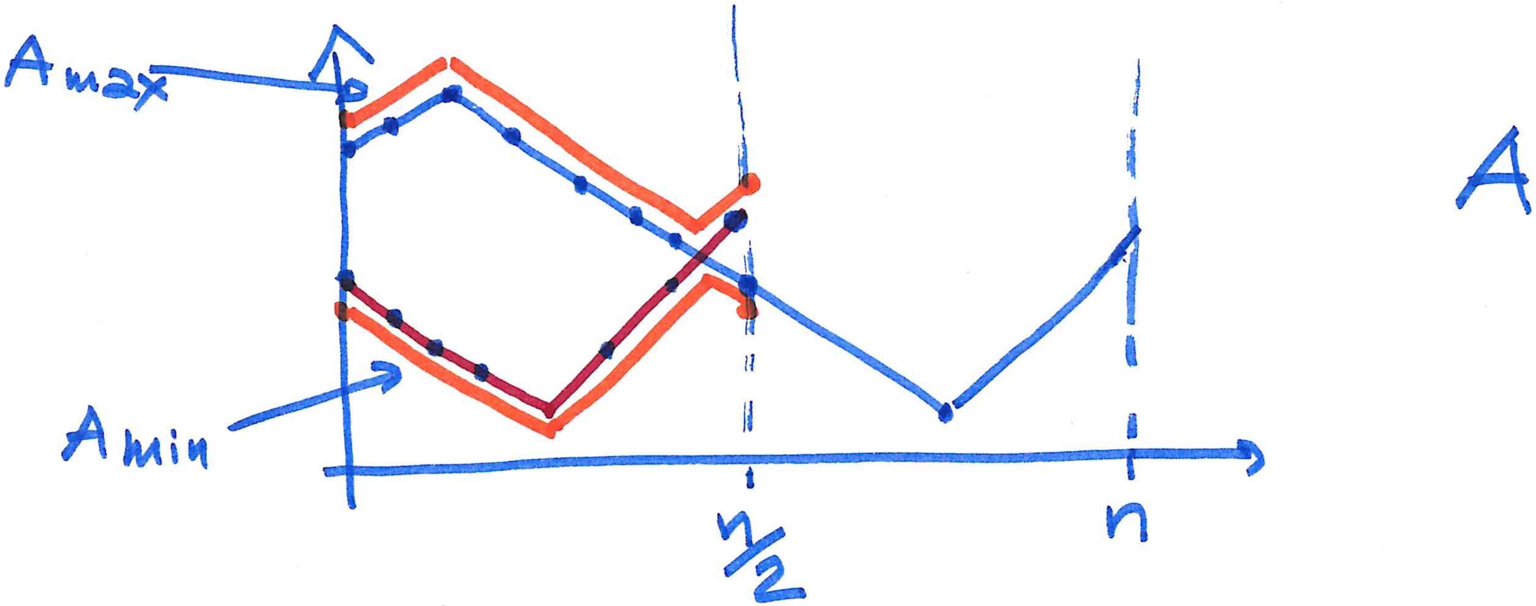
Proposizione per seq. BITONICHE

Sia A bitonica, dopo MINMAX(A) si ha:

1) A_{min} e A_{max} sono ancora bitoniche

2) A_{min} è fatto di elementi minori degli elementi di A_{max}



Tali proprietà suggeriscono un approccio "Divide et Impera" per le BITONICHE

1) MINMAX suddivide il problema su n elementi in istanze più piccole: Amin, Amax grazie alla prop. 1

2) ordinando Amin e Amax la fusione (merge) avviene per concatenazione grazie alla prop. 2

Bitmerge sequenziale

Procedura bit-merge$(\underbrace{A[1], \ldots, A[n]}_{\text{bitonico}})$

```
{   MINMAX (A)
    if ( |A| > 2 )
      { bit-merge (Amin)
        bit-merge (Amax)
      }
    return (A)
}
```

Bit-merge ordina sequenze BITONICHE!

