Prodotto interno di vettori

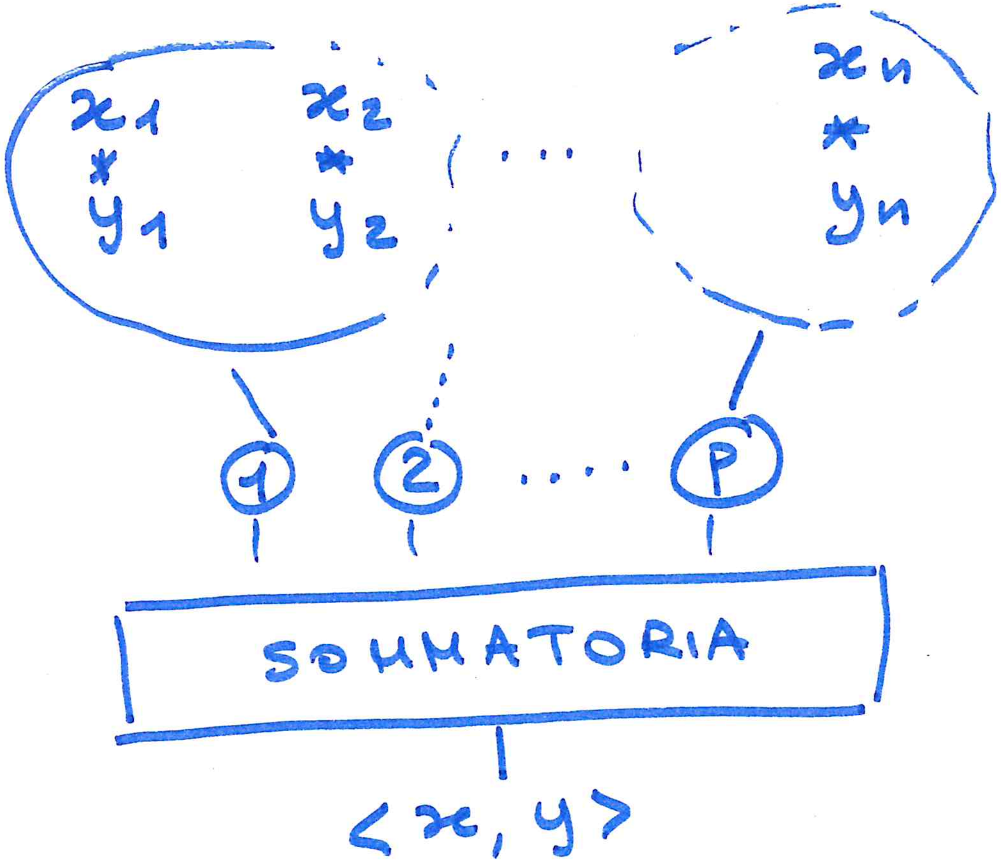
Input: $x, y \in \mathbb{N}^n$

Output: $\langle x, y \rangle = \sum_{i=1}^{n} x_i * y_i$

Tempo sequenziale

$2n - 1$

Soluzione EREW:



I fase:

I° passo Δ prodotti in sequenza per processore e somma sequenziale

II fase:

somma di p numeri in parallelo

Per sommatoria:

$p = c_1 \frac{n}{\log n}$ $\quad t_{II} = c_2 \log n \quad \Big\}$ II fase

$p \sim \frac{n}{\log n} \Rightarrow \Delta = \frac{n}{p} = \log n \Rightarrow t_I = c_3 \log n \quad \Big\}$ I fase

<u>Riassumendo</u>:

$\langle x, y \rangle$ costa: $p \sim \frac{n}{\log n}$ $\quad t = t_I + t_{II} \sim \log n$

$E \sim \frac{2n-1}{\frac{n}{\log n} \cdot \log n} \rightarrow C \neq 0$ costante

Prodotto matrice vettore

Input: $A \in \mathbb{N}^{n \times n}$, $x \in \mathbb{N}^{n}$

Output: Ax

Tempo sequenz.

