Calcolo Parallelo e Distribuito

A. Murli

Problema

Progettare un algoritmo parallelo per

l'ordinamento di un vettore

su un calcolatore MIMD a memoria

distribuita con p processori

Esempio:

- input

L= (25,7,1,9,81,3,28,12,6,20)

- output

L= (1,3,6,7,9,12,20,25,28,81)

(ordinamento crescente di 10 numeri interi)

Qual è l'algoritmo sequenziale?

- selection sort
- insertion sort
- bubble sort
- .....

$O(n) \leq T(n) \leq O(n^2)$ confronti

- Bitonic-merge sort

$O(n \log^2 n)$ confronti

- Quick sort
- Merge sort
- Heap sort

$O(n \log n) \leq T(n) \leq O(n^2)$ confronti

$O(n \log n)$ confronti

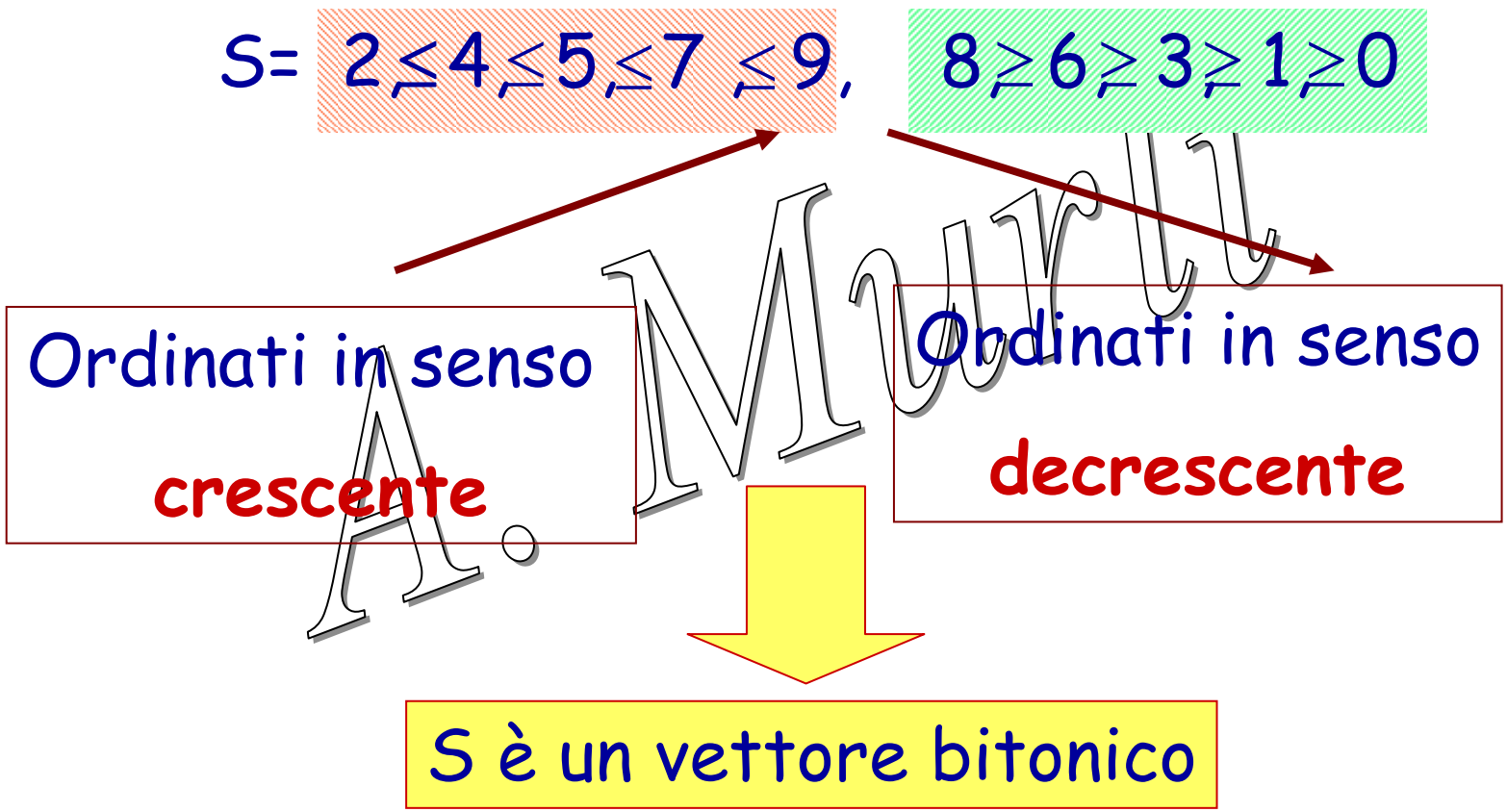
Punti da analizzare:

1. Algoritmo per l'ordinamento di un vettore **"bitonico" mediante il** Bitonic Sort

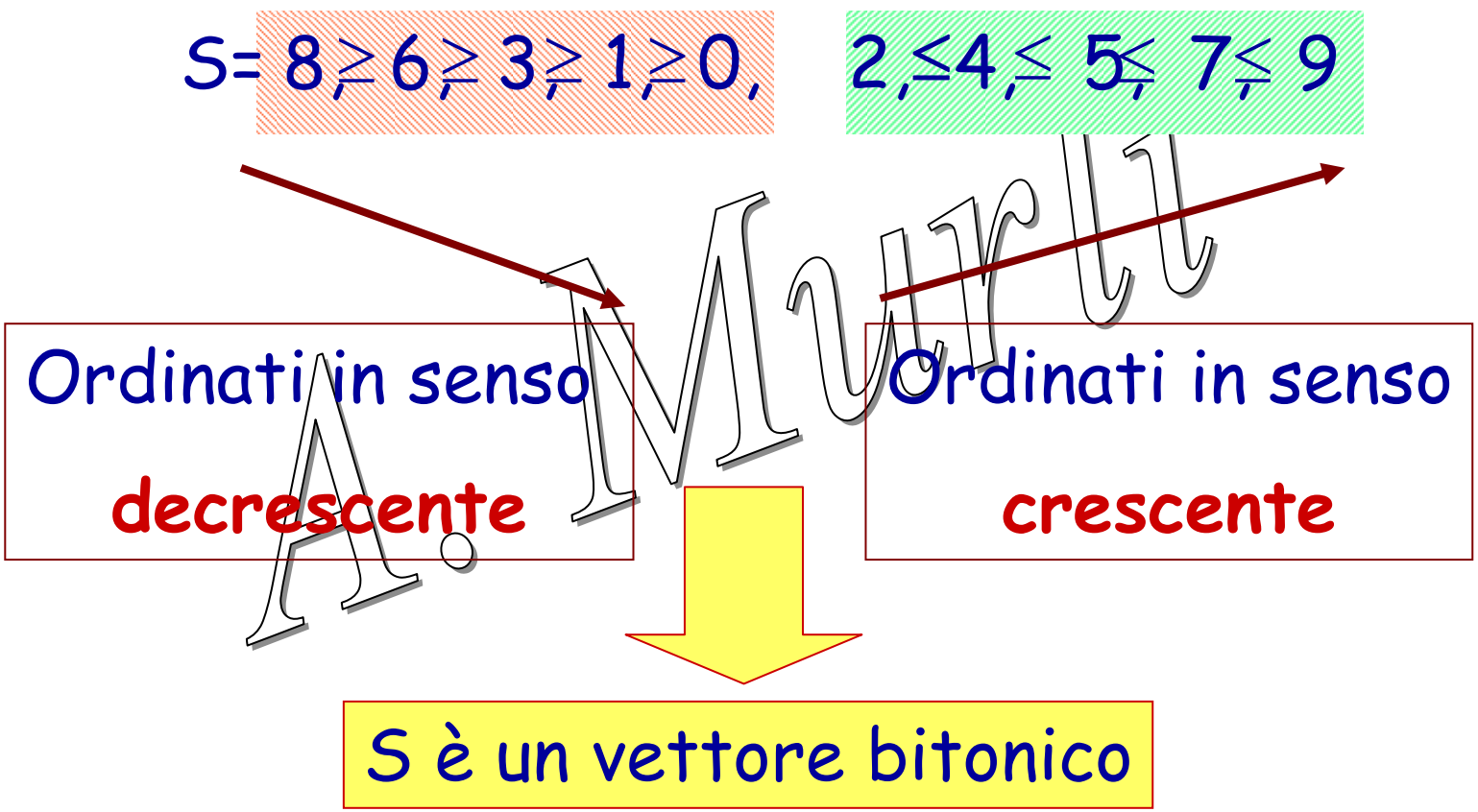
2. Algoritmo per l'ordinamento un vettore **qualsiasi mediante il** General Bitonic Sort

3. Algoritmo **Parallelo** per l'ordinamento di un **vettore qualsiasi** (Parallel General Bitonic Sort)

Cos’ è un vettore bitonico ?

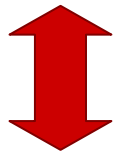


Cos' è un vettore bitonico ?



Cos' è un vettore bitonico ?

$S = s_1\, s_2\, s_3 \ldots\ldots s_n$ è un vettore bitonico



Definizione

Esiste un indice k $(1 \le k \le n)$ tale che

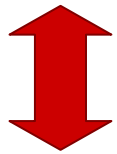
$S = s_1\, s_2\, s_3\, s_k \ldots\ldots s_n$

$s_1 \le s_2 \le s_3 \le \ldots\ldots \le s_k$ $\qquad$ $s_{k+1} \ge s_{k+2} \ge s_{k+3} \ge \ldots\ldots \ge s_n$

(se k=n il vettore si dice MONOTONICO)

Cos' è un vettore bitonico ?

$S = s_1\, s_2\, s_3 \ldots\ldots s_n$ è un vettore bitonico



Definizione

Esiste un indice k $(1 \le k \le n)$ tale che

$S = s_1\, s_2\, s_3\, s_k \ldots\ldots s_n$

$s_1 \ge s_2 \ge s_3 \ge \ldots\ldots \ge s_k$

$s_{k+1} \le s_{k+2} \le s_{k+3} \le \ldots\ldots \le s_n$

(se k=n il vettore si dice MONOTONICO)

Come ordinare un vettore bitonico?

BBS: IDEA

trasformare il vettore

di lunghezza $n = 2^q$

in un "insieme ordinato"

di coppie di elementi

Come ordinare un vettore bitonico?

BBS: IDEA

A. Murli

L'algoritmo si basa su un procedimento di tipo

Divide and conquer:

L'operazione di base (confronto – scambio)

viene applicata ricorsivamente ai sottovettori di lunghezza

uguale alla metà di quella precedente

Come ordinare un vettore bitonico?

BBS: IDEA

A partire dalle 2 metà (già ordinate)

Si confrontano gli elementi omologhi scambiando gli

elementi di ciascuna coppia se non si trovano nell'ordine desiderato.

Si ripete questo procedimento applicandolo separatamente a ciascuna metà.

Si prosegue fino a quando si arriva ad applicarlo a coppie di elementi.

BBS: Ordinamento di un vettore bitonico

ad ogni passo e per ogni coppia,
si effettuano operazioni di
compare-exchange

compare: confronto **exchange**: Scambio

BBS: Ordinamento di un vettore bitonico

due tipi di **compare-exchange**

Compare Exchange Low
(Co-Ex-Lo)

Se si vuole ordinare la coppia in senso crescente

(min, max)

Compare Exchange High
(Co-Ex-Hi)

Se si vuole ordinare la coppia in senso decrescente

(max, min)

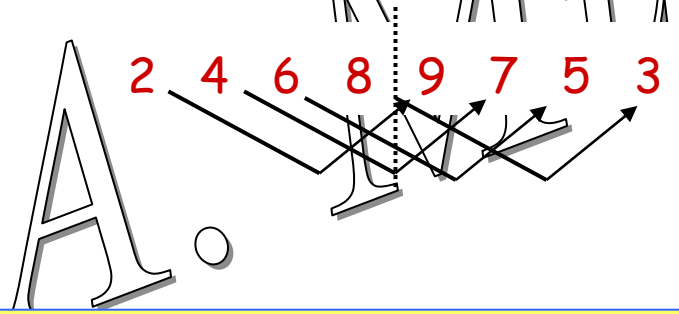
BBS: Ordinamento crescente di un vettore bitonico

Esempio: $n=8=2^3$

2 4 6 8 | 9 7 5 3

è un vettore bitonico

Passo 1 – si considerano i due sottovettori della I e della II metà
e si confrontano gli elementi omologhi



si spostano i più piccoli

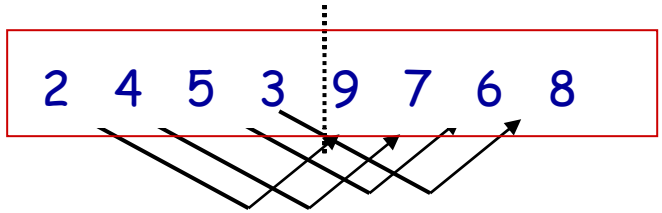
a sinistra e i più grandi a destra

CO-EX-LO

BBS:Ordinamento crescente di una lista bitonica

Applichiamo il Co-Ex-Lo

passo 1

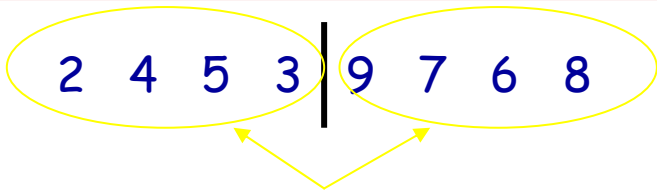


min (2, 9) = 2
max (2, 9) = 9
min (4, 7) = 4
max (4, 7) = 7
min (6, 5) = 5
max (6, 5) = 6
min (8, 3) = 3
max (8, 3) = 8

Murli

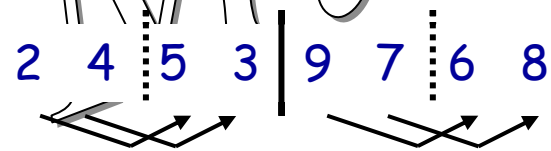
Il vettore iniziale è composto da due vettori bitonici

BBS: Ordinamento crescente di una lista bitonica



Passo 2 Si considerano i 2 sottovettori bitonici costruiti al Passo 1

Su ciascuno si applica il procedimento precedente.