Correttezza di BITMERGE

Si usa l'induzione

<u>caso base</u> : $n = 2$

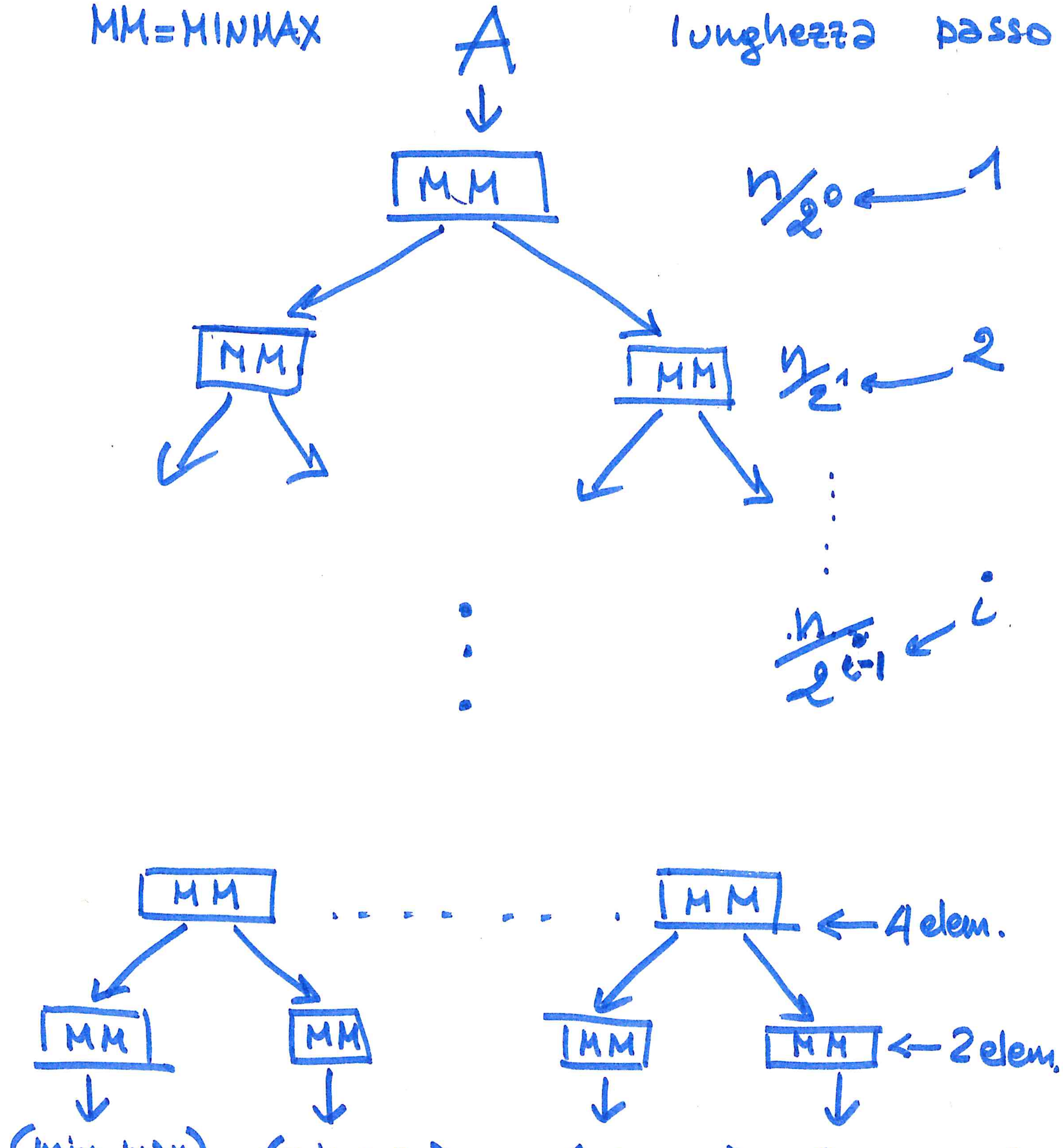
Una sequenza di lunghezza 2 è banalmente ordinata da MINMAX