$n(2n-1) =$
$= 2n^2 - n$

Idea: uso il modulo $\langle \dots, \dots \rangle$ in parallelo n-volte

Attenzione: il vettore x è acceduto simultaneamente dai moduli $\langle \dots, \dots \rangle \Rightarrow$ CREW

Prestazioni

$p(n) \sim n \cdot \frac{n}{\log n}$ $\quad T(n, p(n)) \sim \log n$

$$E(n, p(n)) \sim \frac{n^2}{\frac{n^2}{\log n} \cdot \log n} \longrightarrow C \neq 0_{\text{COSTANTE}}$$

Prodotto MATRICE MATRICE

Input: $A, B \in \mathbb{N}^{n \times n}$

Output: $A \cdot B$

Tempo sequenziale

$n^{2,80}$

STRASSEN

n^2,37
Le Gall
2014

Idea: uso n^2 prodotti $\langle \dots, \dots \rangle$ in parallelo

Attenzione: ogni riga di A e colonna di B viene acceduta simult.

$\Rightarrow$ algoritmo CREW

Prestazioni:

$p(n) \approx n^2 \cdot \frac{n}{\log n}$ $\qquad T(n, p(n)) \approx \log n$

$$E(n, p(n)) \approx \frac{n^{2,8}}{\frac{n^3}{\cancel{\log n}} \cdot \cancel{\log n}} = \frac{n^{2,8}}{n^3} \to 0 \text{ lentamente}$$

Potenza di matrice

(*)
nota:
è un prodotto iterato
della stessa matrice M

Input: $A \in \mathbb{N}^{n \times n}$

Output: A^n, $n = 2^k$

sequenzialmente: (*)

for $i = 1$ to $\log n$ do
$A = A \cdot A$

Tempo sequenziale

$n^{2,80} \cdot \log n$

parallelo: $\log n$ volte il prodotto $A \cdot A$
(è il sequenziale dove sostituisco "·" tra le A da sequenziale a parallelo) $\Rightarrow$ CREW

prestazioni:

$P(n) = \frac{n^3}{\log n}$ $\qquad T(n, P(n)) = \log n \cdot \log n = \log^2 n$

$$E \sim \frac{n^{2,8} \cdot \log n}{\frac{n^3}{\log n} \cdot \log^2 n} = \frac{n^{2,8}}{n^3} \to 0$$

