_b a} \log^k n)$ $\quad \boxed{k \geq 0}$

caso 2, con $k = 1$

$$T(n) = \Theta(n^3 \log^2 n)$$

$$T(n) = 5\,T\left(\frac{n}{2}\right) + \Theta(n^3)$$

$a = 5, \quad b = 2, \quad n^{\log_b a} = n^{\log_2 5}$ $\qquad 2 < \log_2 5 < 3$

$f(n) = \Theta(n^3) = \Omega\left(n^{\log_2 5 + \varepsilon}\right)$ $\qquad$ ok per $0 < \varepsilon < \underbrace{3 - \log_2 5}_{>0}$

$3 \geq \log_2 5 + \varepsilon \implies 0 < \varepsilon \leq 3 - \log_2 5$

Condizione di regolarità

$$a\,f\left(\frac{n}{b}\right) \leq c\,f(n) \qquad c < 1 \text{ costante}$$

$$5\left(\frac{n}{2}\right)^3 = \frac{5}{8}n^3 \leq \frac{5}{8}f(n)$$ verificato per $c = \frac{5}{8} < 1$ ✓

$\implies$ III° caso $\qquad T(n) = \Theta(n^3)$

$$T(n) = 27\,T\left(\frac{n}{3}\right) + \Theta\left(\frac{n^3}{\log n}\right)$$

$a = 27, \quad b = 3, \quad \log_b a = 3 \qquad n^{\log_b a} = n^3$

$n^{\log_b a} = n^3$ cresce più velocemente di $\dfrac{n^3}{\log n}$

ma non per un fattore polinomiale

il I° caso non si applica. $\qquad$ NO 1

$$f(n) = \Theta(n^3 \log n^{-1}) = \Theta(n^{\log_b a} \cdot \log n^{-1})$$

$k = -1$

NO 2

(Per il 2° caso, deve essere $k \geq 0$)

Moltiplicazione veloce di interi di lunghezza arbitraria

A, B interi di n cifre decimali, memorizzati in due array di dimensione n
(ogni elemento dell'array memorizza una cifra decimale)

l'algoritmo elementare ha costo $\Theta(n^2)$
(n^2 moltiplicazioni di cifre + n addizioni di interi di lunghezza n)

Algoritmo Divide et Impera

supponiamo che n sia una potenza di 2
(altrimenti si aggiungono 0 a sinistra)

12345 $\rightarrow$ 00012345

$$A = \overbrace{\underbrace{A_1}_{n/2} \mid \underbrace{A_2}_{n/2}}^{n} \qquad B = \overbrace{\underbrace{B_1}_{n/2} \mid \underbrace{B_2}_{n/2}}^{n}$$

$n = 1 \rightarrow$ return $A * B$ (A e B hanno una sola cifra)

$$A = A_1 * 10^{n/2} + A_2 \qquad B = B_1 * 10^{n/2} + B_2$$

$$A = \underbrace{2340}_{A_1} \mid \underbrace{2185}_{A_2} = 2340 \cdot 10^4 + 2185$$

$$A * B = (A_1 \cdot 10^{n/2} + A_2)(B_1 \cdot 10^{n/2} + B_2) =$$

$$A_1 \cdot B_1 \cdot 10^n + (A_1 \cdot B_2 + A_2 \cdot B_1) \cdot 10^{n/2} + A_2 B_2$$

MOLT (A, B, n)

if (n==1) return A*B; $O(1)$

else {

divide A e B in A_1, A_2, B_1, B_2 → $\Theta(n)$

$X = MOLT(A_1, B_1, n/2);$

$Y = MOLT(A_2, B_2, n/2);$ $\Big\} 4T(\frac{n}{2})$

$Z = MOLT(A_1, B_2, n/2) + MOLT(A_2, B_1, n/2)$

return $X \cdot 10^n + Z \cdot 10^{n/2} + Y$ } $\Theta(n)$

}

$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ 4T(\frac{n}{2}) + \Theta(n) & n > 1 \end{cases}$$

$a = 4$, $b = 2$, $n^{\log_2 4} = n^2$

$f(n) = \Theta(n^1) = O(n^{2-\varepsilon})$ $1 < 2 - \varepsilon \Rightarrow \varepsilon < 1$

ok per $0 < \varepsilon < 1$

I° caso:

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

Nuovo tentativo: osserviamo che

$A_1B_2 + A_2B_1 = (A_1 + A_2) \cdot (B_1 + B_2) - A_1B_1 - A_2B_2$

$\cancel{A_1B_1} + A_1B_2 + A_2B_1 + \cancel{A_2B_2} - \cancel{A_1B_1} - \cancel{A_2B_2}$ ✓

$X = A_1 \cdot B_1$

$Y = A_2 \cdot B_2$ $Z = A_1B_2 + A_2B_1 = (A_1 + A_2)(B_1 + B_2) - X - Y$

$\underline{\text{MOLT2}(A, B, n)}$

if $(n==1)$ return $A*B$; $\qquad O(1)$

else {

dividi A e B in A_1, A_2, B_1, B_2 $\qquad \Theta(n)$

$3T(\frac{n}{2})$ $\begin{cases} X = \text{MOLT2}(A_1, B_1, n/2) \\ Y = \text{MOLT2}(A_2, B_2, n/2) \\ Z = \text{MOLT2}(A_1+A_2, B_1+B_2, n/2) - X - Y \end{cases}$

return $X \cdot 10^n + Z \cdot 10^{n/2} + Y$ $\qquad \Theta(n)$

}

$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ 3T(n/2) + \Theta(n) & n > 1 \end{cases}$$

