Algoritmi e Strutture Dati

Lezione 10

15 ottobre 2025

Tecnica "divide–et–impera"

$\mathcal{P}$ Problema

$\mathcal{I}$ Istanza di $\mathcal{P}$

- Se $\mathcal{I}$ è piccola risolvi $\mathcal{P}$ su $\mathcal{I}$ direttamente

  altrimenti

  - dividi $\mathcal{I}$ in istanze di lunghezza minore della lunghezza di $\mathcal{I}$
  - risolvi queste istanze separatamete
  - ricava la soluzione di $\mathcal{I}$ combinando opportunamente le soluzioni ottenute

ALGORITMO risolviP(istanza I) → soluzione

IF $|I| \leq C$ THEN

risolvi P su I direttamente

RETURN soluzione

← caso base

ELSE

dividi I in m istanze $I_1, I_2 \dots, I_m$ con $|I_j| < |I|$ $j = 1 \dots m$

$sol_1 \leftarrow$ risolviP(I_1)

$sol_2 \leftarrow$ risolviP(I_2)

⋮

$sol_m \leftarrow$ risolviP(I_m)

← ricorsione

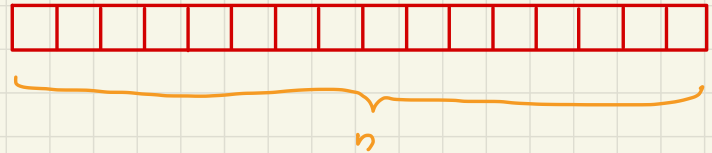
RETURN combina ($sol_1, sol_2 \dots sol_m$)

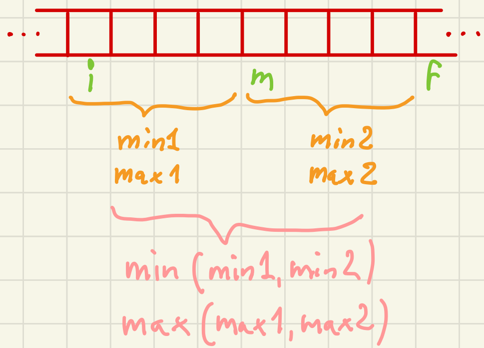
$$T(I) = \begin{cases} \text{costante} & \text{se } |I| \leq C \\ T_{divide}(I) + T(I_1) + T(I_2) + \dots + T(I_m) + T_{combin}(sol_1, \dots, sol_m) & \text{altrimenti} \end{cases}$$

ESEMPIO: dato array $A[0..n-1]$, $n>0$
trovare l'elemento minimo e l'elemento massimo

```
ALGORITMO minMax (array A[0..n-1]) → (elemento, elemento)
    min ← A[0]
    max ← A[0]
    FOR i ← 1 TO n-1 DO
        IF A[i] < min THEN min ← A[i]
        IF A[i] > max THEN max ← A[i]
    RETURN (min, max)
```