<u>passo induttivo</u> : supponiamo corretto per $n = 2^k$
dimostriamo che è valido per $2^{k+1} =$

- MINMAX mi ritorna Amin e Amax di lung. 2^k
- Bitmerge (Amin) e Bitmerge (Amax) ordinano Amin e Amax per ipotesi induttiva
- return (A) ritorna A ordinato

IMPLEMENTAZIONE PARALLELA DI BITMERGE

MM=MINMAX



VALUTAZIONE

- è EREW-PRAM

- **Tempo**

al passo i-esimo operiamo su istanze di lunghezza $n/2^{i-1}$

$$\frac{n}{2^{i-1}} = 2 \leftarrow \text{ultimo passo}$$

si ottiene $i = \log n$

Dato che MINMAX costa 5 op.

$T(n) = 5 \log n$

- **Processori**

passo	n^2 proc. per MM
1	$n/2$
2	$n/4 + n/4 = \frac{n}{2}$
$\vdots$	

$\Rightarrow P = \frac{n}{2}$

Tempo visto con un'equazione di ricorrenza:

$$T(n) = \begin{cases} 5 & n = 2 \\ T\left(\frac{n}{2}\right) + 5 & \text{altrimenti} \end{cases}$$

$$T(n) = 5 \log n$$

- <u>efficienza</u>

$$E = \frac{n \cdot \log n}{\frac{n}{2}\, 5 \log n} \rightarrow \text{Costante} \neq 0$$

Bit-sort sequenziale (Batcher 1968)

Procedura bit-sort $\big(\overbrace{A[1], \ldots, A[n]}^{\text{generico}}\big)$

```
{   MINMAX(A)
    if ( |A| > 2)
      {  bit-sort (Amin)
         bit-sort (Amax)
  ──→    bit-merge( Amin · REV(Amax) )
      }
    return (A)
}
```

$$\text{bit-merge}\big(\underbrace{A_{min} \cdot REV(A_{max})}_{\text{UNIMODALE} \Rightarrow \text{BITONICA}}\big)$$