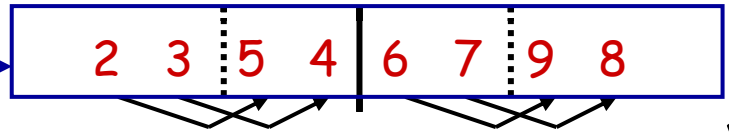


si spostano i più piccoli (di ciascuna metà)

a sinistra e i più grandi a destra

CO-EX-LO

BBS:Ordinamento crescente di una lista bitonica

Fine Passo 2 → 

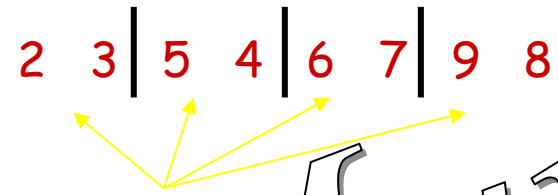
min (2, 5) = 2
max (2, 5) = 5
min (9, 6) = 6
max (9, 6) = 9
min (4, 3) = 3
max (4, 3) = 4
min (7, 8) = 7
max (7, 8) = 8

Murli

Il vettore iniziale è composto da 4 sottovettori bitonici

BBS:Ordinamento crescente di una lista bitonica

Passo 3 Si considerano i 4 sottovettori bitonici costruiti al Passo 2



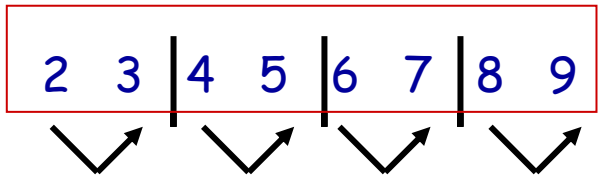
Su ciascuno si applica il procedimento precedente

si spostano i più piccoli (di ciascuna coppia)

a sinistra e i più grandi a destra

CO-EX-LO

BBS:Ordinamento crescente di una lista bitonica



min (2, 3) = 2
max (2, 3) = 3
min (5, 4) = 4
max (5, 4) = 5
min (6, 7) = 6
max (6, 7) = 7
min (9, 8) = 8
max (9, 8) = 9

Murli

Fine III passo

vettore ordinato

Qual è la complessità di tempo?

- L'algoritmo è costituito da $\log_2 n$ passi
- Ad ogni passo si effettuano n/2 Co-Ex

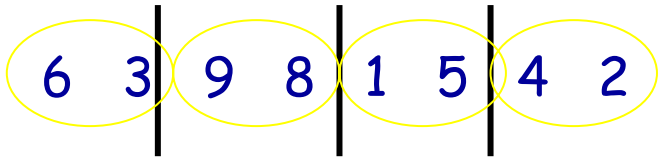
La complessità di tempo dell'algoritmo BBS è $O(n \log_2 n)$

Punti da analizzare:

1. Descrizione dello schema per l'ordinamento **Sequenziale** di una **vettore bitonico** (Bitonic Sort)
2. Descrizione dello schema per l'ordinamento **Sequenziale** di un **vettore qualsiasi** (General Bitonic Sort)
3. Descrizione dello schema per l'ordinamento **Parallelo** di un **vettore qualsiasi** (Parallel General Bitonic Sort)

Come ordinare un vettore qualsiasi ?

GBS: fase a

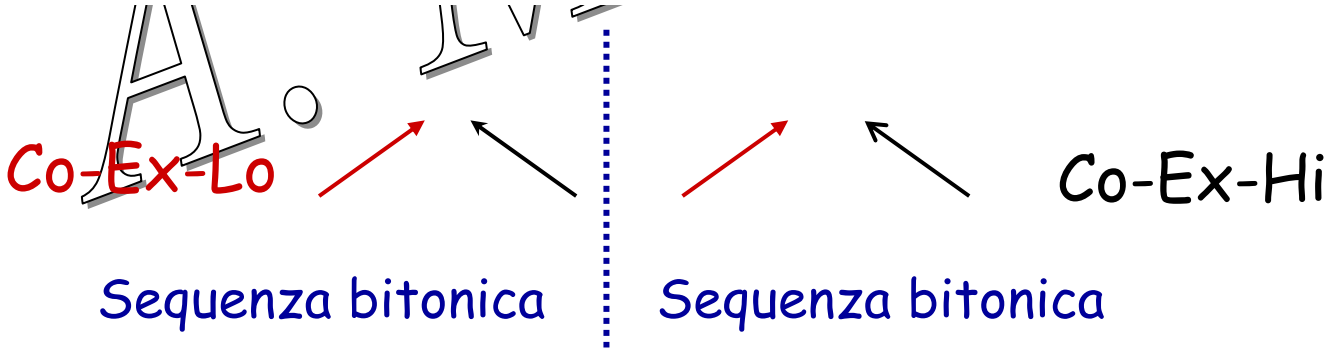


Partizioniamo il vettore in 4 coppie

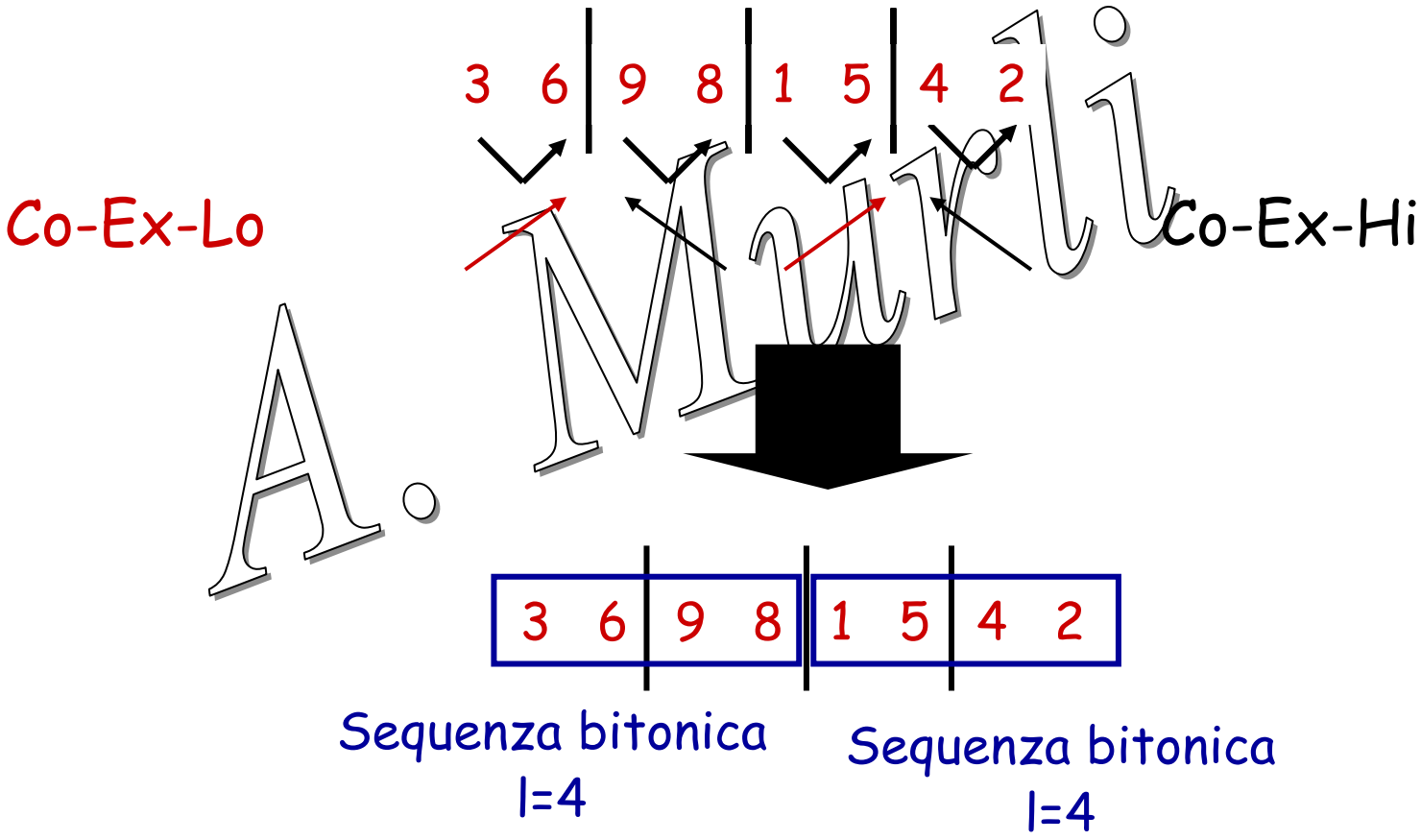
ovvero in 4 sottosequenze bitoniche di lungh = 2

e ordiniamole in modo da avere

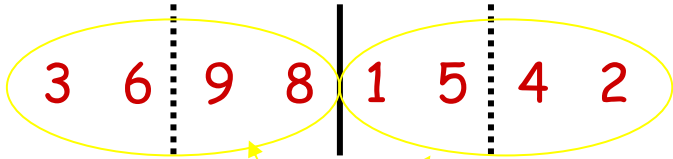
2 sottosequenze bitoniche di lungh = 4



GBS: fase a - STEP 1



GBS: fase a - STEP 2



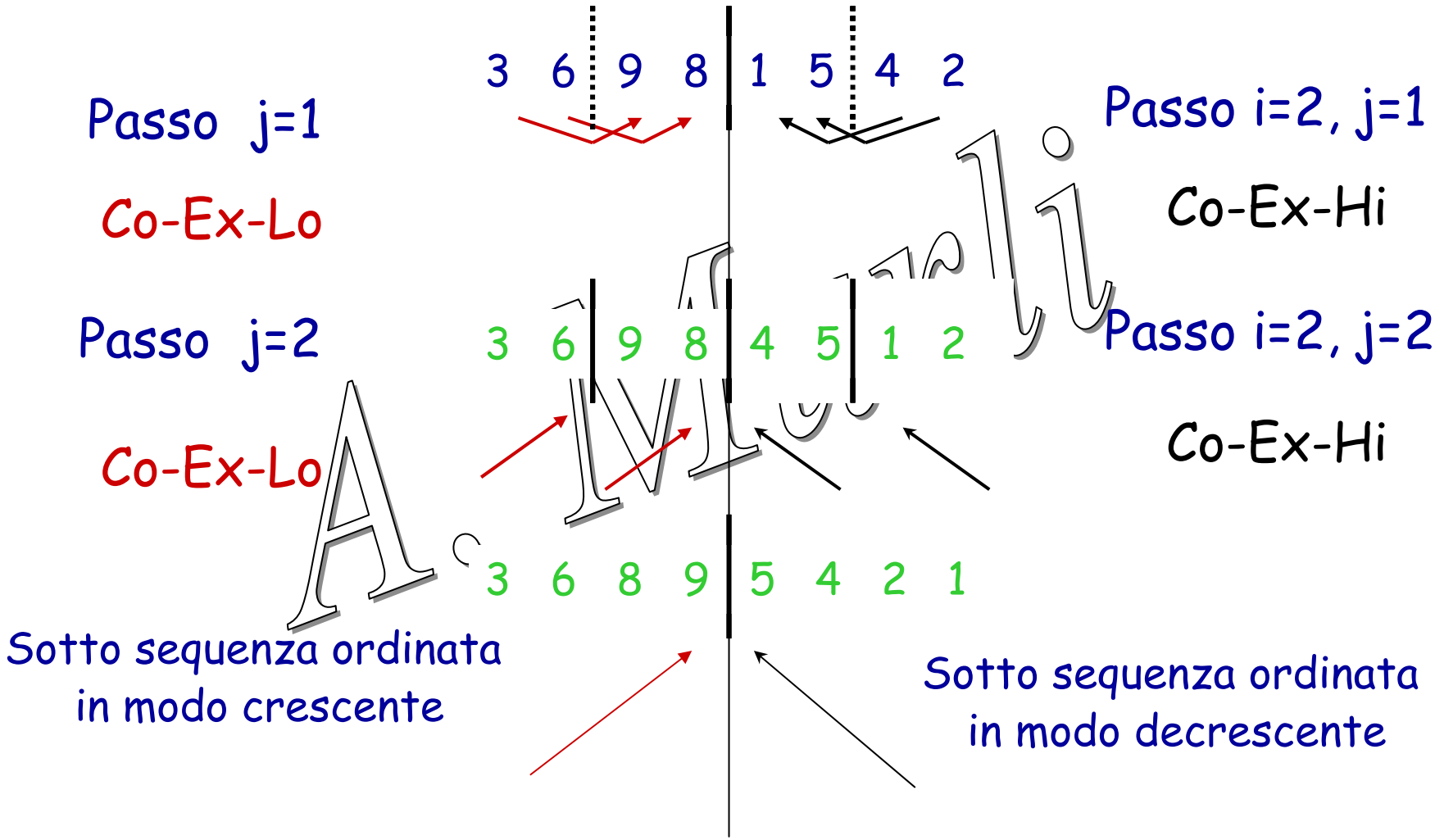
Ordiniamo ciascuno dei 2 sottovettori (con il BBS)

