

فصل ۳

ادغام موازی

سید ناصر رضوی

razavi@iust.ac.ir

۱۳۹۰

معرفی مسئله انتخاب

۲

□ مسئله ادغام:

□ ورودی:

■ دنباله های مرتب A و B

□ خروجی:

■ دنباله مرتب C شامل عناصر A و B

□ مثال:

A = [2, 5, 7, 10] **B** = [1, 4, 8, 9] □ ورودی:

C = [1, 2, 4, 5, 7, 8, 9, 10] □ خروجی:

فهرست مطالب

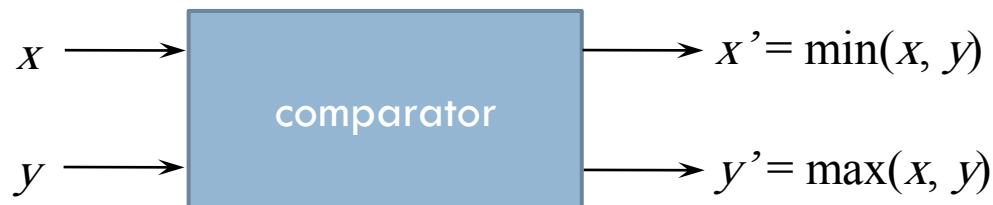
۳

- یک معماری موازی خاص منظوره برای ادغام CREW SIMD
- یک الگوریتم بهینه برای مدل EREW SIMD فوق بر روی شبیه سازی الگوریتم
- یک الگوریتم بهینه برای مدل EREW SIMD

شبکه ادغام گر (r, s)

۴

یک شبکه از پردازشگرهای ساده و یکسان با نام مقایسه گرها □



مقایسه گر: □

شبکه ادغام گر: □

دنباله های مرتب A و B □

خروجی: دنباله مرتب C حاصل از ادغام دنباله های A و B □

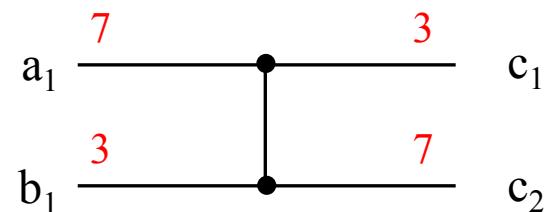
$$C = [c_1, c_2, \dots, c_{r+s}]$$

ساده سازی: فرض می کنیم □

سُبکه ادغام ڪر (l, l)