$a = 3 \qquad b = 2 \qquad n^{\log_b a} = n^{\log_2 3} = n^{1.585\ldots}$

$\underline{f(n) = \Theta(n) \overset{?}{=} O(n^{\log_2 3 - \varepsilon})}$ $\qquad 0 < \varepsilon < \log_2 3 - 1$

$\varepsilon = 0.1$

$1 < \log_2 3 - \varepsilon$

Teorema principale

1° caso $\qquad T(n) = \Theta(n^{\log_2 3}) =$

$\Theta(n^{1.585\ldots})$ ✓

Sviluppando questa idea e spezzando i numeri in parti più piccole si ottiene

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

ESERCIZIO

Scrivere e risolvere la relazione di ricorrenza che esprime la complessità in tempo della funzione

```
Test (n)
 if (n < 10) return 5;          } Θ(1)
 a = 0;
 for ( i = 0; i < n; i++) {          → si ripete n volte
     for ( j = 0; j < n/2; j++) {    si ripete n/2 volte
          for ( k = 0; k < 42; k++) {   → si ripete 42 volte
                    a++;  } Θ(1)
     }     }
 }
 return 2 * Test (n/2) + 2 * a;   } T(n/2)
```

$\underbrace{42 * n * \frac{n}{2} * \Theta(1)}_{=}$

$\Theta(n^2)$

$\frac{42}{2} n^2$ cresce come n^2

$$T(n) = \begin{cases} \Theta(1) & n < 10 \\ T(n/2) + \Theta(n^2) \end{cases}$$

$a = 1 \qquad b = 2 \qquad n^{\log_b a} = n^{\log_2 1} = n^0 = 1$

$f(n) = \Theta(n^2) = \Omega(n^{0+\varepsilon}) \qquad 0 < \varepsilon \leq 2$

$\underbrace{2 \geq 0 + \varepsilon}$

$\underline{\varepsilon \leq 2}$

<u>Condizione di regolarità</u>

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

n sufficientemente grande

$\left(\frac{n}{2}\right)^2 \overset{?}{\leq} c \cdot n^2$

$\frac{n^2}{4} \leq \frac{1}{4} n^2$ ok con $c = \frac{1}{4} < 1$ ✓

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

Mistero (n)

```
if (n < 10) return 1;                              Θ(1)
x = Mistero(⌊n/4⌋) + Mistero(⌊n/4⌋);               2T(n/4)
i = 1;
while (i < n) {                   ──▷ si ripete log_3 n volte
      j = 1;
      while (j < n) {             ──▷ si ripete n/5 volte
                 j = j + 5;
      }
      i = 3 * i;
}
y = Mistero(⌊n/4⌋);                                T(n/4)
return x * y;                                      Θ(1)
```

$\Theta\left(\frac{n}{5} \cdot \log_3 n\right) \rightarrow \Theta(n \log n)$

$$T(n) = \begin{cases} \Theta(1) & n \leq 10 \\ 3T(n/4) + \Theta(n \log n) & n > 10 \end{cases}$$

$a = 3$, $b = 4$ $\qquad n^{\log_b a} = n^{\log_4 3}$

$$0 < \log_4 3 < 1$$

$$f(n) = \Theta(n^{1} \log n) = \Omega\left(n^{\log_4 3 + \varepsilon}\right)$$

$$1 \geq \log_4 3 + \varepsilon \Rightarrow 0 < \varepsilon \leq \underbrace{1 - \log_4 3}_{>0}$$

3° caso.

Condizione di regolarità:

$$\underbrace{3\left(\frac{n}{4}\log\frac{n}{4}\right)}_{af\left(\frac{n}{5}\right)} = \frac{3}{4}n\cdot\log\frac{n}{4} \le \frac{3}{4}n\log n = \frac{3}{4}f(n)$$

ok con

$c = 3/4 < 1$ ✓

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

Evitando le 3 chiamate identiche:

$$T(n) = T(n/4) + \underline{\Theta(n\log n)}$$

$a = 1,\ b = 4 \qquad n^{\log_4 1} = n^0 = 1$

Ancora 3° caso

```
F(n)
  if (n < 10) return n;
  if (n < 30) return n^2;
  x = n;
  j = x;
  while (j > 1) {
        x = x + j;
        j = j/2;
  }
  return x * F(⌊n/3⌋) * F(⌊n/3⌋);
```

$\Theta(1)$

$O(1)$

si ripete $\log_2 n$ volte $\rightarrow \Theta(\log n)$

$2T(n/3)$

```
z = F(⌊n/3⌋);
return x * z * z
```

$$T(n) = \begin{cases} \Theta(1) & n < 30 \\ 2T\left(\frac{n}{3}\right) + \Theta(\log n) & n \geq 30 \end{cases}$$

$a = 2, \quad b = 3, \quad n^{\log_b a} = n^{\log_3 2}$

$\log n = o(n^{\varepsilon}) \quad \forall \varepsilon > 0$

$0 < \log_3 2 < 1$

$f(n) = \Theta(n^0 \cdot \log n) = O\left(n^{\log_3 2 - \varepsilon}\right)$ ✓

$\log_3 2 - \varepsilon > 0$

$\underline{0 < \varepsilon < \log_3 2}$

1° caso $\quad T(n) = \Theta\left(n^{\log_3 2}\right)$

col miglioramento:

$$T(n) = \begin{cases} \Theta(1) & n < 30 \\ T\left(\frac{n}{3}\right) + \Theta(\log n) & n \geq 30 \end{cases}$$

$a = 1, \quad b = 3, \quad n^{\log_b a} = n^{\log_3 1} = n^0 = 1$

$f(n) = \Theta(1 \cdot \log n)$

II° caso con $k = 1$

$\Big\}\ T(n) = \Theta(\log^2 n)$