in modo da avere un unico vettore bitonico

Il primo in ordine crescente

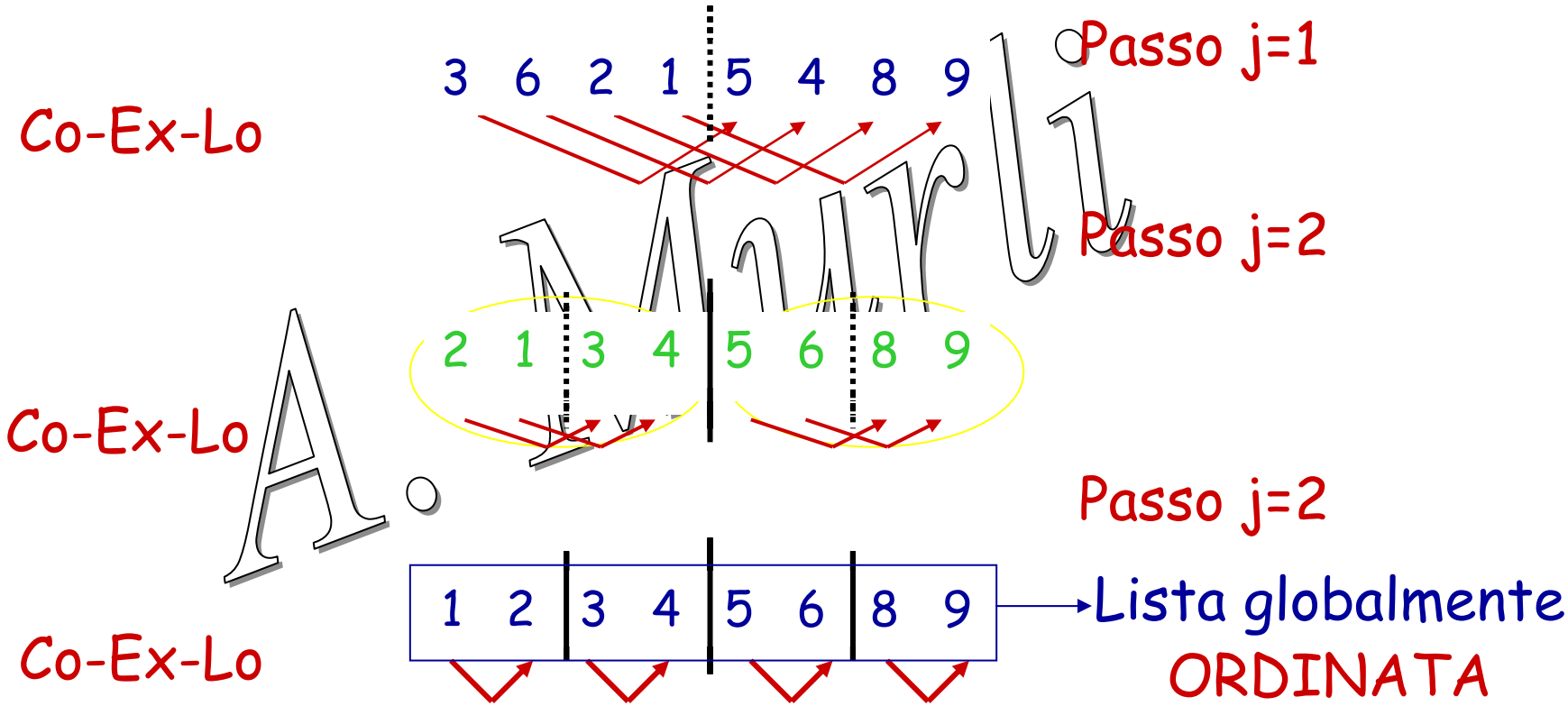
il secondo in ordine decrescente.

GBS: fase a - STEP 1



GBS: fase b - STEP 1

Applichiamo il BBS al vettore bitonico



Qual è la complessità di tempo?

- L'algoritmo è costituito da $\log_2 n$ passi
- Ad ogni passo si utilizza l'algoritmo BBS su sequenze di lunghezza minore di n, la cui complessità di tempo è $O(n \log_2 n)$

La complessità di tempo dell'algoritmo GBS è $O(n \log_2^2 n)$

Punti da analizzare:

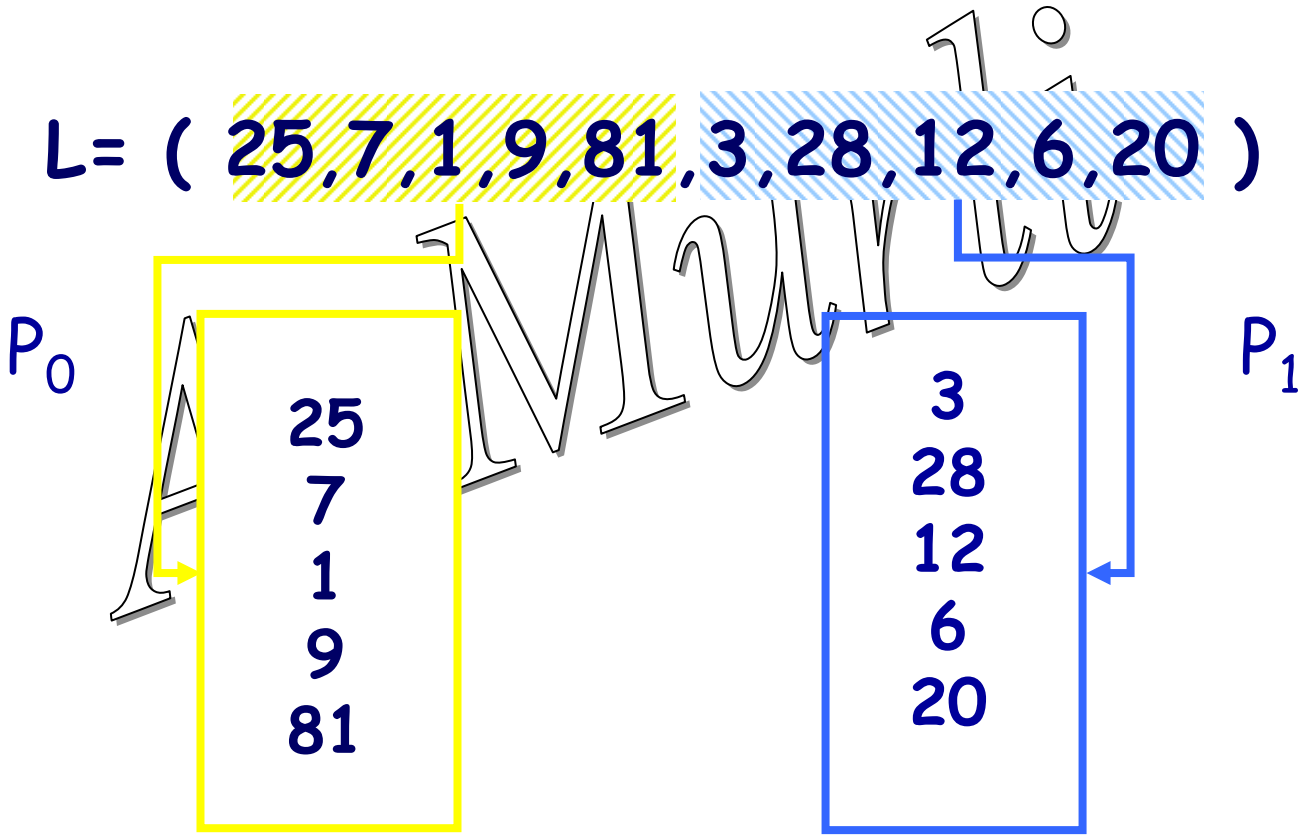
1. Descrizione dello schema per l'ordinamento **Sequenziale** di una **lista bitonica** (Bitonic Sort)

2. Descrizione dello schema per l'ordinamento **Sequenziale** di una **lista generica** (General Bitonic Sort)

3. Descrizione dello schema per l'ordinamento **Parallelo** di una **lista generica** (Parallel General Bitonic Sort)

Esempio: p=2

- Distribuiamo il vettore



PGBS:Ordinamento di un vettore

IDEA DI BASE DELL'ALGORITMO PARALLELO

l'operazione

di confronto – scambio

[che nell'algoritmo sequenziale opera

su una coppia di componenti **a < b**]

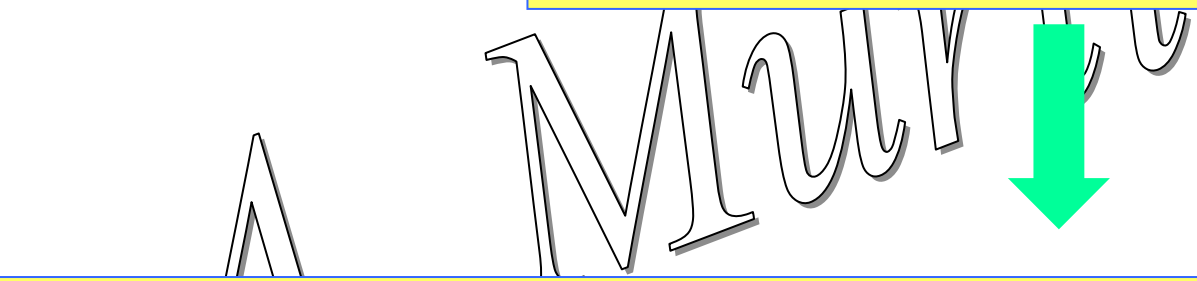
viene ora applicata

ad una coppia di processori

In che senso $P_0 < P_1$?

PGBS:Ordinamento di un vettore

$P_0 < P_1$ ⟺ Tutti gli elementi di P_0 sono minori di tutti gli elementi di P_1



Se gli elementi di P_0 e di P_1 sono ordinati localmente per effettuare il confronto $P_0 < P_1$ basta confrontare il più piccolo di P_0 con il più grande di P_1

Strategia generale

FASE 1 (SORTING LOCALE)

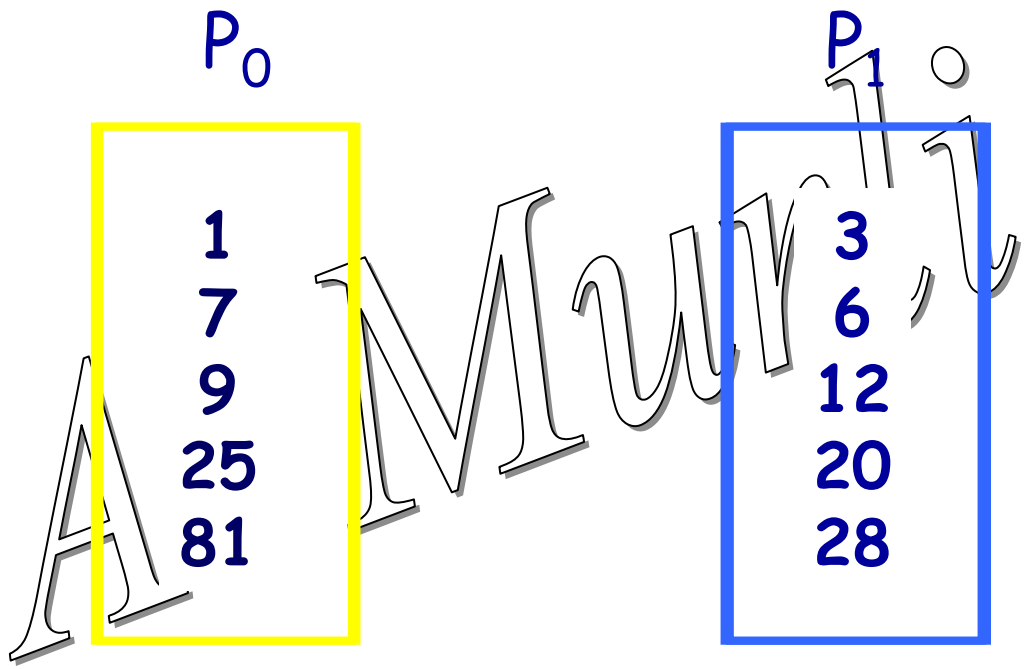
Ciascun processore ordina il proprio vettore utilizzando un **algoritmo di ordinamento**

FASE 2 (MERGING)

Si confrontano gli elementi in coppie di Processori

PGBS: Ordinamento di un vettore

FASE 1 (SORTING LOCALE)

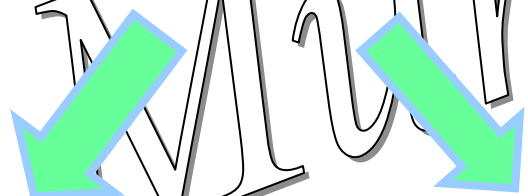


Ogni processore ordina in modo crescente il proprio vettore

PGBS:Ordinamento di un vettore

FASE 2 (MERGING)

l'operazione di
COMPARE EXCHANGE viene applicata ai 2 processori



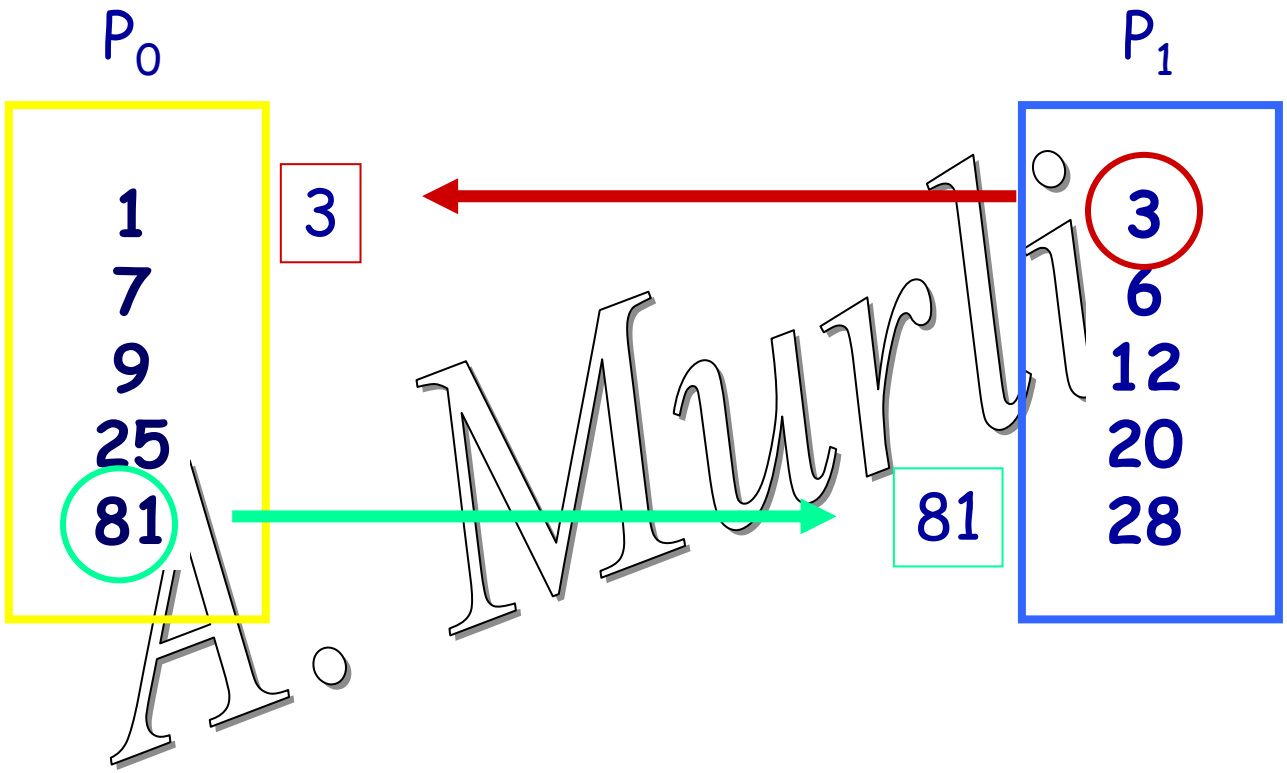
Il processore P_0 contiene gli elementi più piccoli