۶

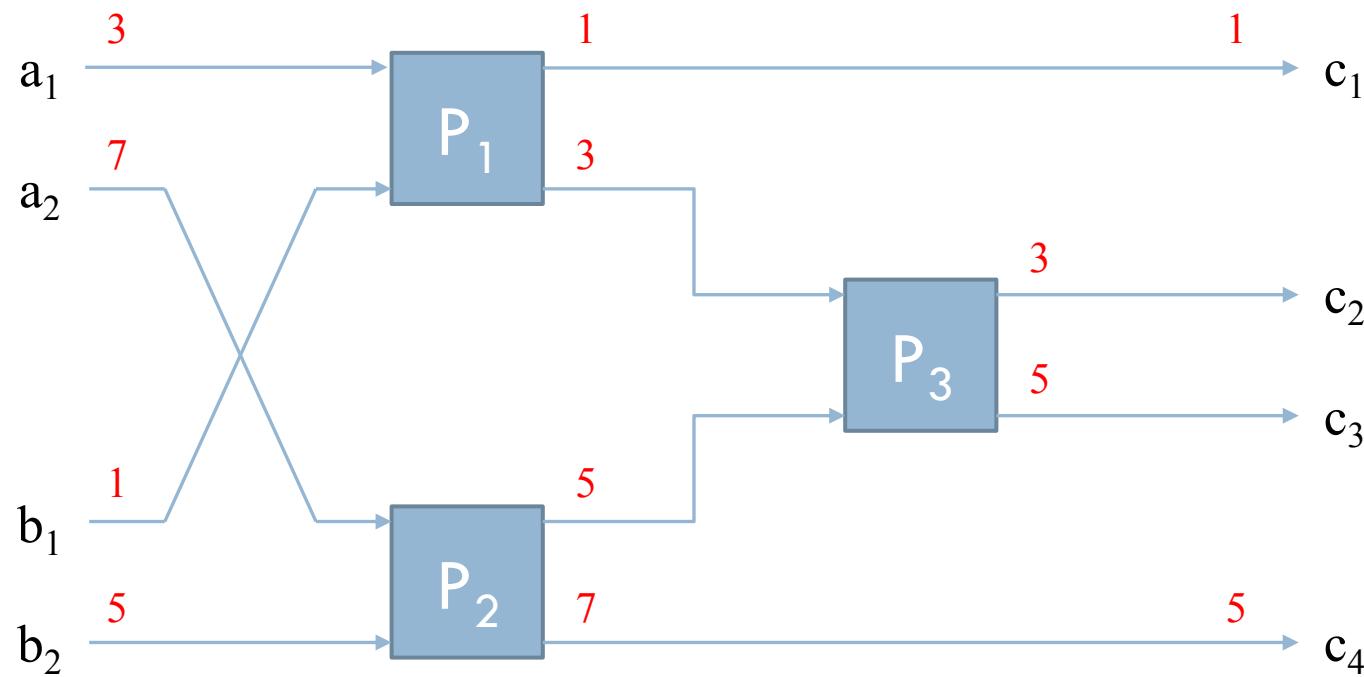
□ **n = 1**



سُبکه ادغام کر (P, P) میگیرد

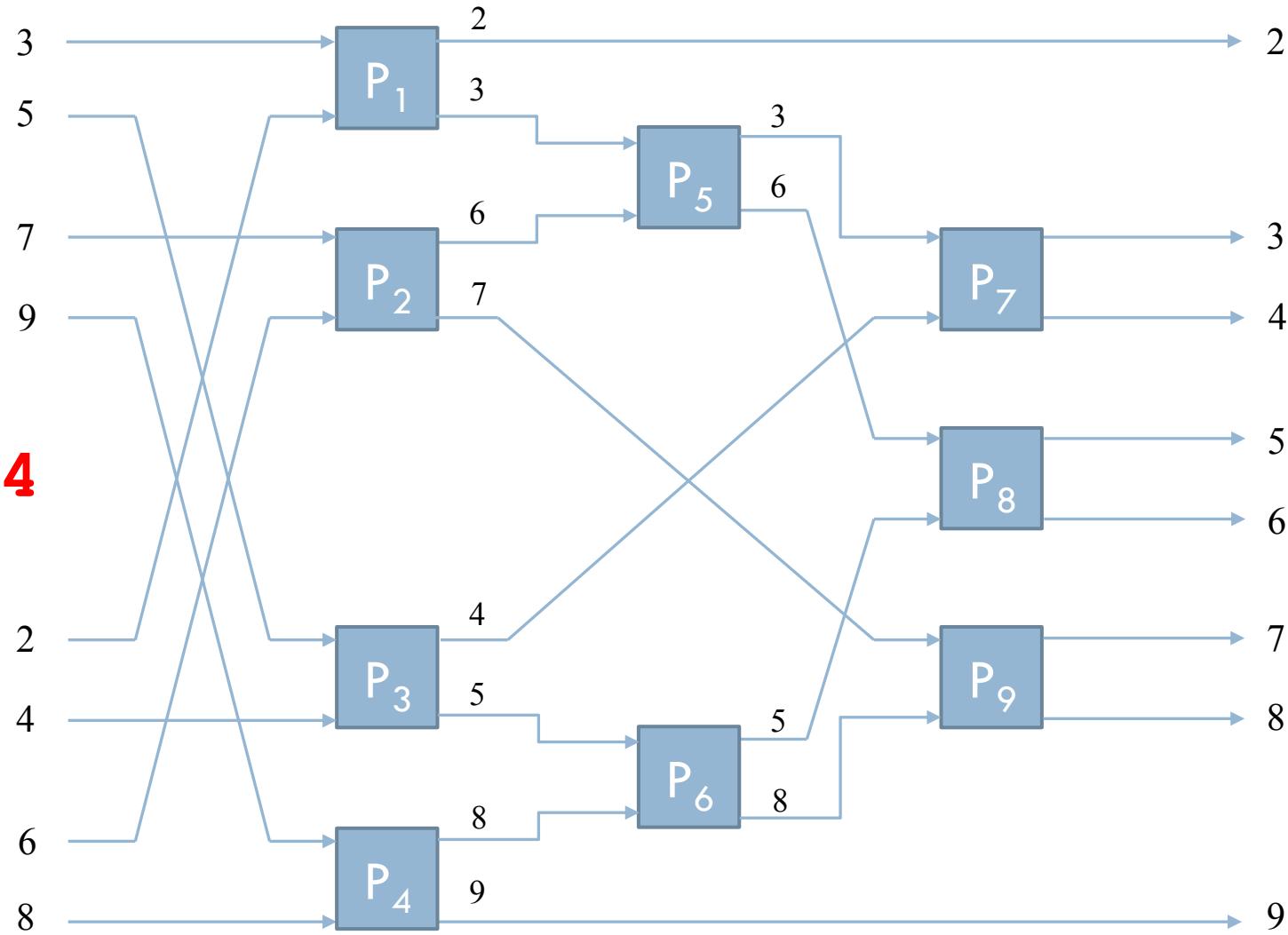
۶

□ $n = 2$



سُبکہ ادغام کر (K, K) میں

v



شبکه ادغام گر (n, n)