lentamente

SOMME PREFISSE

+ applicazioni

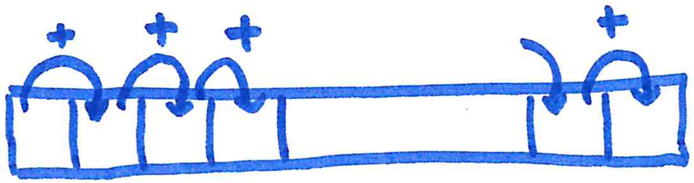
Problema delle somme prefisse

Input: $M[1], M[2], \ldots, M[n]$

Output: $\sum_{i=1}^{k} M[i] \to M[k]$ $\quad 1 \le k \le n$

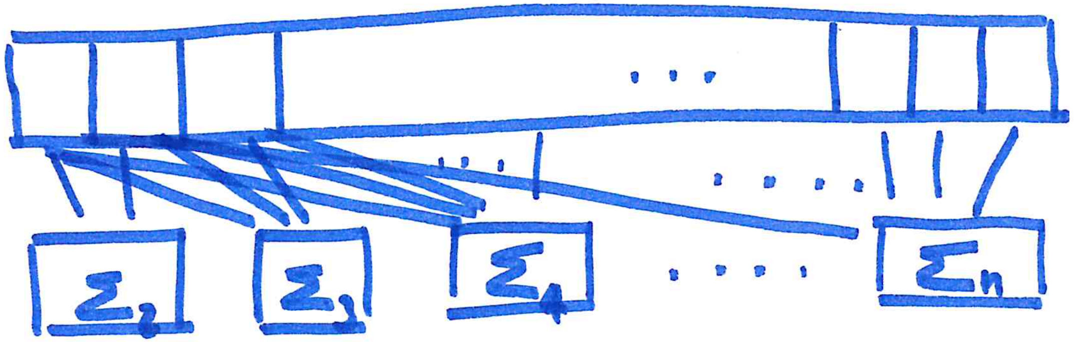
Assumiamo n potenza di 2

Algoritmo sequenziale:



	Tempo
for k = 2 to n do	
M[k] = M[k] + M[k-1];	n - 1

Una proposta parallela per SOMME PREFISSE



Risolvo con SOMMATORIA tutti i possibili prefissi.

Problemi:

1) non è EREW ma potrei risolvere il problema
2) Ho un CREW algoritmo su PRAM con:

$$p(n) \leq (n-1) \cdot \frac{n}{\log n} \sim \frac{n^2}{\log n}$$

$$= \sum_{i=2}^{n} i/\log i \geq \frac{1}{\log n} \sum_{i=2}^{n} i \sim \frac{n^2}{\log n}$$

$$T(n, p(n)) \sim \log n$$

$$E \sim \frac{n-1}{\frac{n^2}{\log n} \cdot \log n} \rightarrow 0$$

POCO EFFICIENTE

Nel '73 KOGGE-STONE usa il <u>pointer doubling</u>

<u>IDEA</u>: si stabiliscano dei legami tra i numeri
ogni processore si occupa di un legame
e ne fa la somma così:

posizioni: $\quad\quad\quad i \quad\quad\quad\quad j$

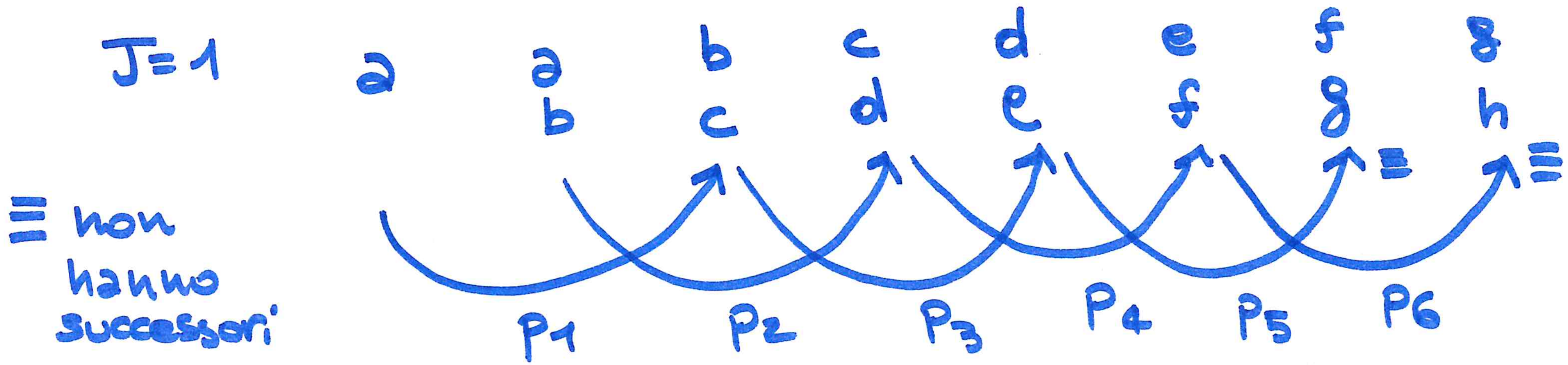
numeri: $\quad \dots \quad m \quad \dots \quad k \quad \dots$