Processore "basso"

Il processore P_1 contiene gli elementi più grandi

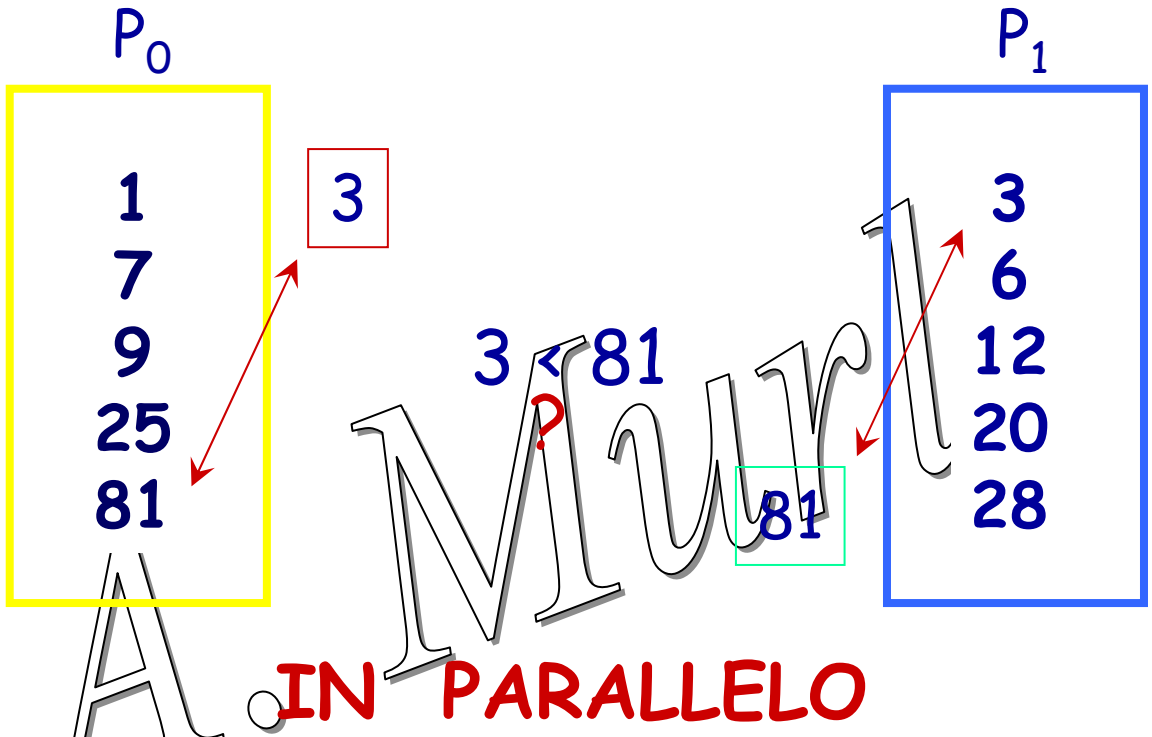
Processore "alto"

I PASSO: step 1.0



Step 1.0: P_0 invia a P_1 il proprio max = 81,
mentre P_1 invia a P_0 il proprio min = 3

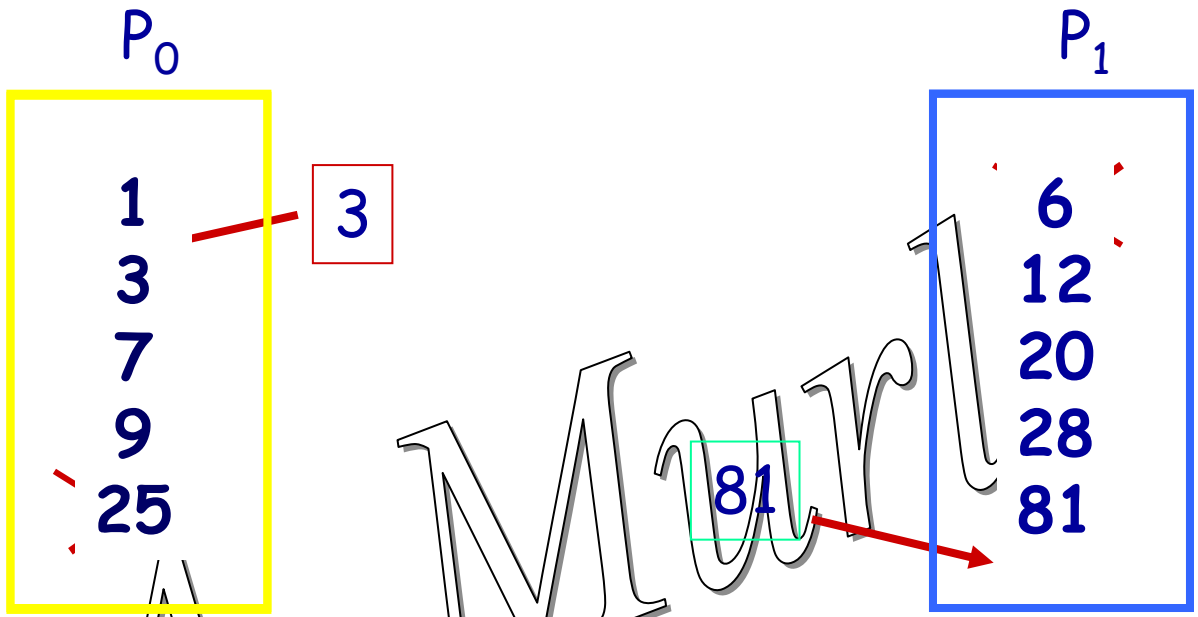
I PASSO : step 1.1



Step 1.1: P_0 confronta 3 con 81: se è minore lo inserisce;

P_1 confronta 81 con 3:se è maggiore lo inserisce

I PASSO: step 1.2

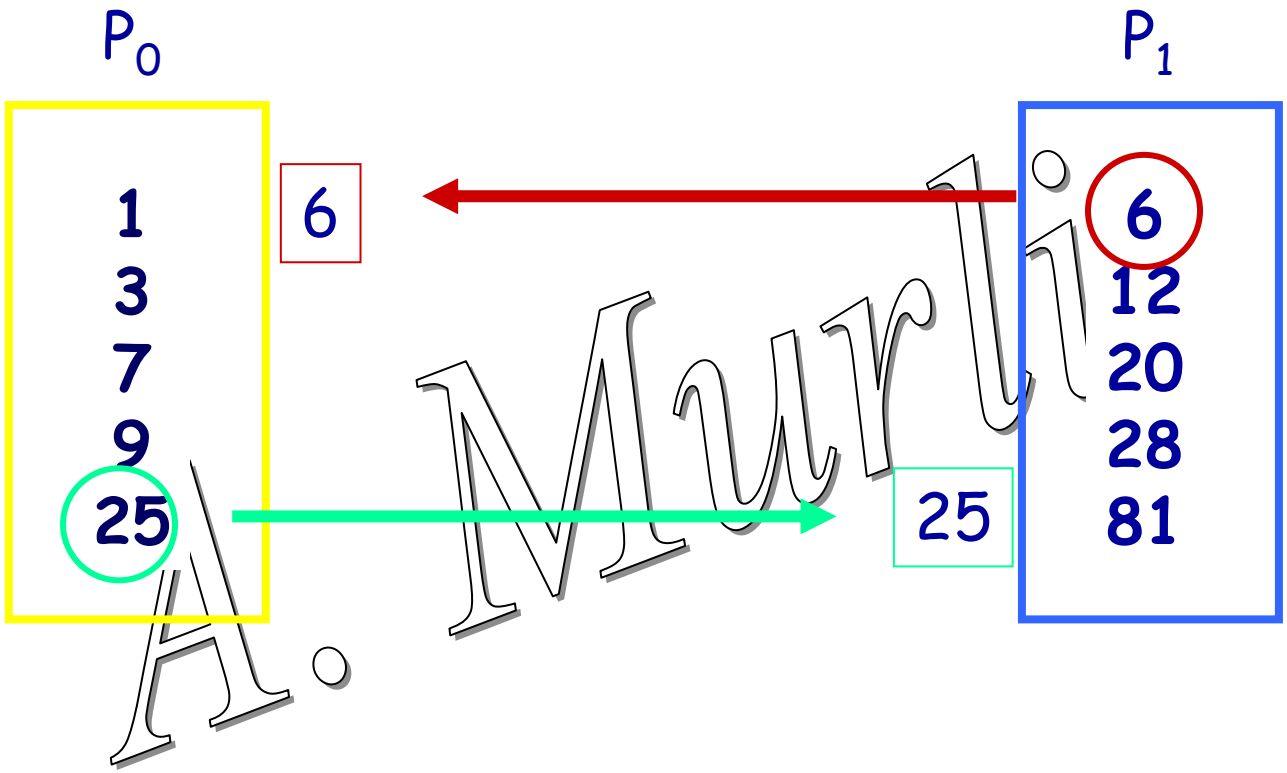


IN PARALLELO

Step 1.2: Ciascun processore

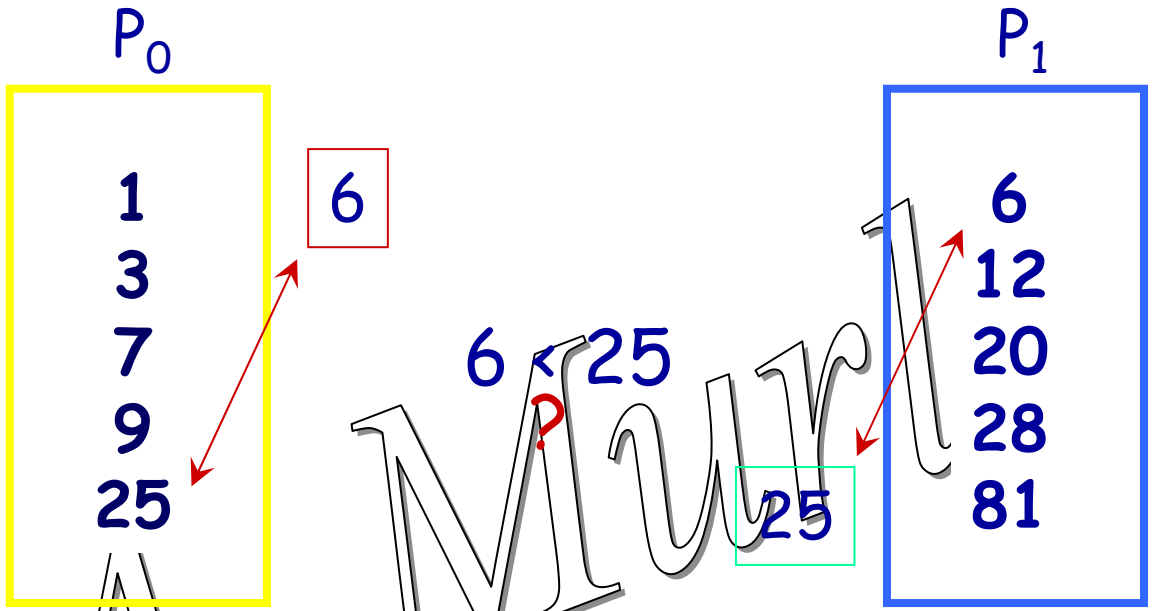
1) Inserisce il numero ricevuto nel proprio vettore
2) Elimina il numero inviato dal proprio vettore
3) Riordina il vettore

II PASSO: step 2.0



Step 2.0: P_0 invia a P_1 il proprio max = 25,
mentre P_1 invia a P_0 il proprio min = 6

II PASSO: step 2.1

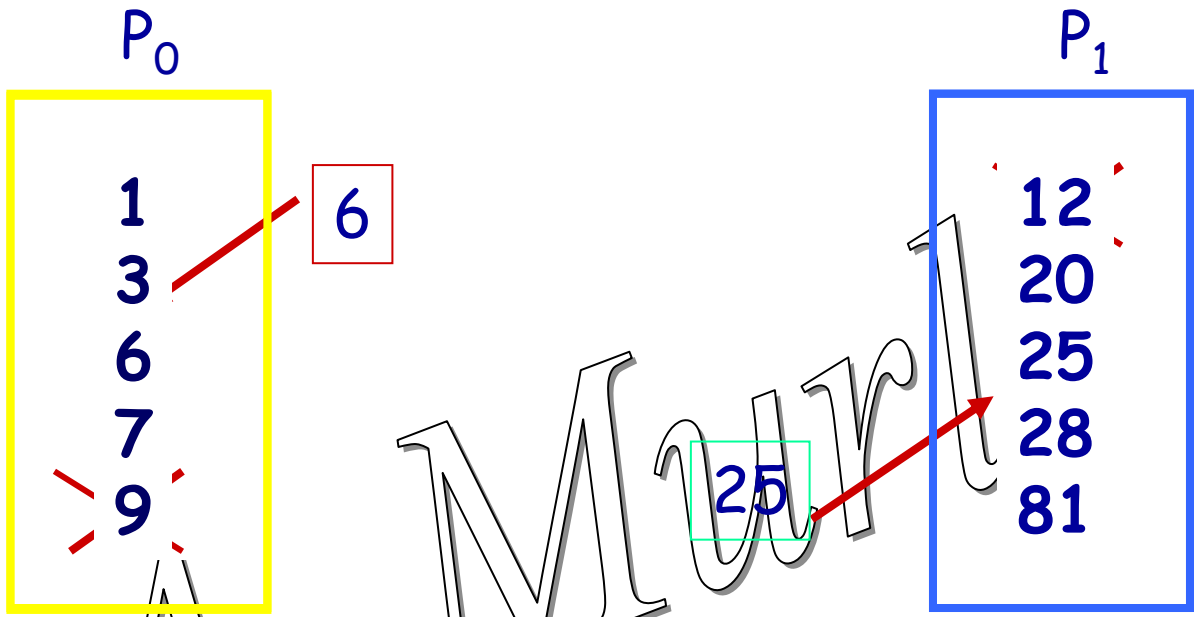


IN PARALLELO

Step 2.1: P_0 confronta 6 con 25: se è minore lo inserisce;