۸

□ معماری شبکه ادغام گر (n, n)

□ ادغام عناصر واقع در مکان های فرد A و B با استفاده از یک ادغام گر (n/2, n/2)

- $A_{\text{odd}} = [a_1, a_3, \dots, a_{n-1}]$
- $B_{\text{odd}} = [b_1, b_3, \dots, b_{n-1}]$
- $D = [d_1, d_2, \dots, d_n]$

□ ادغام عناصر واقع در مکان های زوج A و B با استفاده از یک ادغام گر (n/2, n/2)

- $A_{\text{even}} = [a_2, a_4, \dots, a_n]$
- $B_{\text{even}} = [b_2, b_4, \dots, b_n]$
- $E = [e_1, e_2, \dots, e_n]$

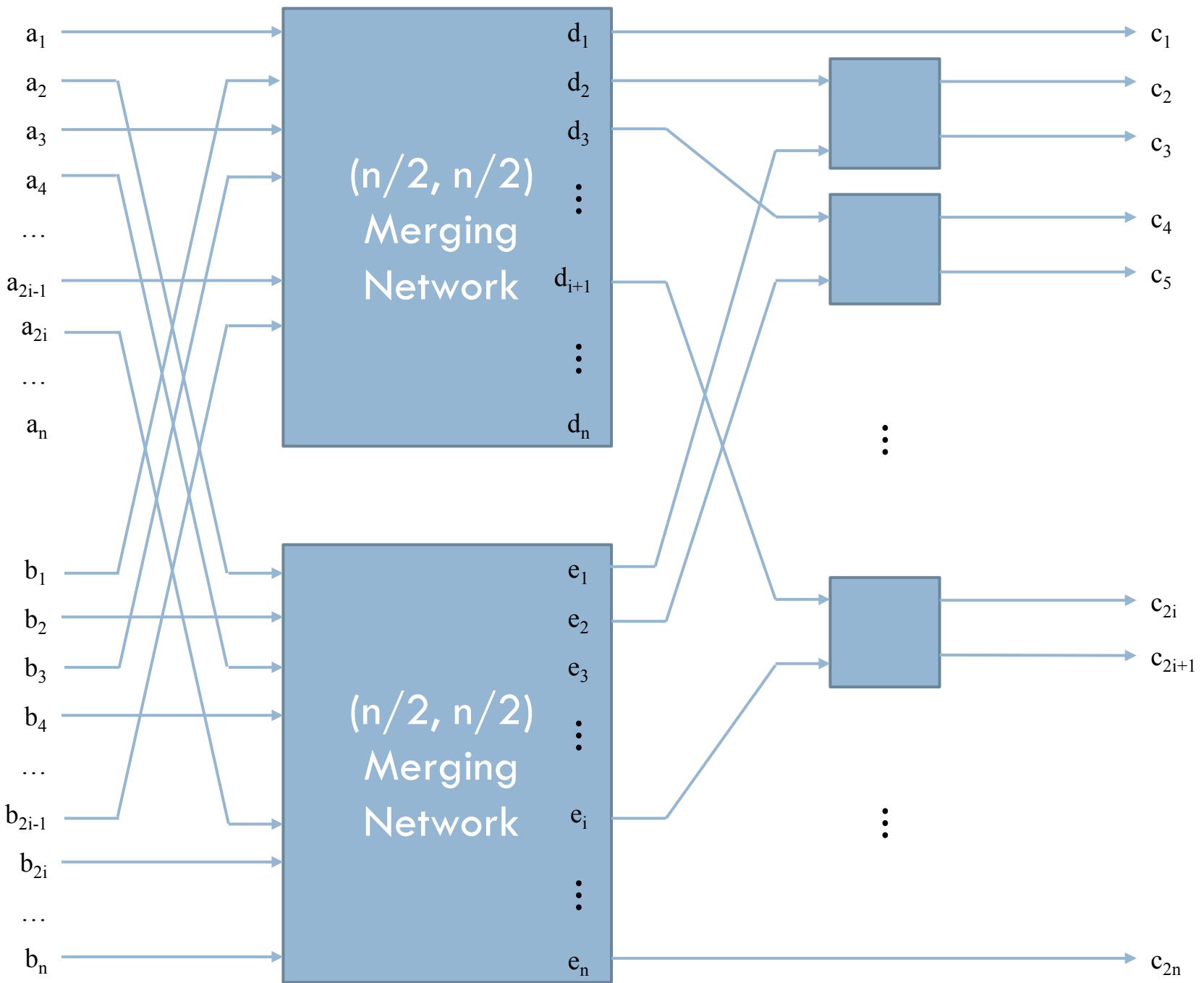
شبکه ادغام کر (n, n)