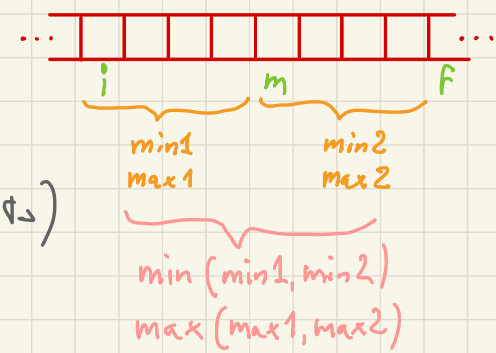
#cfr $2n-2$


n


i
m
f
min1
max1
min2
max2
min (min1, min2)
max (max1, max2)

```
FUNZIONE minMax (array A, indice i, indice f) → (elemento, elemento)

    IF f - i = 1 THEN RETURN (A[i], A[i])                 casi
    ELSE IF f - i = 2 THEN                                base
        IF A[i] < A[i+1] THEN RETURN (A[i], A[i+1])
                         ELSE RETURN (A[i+1], A[i])
    ELSE
        m ← (i+f)/2                                       ricorsione
        (min1, max1) ← minMax (A, i, m)
        (min2, max2) ← minMax (A, m, f)
        IF min1 < min2 THEN min ← min1
                       ELSE min ← min2
        IF max1 > max2 THEN max ← max1
                       ELSE max ← max2
        RETURN (min, max)
```




```
ALGORITMO minMax
    (array A[0..n-1])
        → (elemento, elemento)

    RETURN minMax (A, 0, n)
```

```
FUNZIONE minMax (array A, indice i, indice f) → (elemento, elemento)
  IF f-i = 1 THEN RETURN (A[i], A[i])        ← casi
  ELSE IF f-i = 2 THEN                        ← base
     IF A[i] < A[i+1] THEN RETURN (A[i], A[i+1])
                      ELSE RETURN (A[i+1], A[i])
  ELSE
     m ← (i+f)/2
     (min1, max1) ← minMax(A, i, m)           ← ricorsioni
     (min2, max2) ← minMax(A, m, f)
     IF min1 < min2 THEN min ← min1
                    ELSE min ← min2
     IF max1 > max2 THEN max ← max1
                    ELSE max ← max2
  RETURN (min, max)
```

$$C(n) = \begin{cases} 0 & \text{se } n=1 \\ 1 & \text{se } n=2 \\ \underbrace{C(\lfloor \frac{n}{2} \rfloor) + C(\lceil \frac{n}{2} \rceil) + 2}_{2C(\frac{n}{2})+2} & \text{altrimenti} \end{cases}$$

n pari: $2C(\frac{n}{2})+2$

per n potenza di 2: $C(n) = 2C(\frac{n}{2}) + 2 = 2\left[2C(\frac{n}{2^2})+2\right]+2 =$

$2^2 C(\frac{n}{2^2}) + 2^2 + 2 = 2^2\left[2C(\frac{n}{2^3})+2\right] + 2^2 + 2 = 2^3 C(\frac{n}{2^3}) + 2^3 + 2^2 + 2$

$\dots \; 2^k C(\frac{n}{2^k}) + \sum_{i=1}^{k} 2^i = 2^k C(\frac{n}{2^k}) + 2^{k+1} - 2$

$\frac{n}{2^k} = 2 \qquad n = 2^{k+1}$

$k = \log_2 n - 1$

$= \frac{n}{2} \underbrace{C(2)}_{1} + n - 2 = \frac{3}{2}n - 2$ ← per n potenza di 2

Per n non potenza di 2: $n = 2^j + n' \qquad n' < 2^j$

si ottiene $C(n) = \lceil \frac{3}{2} n \rceil - 2$

Equazioni di ricorrenza

$$F(n) = \begin{cases} b' & n = 1 \\ m\,F\left(\frac{n}{a}\right) + b'' n^c & \text{altrimenti} \end{cases}$$

$m, a, b', b'', c \in \mathbb{R}^+$

$a > 1$

$$\downarrow$$

$$F(n) = \begin{cases} \Theta(n^c) & \text{se } m < a^c \\ \Theta(n^c \lg n) & \text{se } m = a^c \\ \Theta(n^{\lg_a m}) & \text{se } m > a^c \end{cases}$$

Equazioni di ricorrenza: teorema fondamentale

(Master Theorem)

Siano $m, a, b', b'', c \in \mathbb{R}^+$ con $a > 1$. L'equazione

$$F(n) = \begin{cases} b' & \text{se } n = 1 \\ mF\left(\frac{n}{a}\right) + b'' n^c & \text{se } n > 1 \end{cases}$$

soddisfa le seguenti relazioni:

$$F(n) = \begin{cases} \Theta(n^c) & \text{se } m < a^c \\ \Theta(n^c \log n) & \text{se } m = a^c \\ \Theta(n^{log_a m}) & \text{se } m > a^c \end{cases}$$

Il risultato può essere esteso anche al caso in cui per $n > 1$ l'equazione data sia della forma

$$F(n) = m_1 F\left(\left\lfloor \frac{n}{a} \right\rfloor\right) + m_2 F\left(\left\lceil \frac{n}{a} \right\rceil\right) + b'' n^c$$

con $m = m_1 + m_2$

Master theorem: esempi

$$H(n) = \begin{cases} 1 & \text{se } n = 1 \\ H\left(\frac{n}{2}\right) + 1 & \text{altrimenti} \end{cases}$$