P_1 confronta 25 con 6: se è maggiore lo inserisce

II PASSO: step 2.2



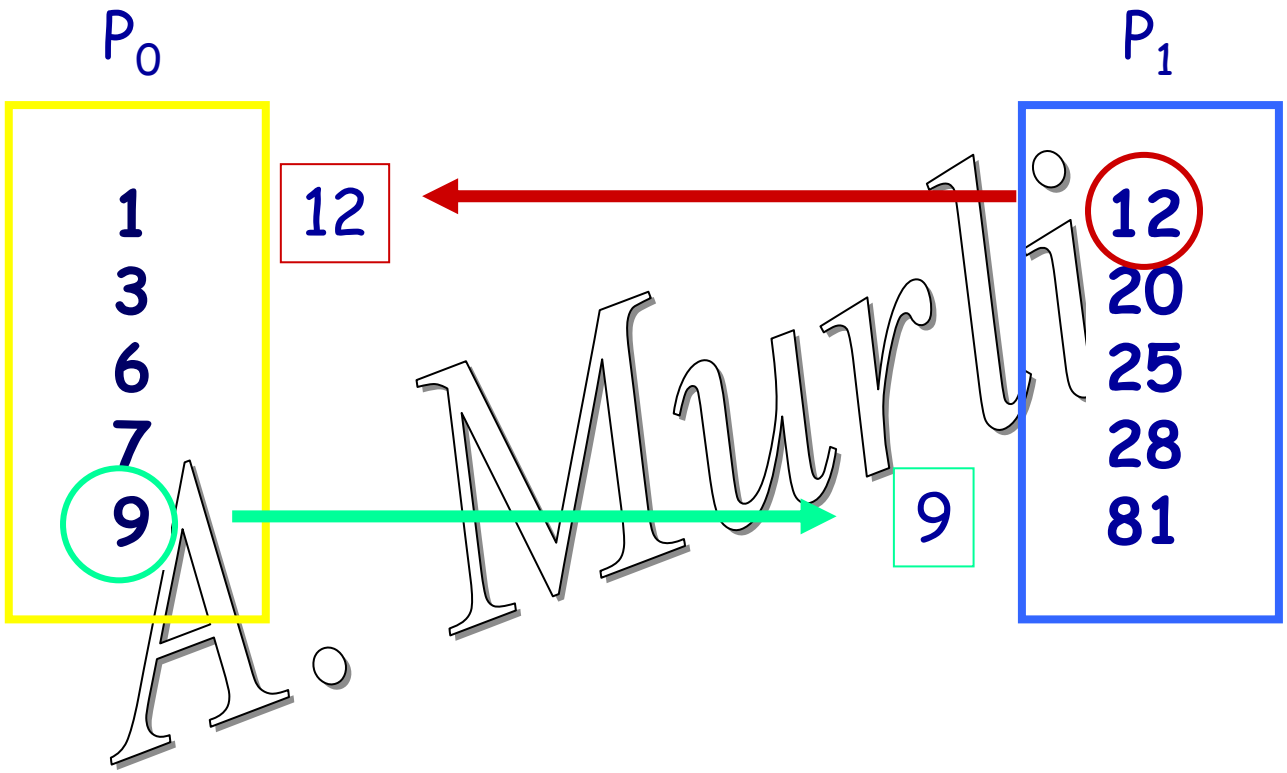
A. Murli

IN PARALLELO

Step 2.2: Ciascun processore

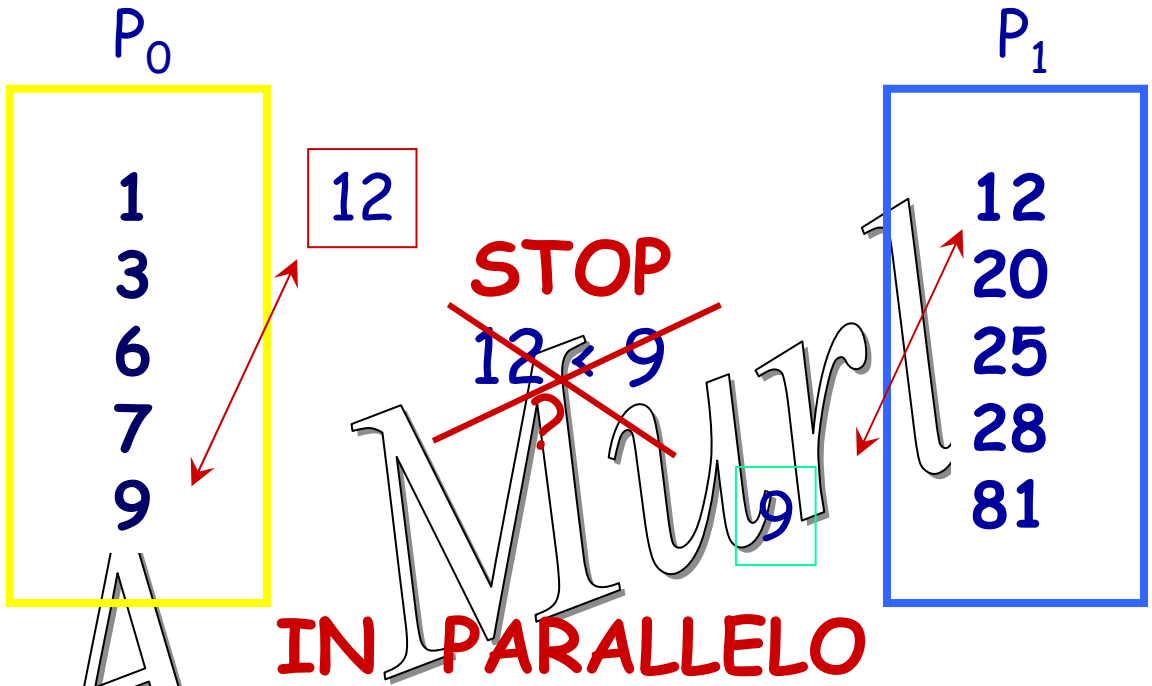
1) Inserisce il numero ricevuto
2) Elimina il numero inviato
3) Riordina il vettore

III PASSO: step 3.0



Step 3.0: P_0 invia a P_1 il proprio max = 9,
mentre P_1 invia a P_0 il proprio min = 12

III PASSO: step 3.1



Step 3.1: P_0 confronta il numero 12 con 9: non è minore e termina l' esecuzione.

P_1 confronta il numero 9 con 12: non è maggiore e termina l'esecuzione

PGBS:Ordinamento parallelo di un vettore

I 5 numeri più **piccoli** sono in P_0
mentre i 5 più **grandi** sono in P_1

Le due sottoliste iniziali sono globalmente ordinate secondo l'ordine crescente dell'identificativo dei processi

Finora abbiamo risolto il problema dell'ordinamento nel caso p=2.

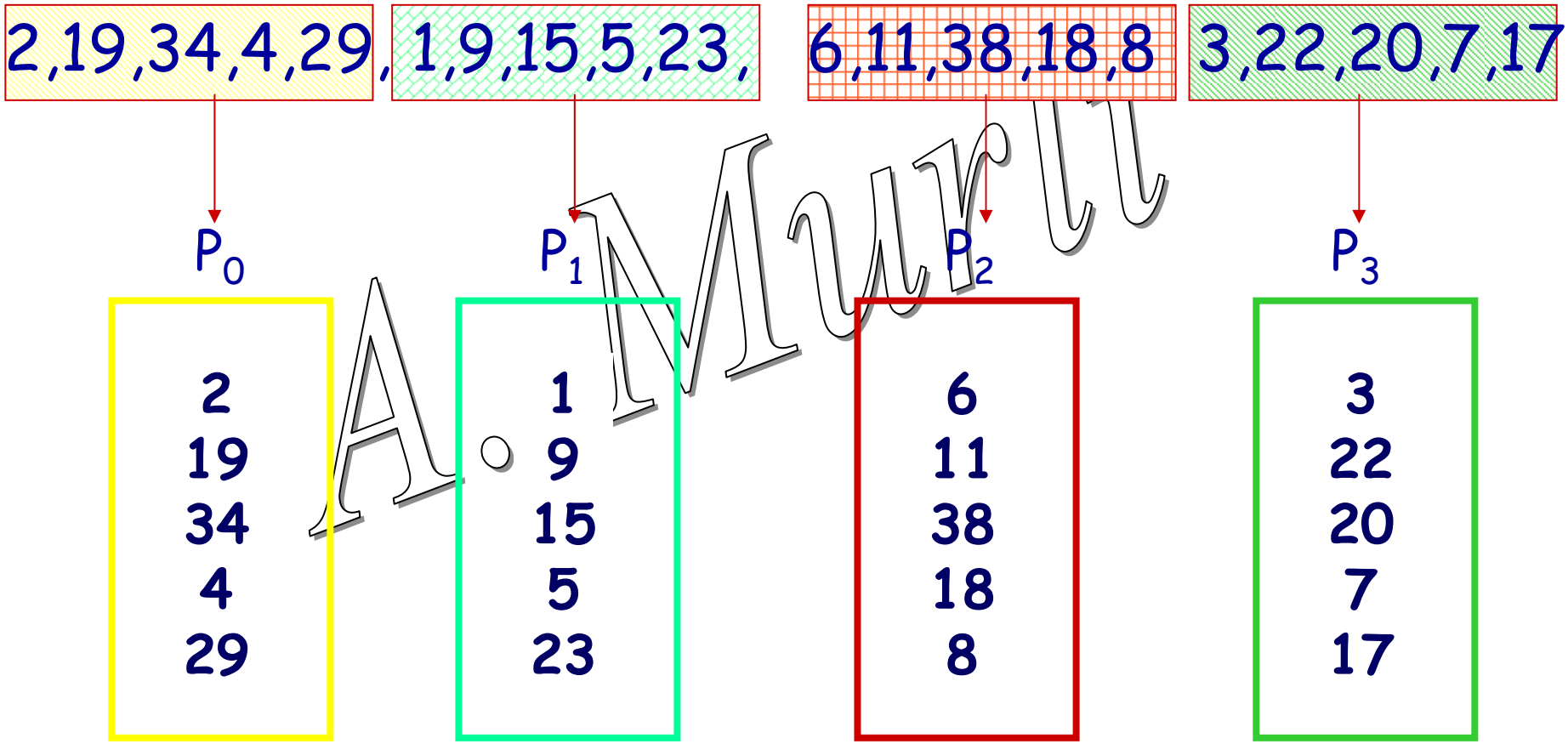
Come fare nel caso

p>2

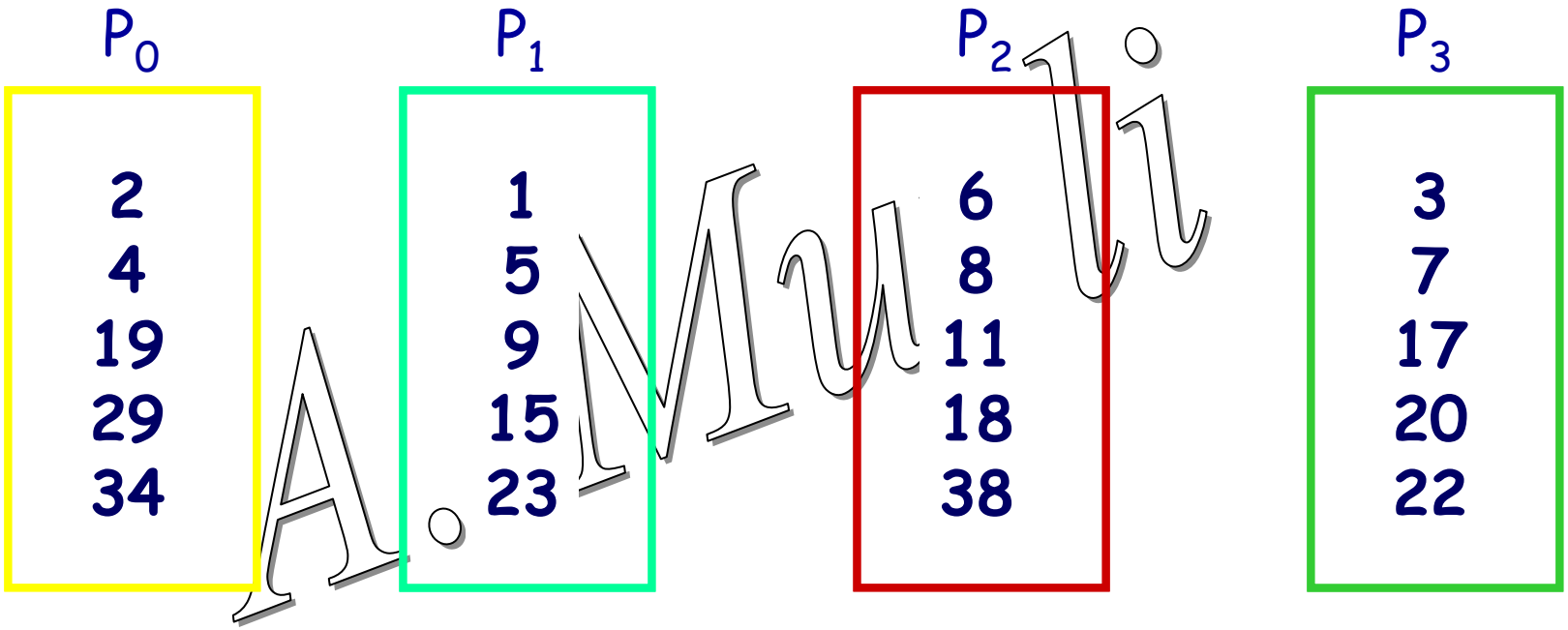
?