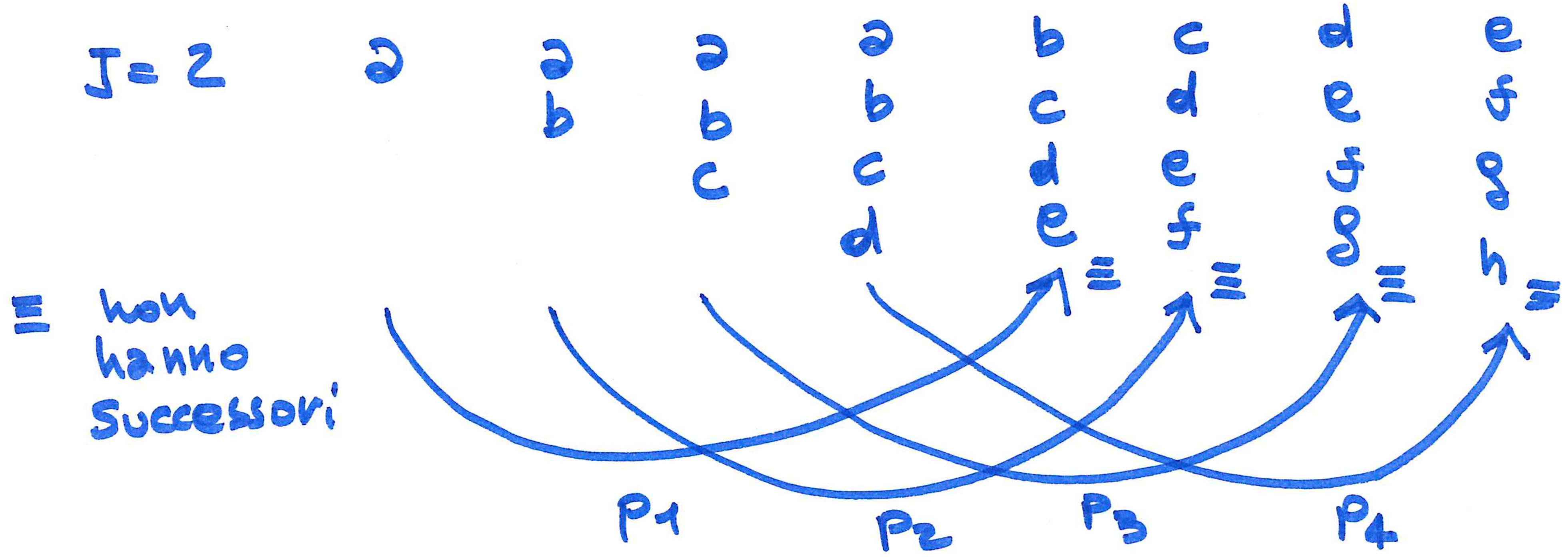
$$m \xrightarrow[\;i\;]{} k$$

i fa $m+k$
e mette il
risultato in
posizione j

<u>Esempio</u>:

$$a \xrightarrow[P_1]{} b \xrightarrow[P_2]{} c \xrightarrow[P_3]{} d \xrightarrow[P_4]{} e \xrightarrow[P_5]{} f \xrightarrow[P_6]{} g \xrightarrow[P_7]{} h$$


J=1
a
a b
b c
c d
d e
e f
f g
g h
≡ non hanno successori
P1
P2
P3
P4
P5
P6


J= 2
a
a b
a b c
a b c d
b c d e
c d e f
d e f g
e f g h
≡ non hanno successori
P1
P2
P3
P4

J = 3

a	a	a	a	a	a	a	a
	b	b	b	b	b	b	b
		c	c	c	c	c	c
			d	d	d	d	d
				e	e	e	e
					f	f	f
						g	g
							h
≡	≡	≡	≡	≡	≡	≡	≡

≡ non hanno successori

FINE!

OSSERVAZIONI:

1) nessun elemento ha un successore

2) ho risolto le <u>somme prefisse</u>