CORRETTEZZA DI BITSORT

Si usa l'induzione

<u>caso base</u> $n = 2$

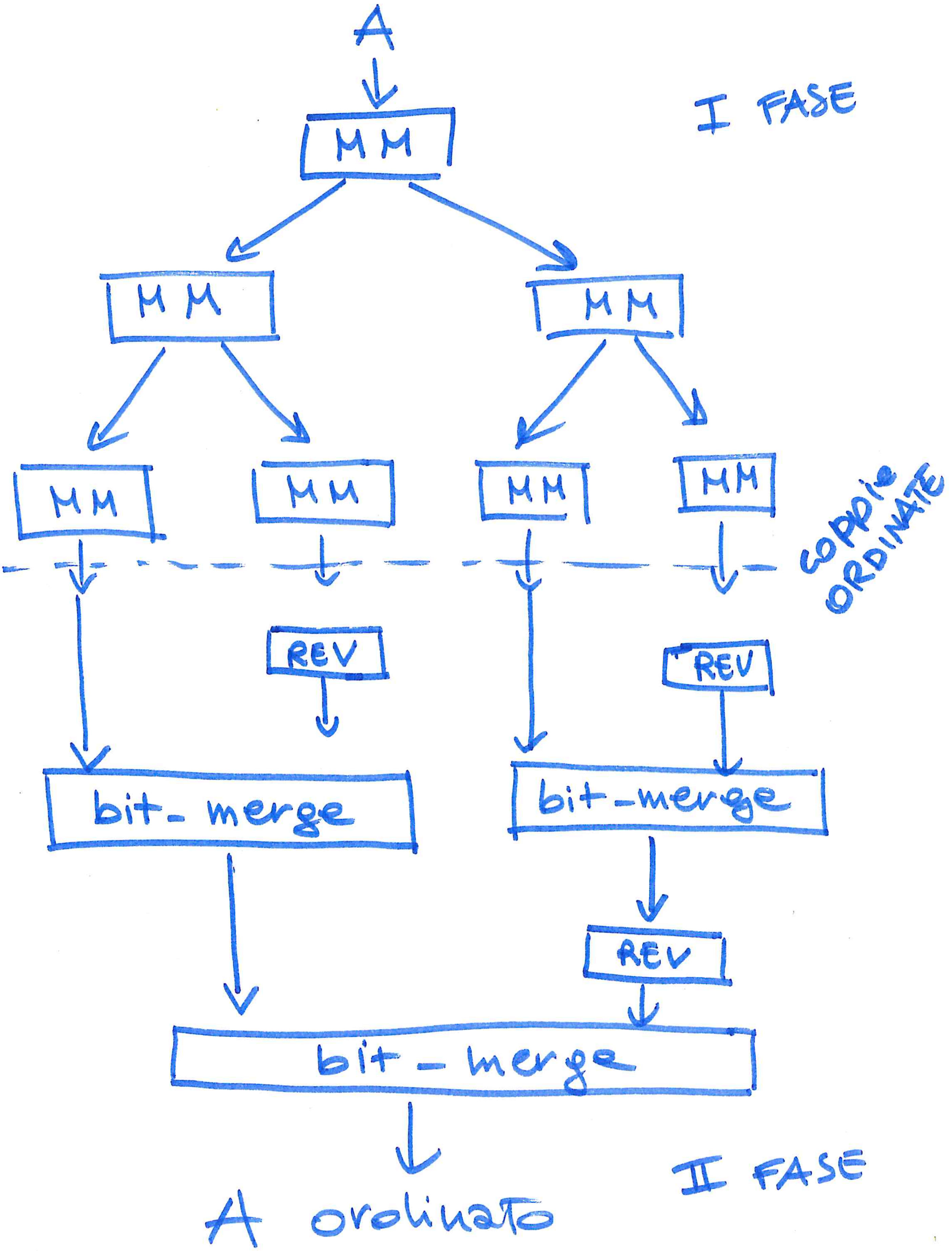
MINMAX su due elementi li ordina ✓

<u>passo induttivo</u> suppongo che sia corretto per 2^k
e dimostro che vale per 2^{k+1}

$|A| = 2^{k+1}$

- MINMAX divide A in Amin e Amax di lung. 2^k
- bit-sort (Amin) e bit-sort (Amax) ordinano Amin e Amax per ipotesi induttiva
- bit-merge ($\underbrace{\text{Amin REV(Amax)}}_{\text{bitonica}}$) che ordina A ✓

IMPLEMENTAZIONE PARALLELA DI BIT-SORT



VALUTAZIONE

- è EREW-PRAM

- <u>tempo</u>

I FASE $\longrightarrow$ i ultimo passo
(come bit-merge) $i = \log n$
si esegue MINMAX che richiede Tempo cost.
$T(n) = O(\log n)$

II FASE $\longrightarrow$ i ultimo passo
$i = \log n - 1$

si esegue REV (costante) e
bit-merge che richiede $O(\log n)$
$T(n) = O(\log^2 n)$

- <u>processori</u>

I FASE $n/2$ processori

II FASE
REV $\longrightarrow n/2$
BIT-merge $\longrightarrow n/2$
$\Rightarrow P = n/2$

Equazione di ricorrenza del Tempo di Bit-sort

$$T(n) = \begin{cases} 5 & n = 2 \\ T\left(\frac{n}{2}\right) + \underset{\substack{\uparrow \\ \text{MM}}}{5} + \underset{\substack{\uparrow \\ \text{REV}}}{4} + \underset{\substack{\uparrow \\ \text{Bit-merge}}}{5 \log n} & \text{altrimenti} \end{cases}$$

$$T(n) = \frac{5 \log^2 n + 23 \log n - 18}{2}$$

$$E = \frac{n \cdot \log n}{\frac{n}{2} \cdot 5 \log^2 n} \rightarrow \frac{\alpha}{\log n} \rightarrow 0$$

lentamente