Esempio: p=4, n=20

Passo 1: distribuzione dati

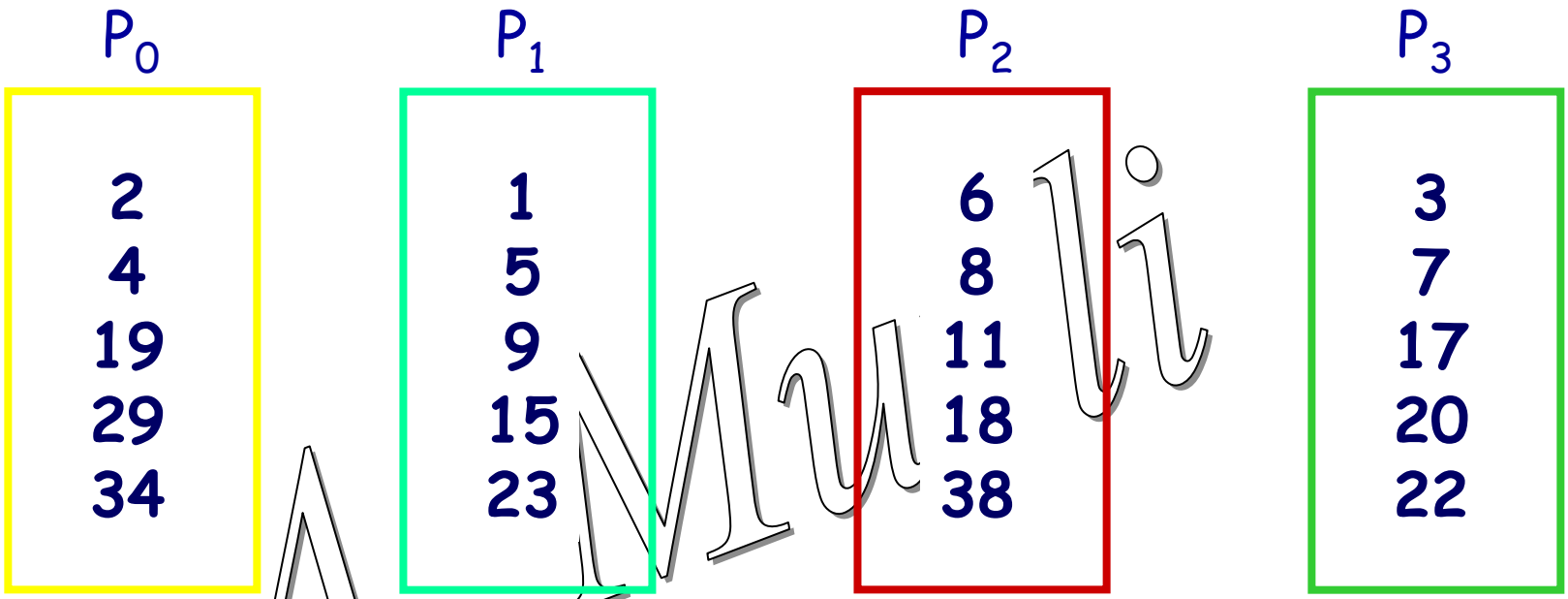


Esempio: p=4, n=20 FASE 1 (SORTING LOCALE)



Ogni processore ordina in modo crescente
la propria sottolista

Esempio: p=4, n=20 FASE 2



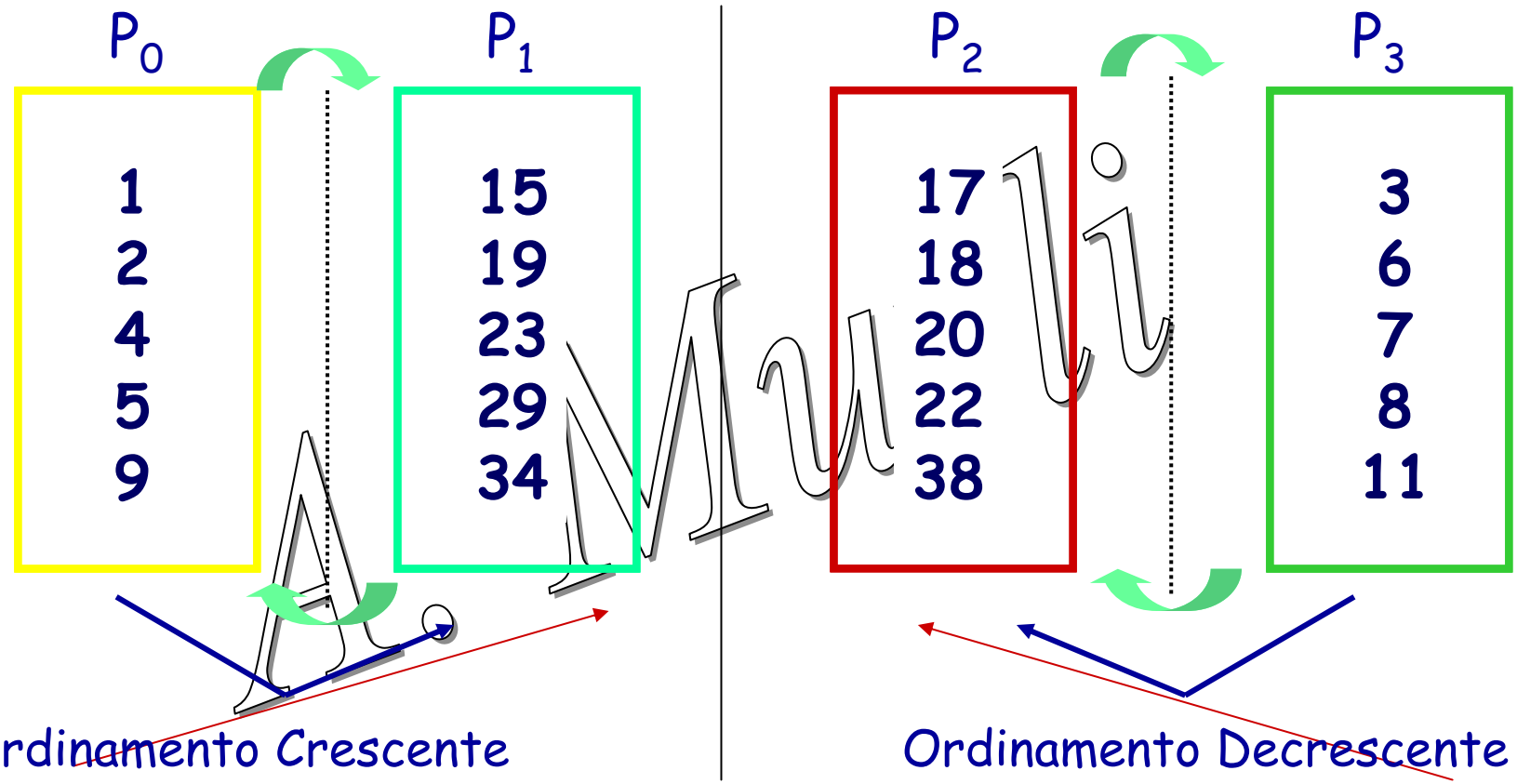
Si applica l'algoritmo GBS:

1) Si trasforma il vettore in uno bitonico;

2) Si ordina il vettore bitonico con il BBS

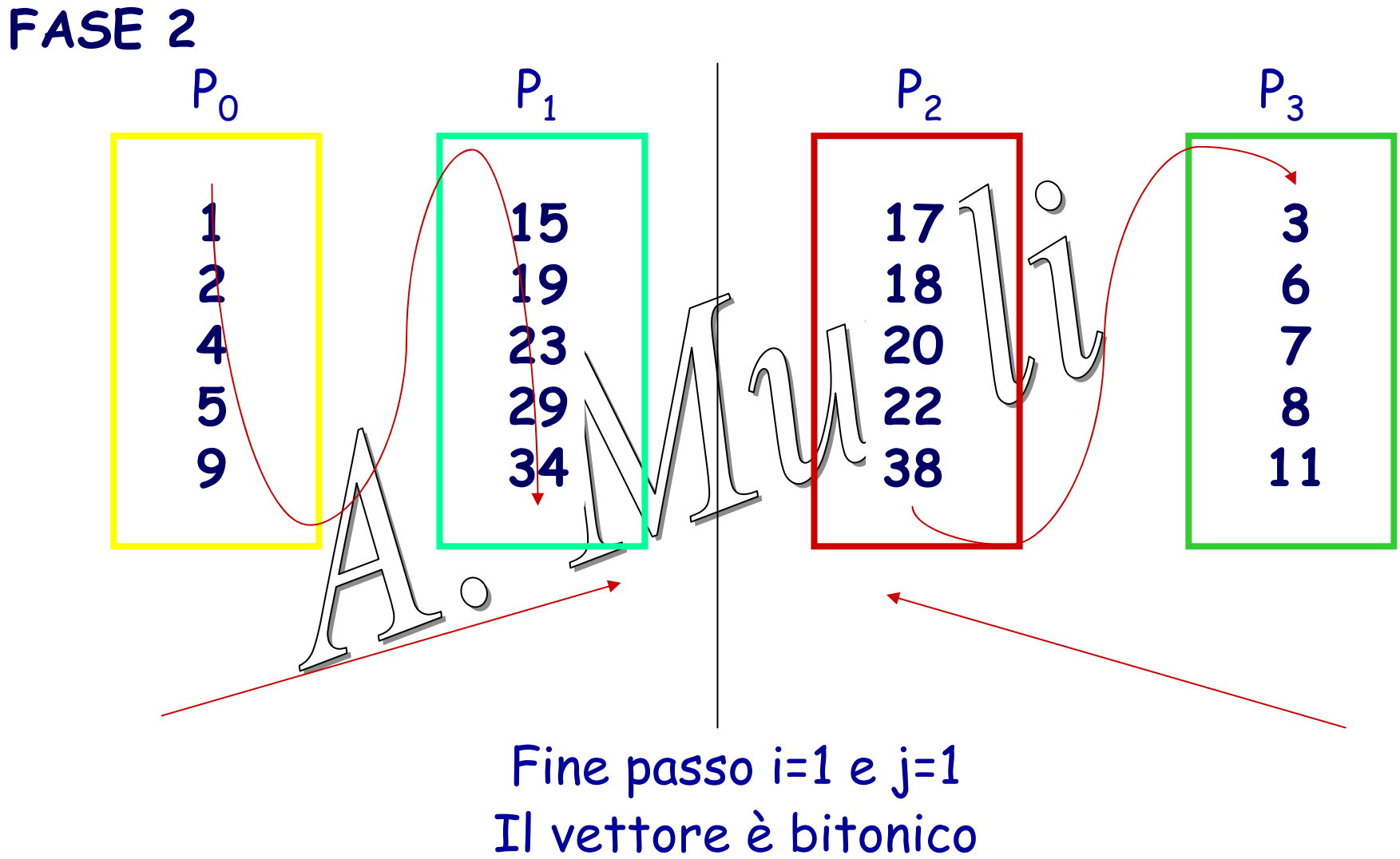
Esempio: p=4, n=20 Passo i=1 (j=1)

FASE 2



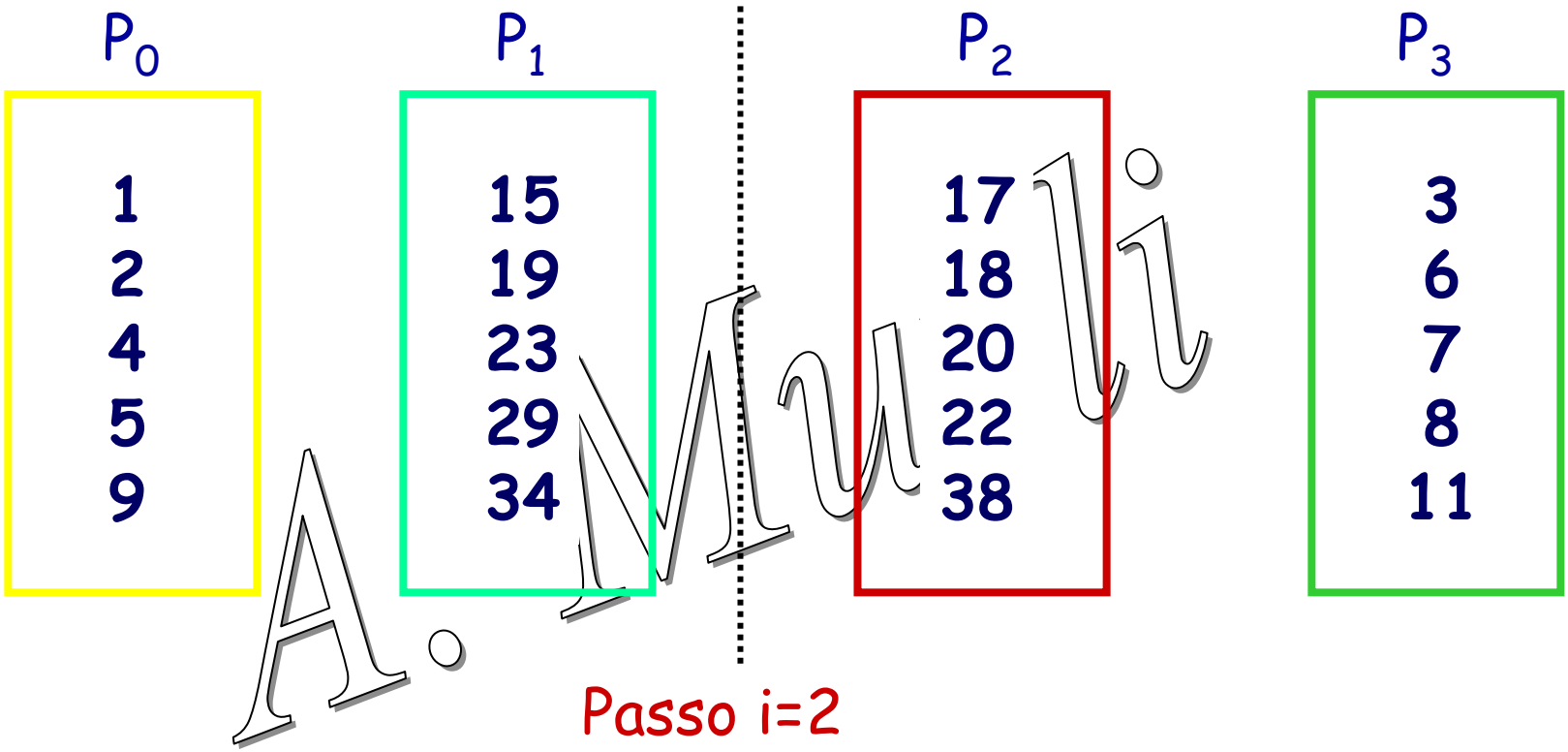
Co-Ex-Lo (proc basso) | Co-Ex-Hi (proc alto) | Co-Ex-Hi (proc alto) | Co-Ex-Lo (proc basso)