$m = 1$

$a = 2$

$c = 0$

$a^c = 1$

$H(n) = \Theta(\lg n)$

Siano $m, a, b', b'', c \in \mathbb{R}^+$ con $a > 1$. L'equazione

$$F(n) = \begin{cases} b' & \text{se } n = 1 \\ mF\left(\frac{n}{a}\right) + b'' n^c & \text{se } n > 1 \end{cases}$$

soddisfa le seguenti relazioni:

$$F(n) = \begin{cases} \Theta(n^c) & \text{se } m < a^c \\ \Theta(n^c \log n) & \text{se } m = a^c \\ \Theta(n^{\log_a m}) & \text{se } m > a^c \end{cases}$$

Master theorem: esempi

$$T(n) = \begin{cases} 1 & \text{se } n=1 \\ 2T\left(\frac{n}{2}\right)+bn & \text{altrimenti} \end{cases}$$

$m=2$

$a=2$

$c=1$

$a^c=2$

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

Siano $m, a, b', b'', c \in \mathbb{R}^+$ con $a > 1$. L'equazione

$$F(n) = \begin{cases} b' & \text{se } n = 1 \\ mF\left(\frac{n}{a}\right) + b''n^c & \text{se } n > 1 \end{cases}$$

soddisfa le seguenti relazioni:

$$F(n) = \begin{cases} \Theta(n^c) & \text{se } m < a^c \\ \Theta(n^c \log n) & \text{se } m = a^c \\ \Theta(n^{\log_a m}) & \text{se } m > a^c \end{cases}$$

Master theorem: esempi

$$F(n) = \begin{cases} 1 & \text{se } n = 1 \\ 8F\left(\frac{n}{2}\right) + n^2 & \text{se } n > 1 \end{cases}$$

$m = 8$

$a = 2$ $\qquad a^c = 4$

$c = 2$

$$F(n) = \Theta\left(n^{\log_2 8}\right) = \Theta(n^3)$$

Siano $m, a, b', b'', c \in \mathbb{R}^+$ con $a > 1$. L'equazione

$$F(n) = \begin{cases} b' & \text{se } n = 1 \\ mF\left(\frac{n}{a}\right) + b''n^c & \text{se } n > 1 \end{cases}$$

soddisfa le seguenti relazioni:

$$F(n) = \begin{cases} \Theta(n^c) & \text{se } m < a^c \\ \Theta(n^c \log n) & \text{se } m = a^c \\ \Theta(n^{\log_a m}) & \text{se } m > a^c \end{cases}$$

Master theorem: esempi

$$G(n) = \begin{cases} 1 & \text{se } n=1 \\ 7G\left(\frac{n}{2}\right) + 6n^2 & \text{se } n>1 \end{cases}$$

$m = 7$

$a = 2$ $\qquad a^c = 4$

$c = 2$

$$G(n) = \Theta\left(n^{\log_2 7}\right) = \Theta(n^{2.81\ldots})$$

Siano $m, a, b', b'', c \in \mathbb{R}^+$ con $a > 1$. L'equazione

$$F(n) = \begin{cases} b' & \text{se } n = 1 \\ mF\left(\frac{n}{a}\right) + b'' n^c & \text{se } n > 1 \end{cases}$$

soddisfa le seguenti relazioni:

$$F(n) = \begin{cases} \Theta(n^c) & \text{se } m < a^c \\ \Theta(n^c \log n) & \text{se } m = a^c \\ \Theta(n^{\log_a m}) & \text{se } m > a^c \end{cases}$$

Esempio: somma di matrici $n \times n$

Quante operazioni "elementari" sono necessarie e sufficienti per calcolare la somma di 2 matrici $n \times n$ di interi?

Input $A = [a_{ij}]$, $B = [b_{ij}]$, matrici $n \times n$ di interi

Output $C = [c_{ij}] = A + B$ somma delle matrici A e B

Definizione di *somma di matrici*: $c_{ij} = a_{ij} + b_{ij}$

1 somma per ogni posizione

in totale n^2 somme

Esempio: prodotto di matrici $n \times n$

$A = (\text{———})$ $B = (\; | \;)$

Input $A = [a_{ij}]$, $B = [b_{ij}]$, matrici $n \times n$ di interi

Output $C = [c_{ij}] = A \cdot B$ prodotto delle matrici A e B

Definizione di *prodotto di matrici*: $c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$