Esempio: p=4, n=20



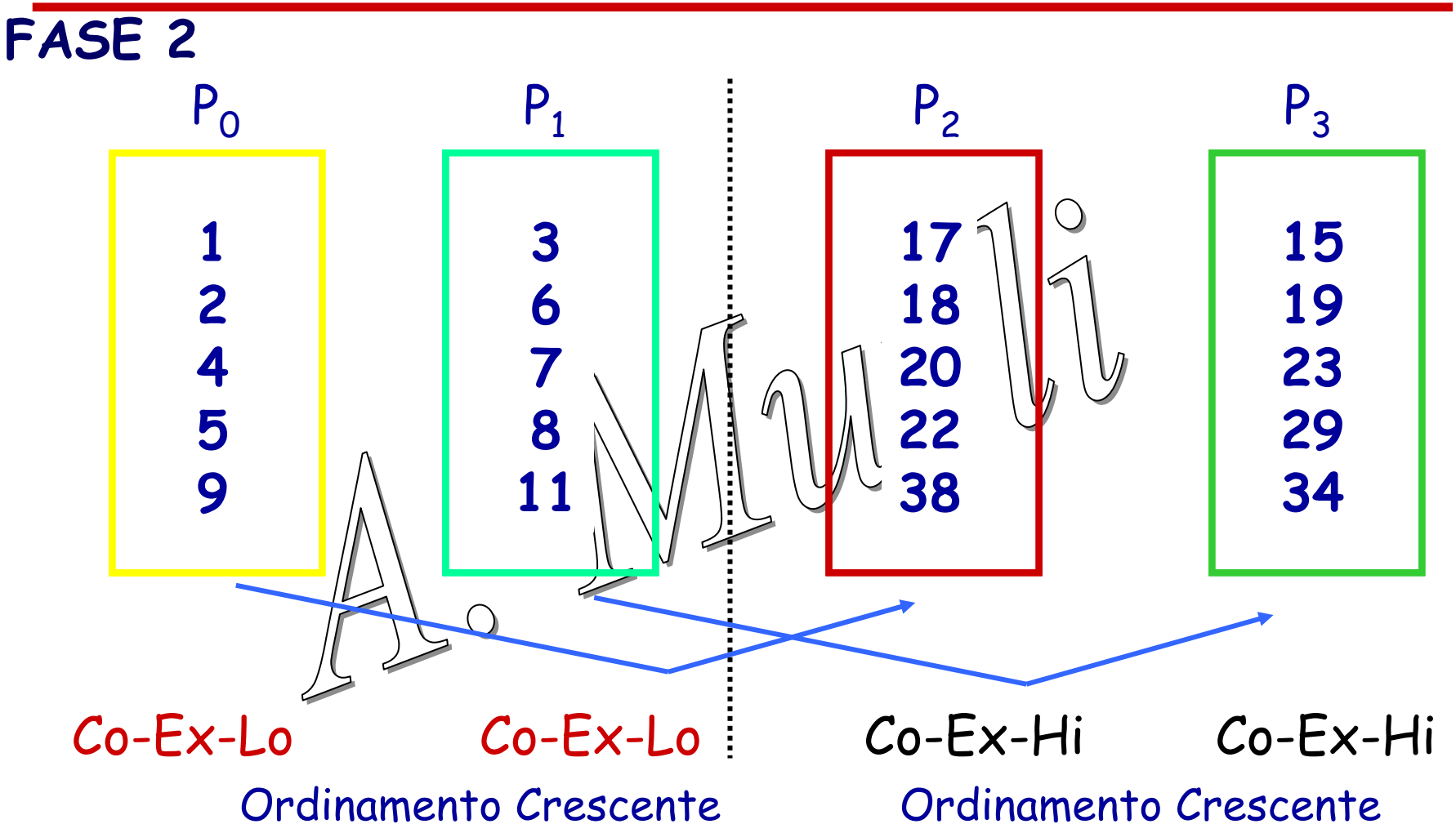
Esempio: p=4, n=20

FASE 2



ordiniamo il vettore con il BBS

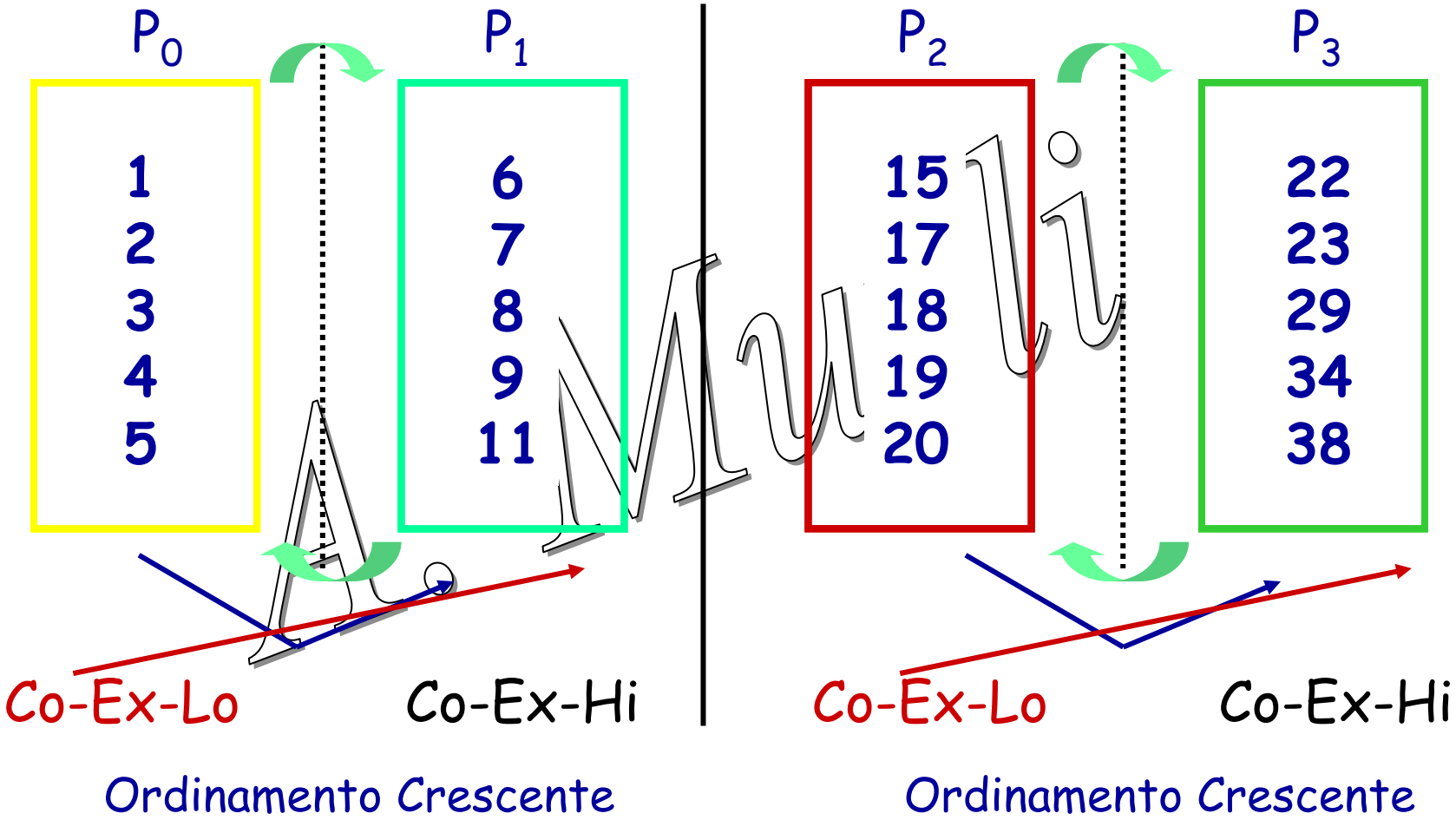
Esempio: p=4, n=20 Passo i=2 (j=1)



Esempio: p=4, n=20

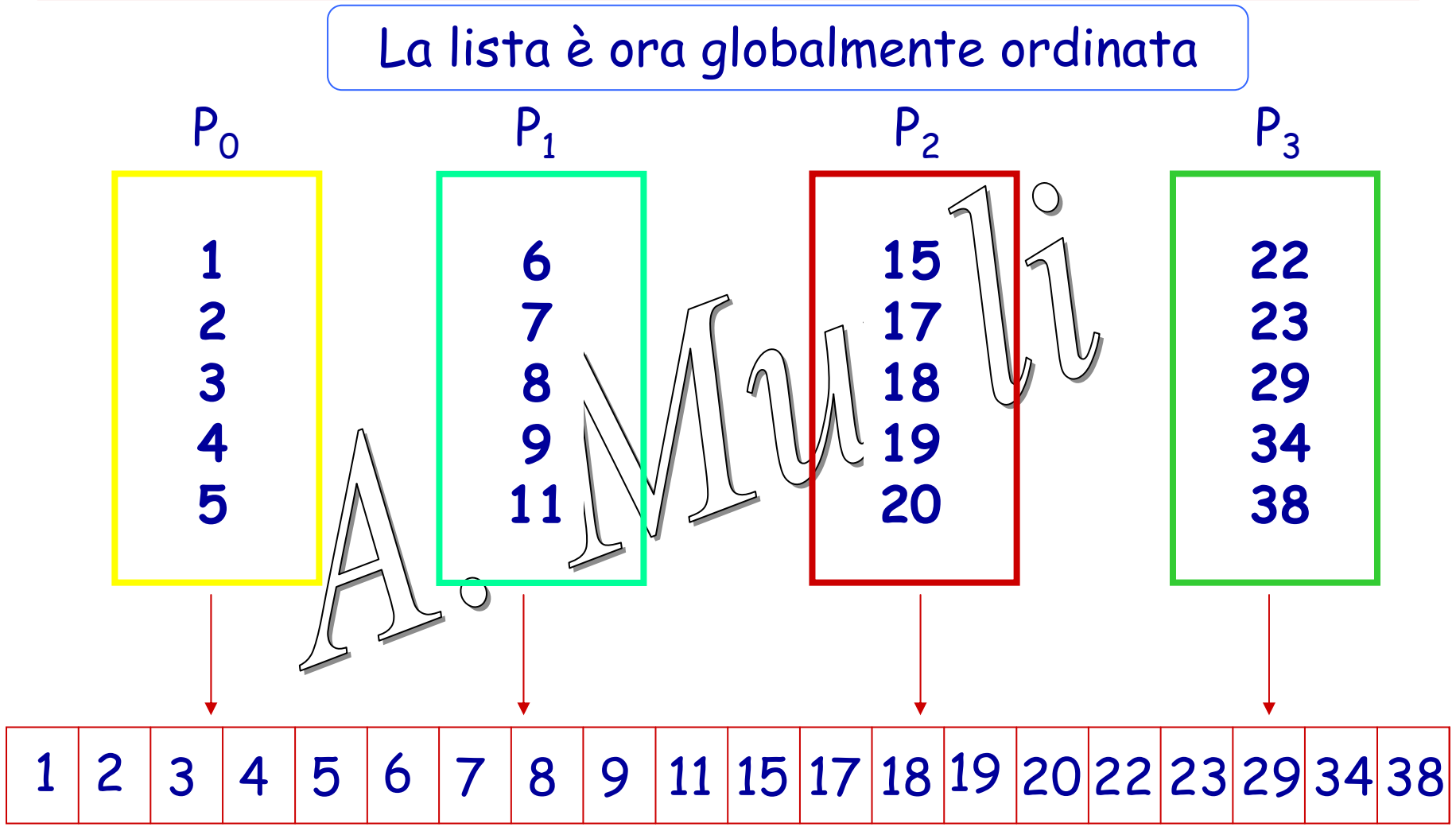
FASE 2 (MER

La lista è ora globalmente ordinata



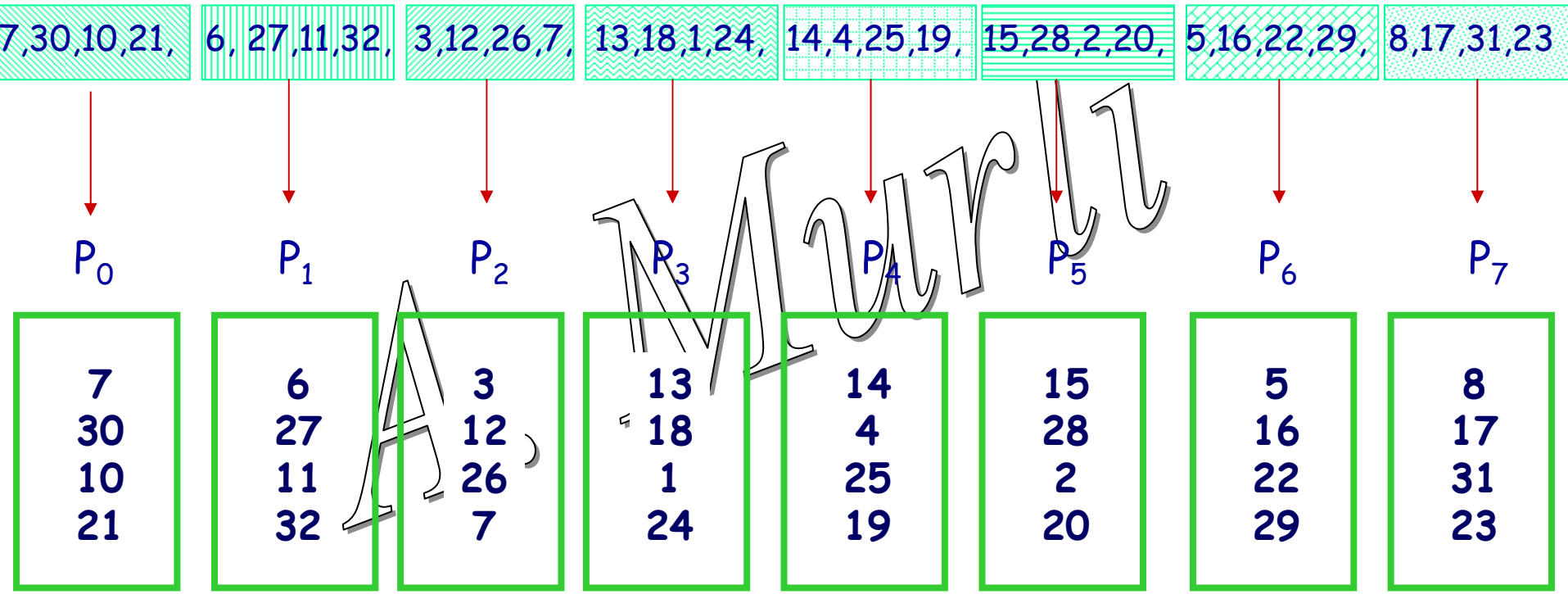
Passo i=2 (j=2)

Esempio: p=4, n=20



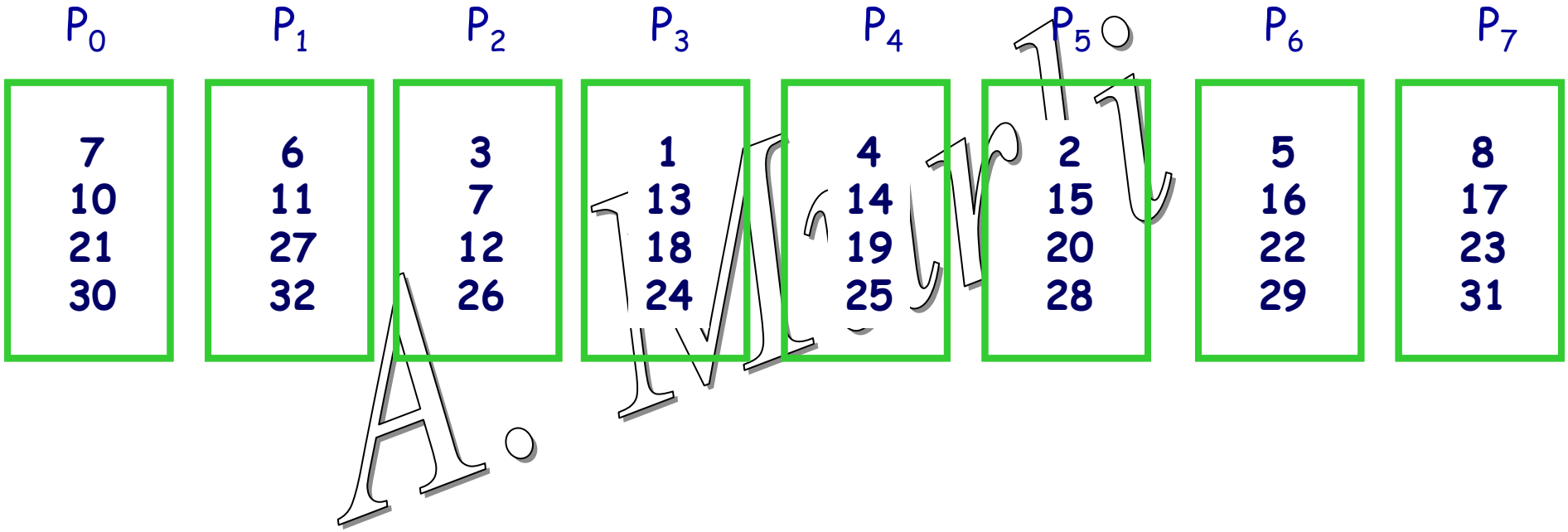
Esempio: p=8, n=32

Passo 1: distribuzione dati



Esempio: p=8, n=32

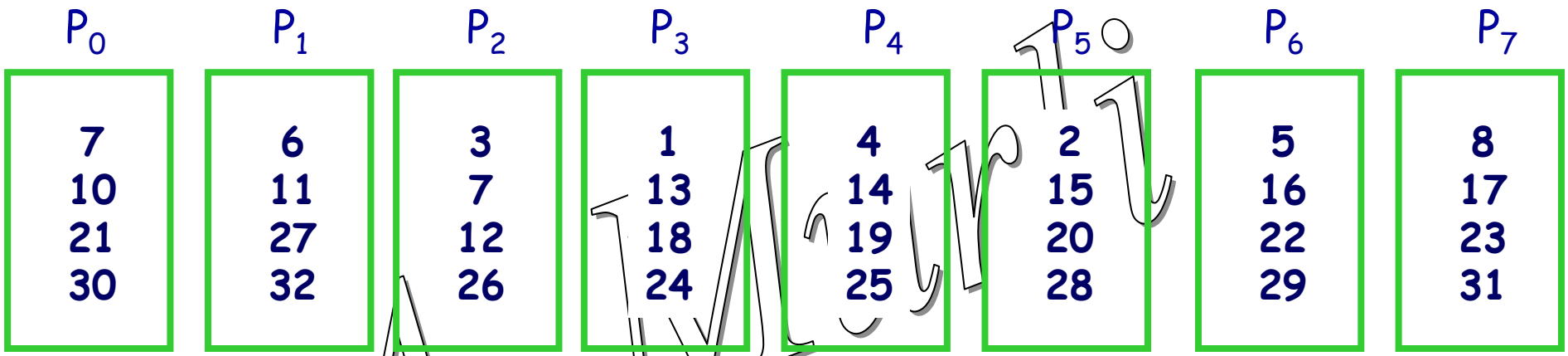
FASE 1 (SORTING LOCALE)



Ogni processore ordina in modo crescente la propria sottolista

Esempio: p=8, n=32

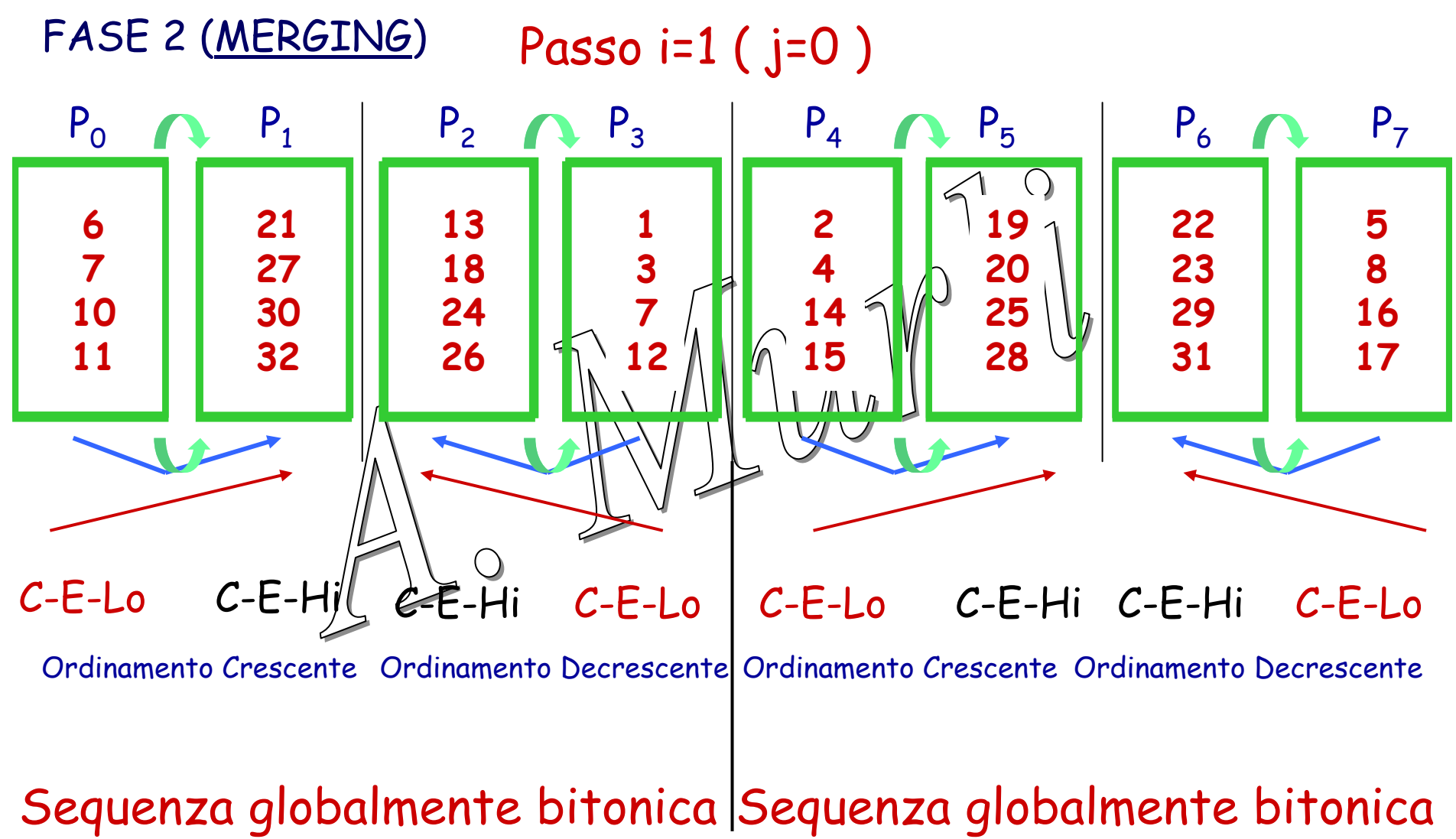
FASE 2 (MERGING)



A. Murli

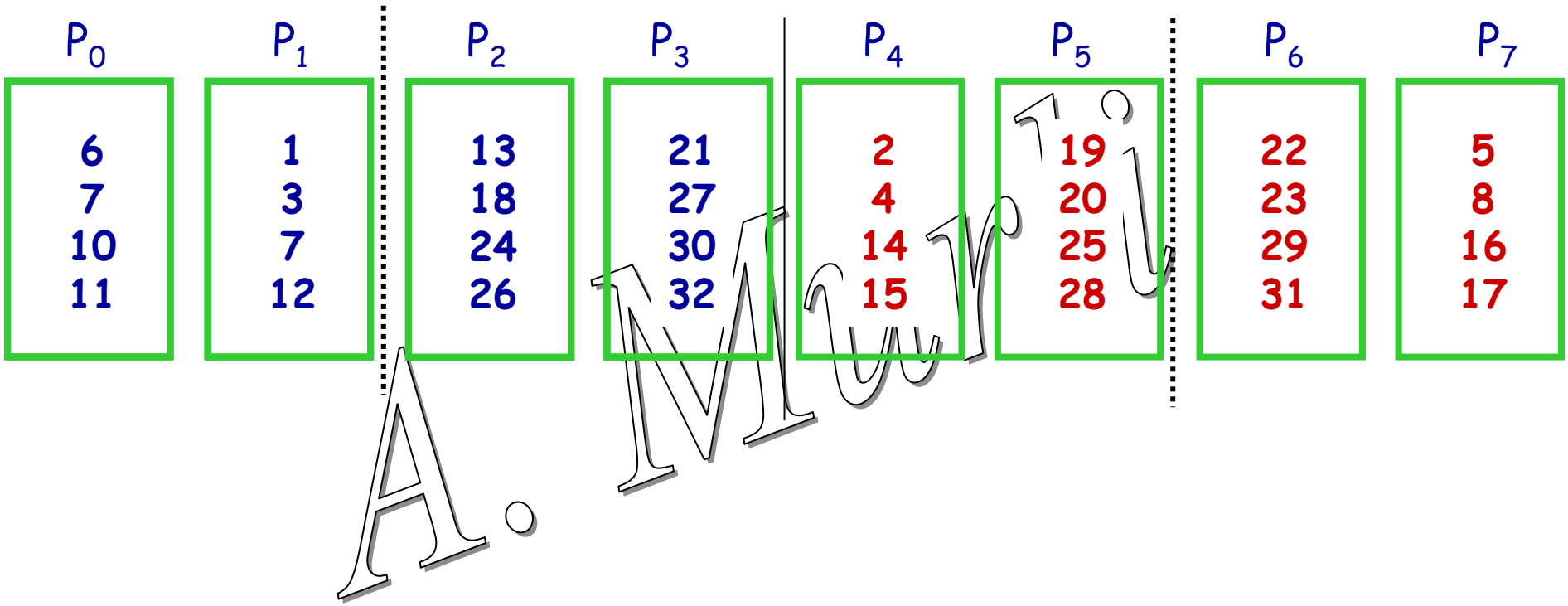
Applichiamo l'algoritmo GBS

Esempio: p=8, n=32



Esempio: p=8, n=32

FASE 2 (MERGING)



Passo i=2 (j=1): ordiniamo il vettore con il BBS

… si continua in esercitazione !

A. Murli

In sintesi

- Non si effettuano operazioni aritmetiche ma solo confronti e scambi
- Ogni scambio comporta una comunicazione tra coppie di processori
- L'overhead di comunicazione incide in maniera significativa sulle prestazioni dell'algoritmo parallelo