n prodotti
n-1 somme
2n-1 oper.
per ciascun elemento

tot op.: $n^2(2n-1) = 2n^3 - n^2 = \Theta(n^3)$

Upper bound $\Theta(n^3)$

Lower bound $\Omega(n^2)$

$$A \quad \left(\begin{array}{cc|cc} 1 & 4 & 7 & 8 \\ 3 & 2 & 10 & 15 \\ \hline 2 & 1 & 20 & 0 \\ 5 & 2 & 1 & 0 \end{array}\right)$$

$$\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \qquad \begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}$$

$$A \cdot B = C$$

$$C = A \cdot B = \begin{array}{c|c} A_{11} B_{11} + A_{12} B_{21} & A_{11} B_{12} + A_{12} B_{22} \\ \hline A_{21} B_{11} + A_{22} B_{21} & A_{21} B_{12} + A_{22} B_{22} \end{array}$$

Prodotto di matrici

Divide-et-impera "immediato"

Se $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ e $B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$

con A_{ij}, B_{ij} matrici $\frac{n}{2} \times \frac{n}{2}$, allora

$$C = A \cdot B = \begin{array}{cc} C_{11} & C_{12} \\ \begin{bmatrix} A_{11} \cdot B_{11} + A_{12} \cdot B_{21} & A_{11} \cdot B_{12} + A_{12} \cdot B_{22} \\ A_{21} \cdot B_{11} + A_{22} \cdot B_{21} & A_{21} \cdot B_{12} + A_{22} \cdot B_{22} \end{bmatrix} \\ C_{21} & C_{22} \end{array}$$

$T(n)$ = # operazioni per moltiplicare due matrici $n \times 2$

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 8\,T\left(\frac{n}{2}\right) + n^2 & \text{altrimenti} \end{cases}$$

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

se $n = 1$: $C = [a_{11} \cdot b_{11}]$ ← caso base ← 1 operazione

se $n > 1$ (potenza di 2) calcola ricorsivamente:

$C_{11} = A_{11} \cdot B_{11} + A_{12} \cdot B_{21}$

$C_{12} = \ldots$

$C_{21} = \ldots$

C_{22}

ricorsione

8 prodotti di matrici $\frac{n}{2} \times \frac{n}{2}$ $\quad 8\,T\left(\frac{n}{2}\right)$

4 somme di matrici $\frac{n}{2} \times \frac{n}{2}$ $\quad 4\left(\frac{n}{2}\right)^2 = n^2$

costo somma matrici $\frac{n}{2} \times \frac{n}{2}$

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

Prodotto di matrici $n \times n$ – Algoritmo di Strassen

$T(n) =$ n° operazioni per moltiplicare matrici $n \times n$

caso base $n = 1$ $\rightarrow$ 1 operazione

Date $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ e $B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$

Considero le matrici:

- $M_1 = (A_{21} + A_{22} - A_{11}) \cdot (B_{22} - B_{12} + B_{11})$
- $M_2 = A_{11} \cdot B_{11}$
- $M_3 = A_{12} \cdot B_{21}$
- $M_4 = (A_{11} - A_{21}) \cdot (B_{22} - B_{11})$
- $M_5 = (A_{21} + A_{22}) \cdot (B_{12} - B_{11})$
- $M_6 = (A_{12} - A_{21} + A_{11} - A_{22}) \cdot B_{22}$
- $M_7 = A_{22} \cdot (B_{11} + B_{22} - B_{12} - B_{21})$

7 prodotti di matrici $\frac{n}{2} \times \frac{n}{2}$ $\quad 7\,T\left(\frac{n}{2}\right)$

24 somme/sottrazioni di matrici $\frac{n}{2} \times \frac{n}{2}$ $\quad 24\left(\frac{n}{2}\right)^2 = 6n^2$

Si può verificare che:

$$C = A \cdot B = \begin{bmatrix} M_2 + M_3 & M_1 + M_2 + M_5 + M_6 \\ M_1 + M_2 + M_4 - M_7 & M_1 + M_2 + M_4 + M_5 \end{bmatrix}$$

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 7\,T\left(\frac{n}{2}\right) + 6n^2 & \text{altrimenti} \end{cases}$$

$$T(n) = \Theta\left(n^{\log_2 7}\right) = \Theta\left(n^{2.81}\right)$$