۹

□ محاسبه دنباله نهايی C به صورت زير:

- $C_1 = d_1,$
- $C_{2i} = \min(d_{i+1}, e_i), \quad i = 1, 2, 3, \dots, n - 1$
- $C_{2i+1} = \max(d_{i+1}, e_i),$
- $C_{2n} = e_n,$

□ اين مرحله به وسيله $n - 1$ مقاييسه گر و به صورت همزمان انجام می شود.



شبکه ادغام کر (n, n)

۱۱

□ تحلیل: زمان اجرا

$$t(2) = 1$$

$$t(2n) = t(n) + 1$$

□ در نتیجه

$$t(n) = \log n + 1$$

□ زمان اجرای به دست آمده از زمان اجرای بهترین الگوریتم ترتیبی
یعنی $O(n)$ بسیار بهتر است.

شبکه ادغام کر (n, n)

۱۲

□ تحلیل: تعداد پردازنده ها

$$p(2) = 1$$

$$p(2n) = 2p(n) + (n - 1)$$

$$p(n) = n \log n + 1$$

□ تحلیل: هزینه

$$c(2n) = p(2n) \cdot t(2n) = O(n \log^2 n)$$

□ در نتیجه شبکه ارائه شده از نظر هزینه بهینه نیست.

ادغام (وی مدل CREW

۱۳

- مشکل عمومی شبکه های ادغام گر: عدم انعطاف پذیری
- تعداد ثابتی از پردازنده ها با یک پیکره بندی ثابت برای ادغام ورودی هایی با یک اندازه ثابت
- یعنی تطبیق پذیر نیستند (بر حسب تعداد پردازنده ها)
- از طرفی شبکه ادغام ارائه شده از نظر هزینه بهینه نیست.
- در بخش بعدی، یک الگوریتم ادغام موازی تطبیق پذیر و بهینه برای مدل SIMD CREW ارائه می کنیم.

الگوریتم ادغام ترتیبی

۱۴

procedure SEQUENTIAL-MERGE (A, B, C)

Step 1: (1.1) $i \leftarrow 1$
 (1.2) $j \leftarrow 1$

Step 2: **for** $k = 1$ **to** $r + s$ **do**
 if ($A[i] < B[j]$) **then**

O(n)

 (i) $C[k] \leftarrow A[i]$
 (ii) $i \leftarrow i + 1$

$A[r+1] = B[s+1] = \infty$

else

 (i) $C[k] \leftarrow B[j]$
 (ii) $j \leftarrow j + 1$

ادغام موازی

۱۵

□ ویژگی های مطلوب:

- i. تعداد پردازنده ها زیرخطی و تطبیق پذیر باشد.
- ii. زمان اجرای الگوریتم تطبیق پذیر باشد و به طور قابل ملاحظه ای از زمان اجرای بهترین الگوریتم ترتیبی کمتر باشد.
- iii. هزینه آن بهینه باشد.

□ رویه های استفاده شده در الگوریتم ادغام موازی:

- i. ادغام ترتیبی
- ii. جستجوی دودویی

الگوریتم جستجوی دودویی

۱۶

procedure BINARY-SEARCH (S, x, k)

Step 1:

- (1.1) $l \leftarrow 1$
- (1.2) $h \leftarrow n$
- (1.3) $k \leftarrow 0$

Step 2:

while $l \leq h$ **do**

O(log n)

- (2.1) $m \leftarrow \lfloor (l + h) / 2 \rfloor$
- (2.2) **if** ($x = S[m]$) **then**
 - (i) $k \leftarrow m$
 - (ii) $l \leftarrow h + 1$**else if** ($x < S[m]$) **then** $h \leftarrow m - 1$ **else** $l \leftarrow m + 1$

الگوریتم ادغام (وی مدل CREW

۱۷

```
for i = 1 to N - 1 do in parallel
```

O(1) Processor P_i determines a'_i , b'_i from

$$(1.1) \quad a'_i = a_{i[r/N]}$$

$$(1.2) \quad b'_i = b_{i[s/N]}$$

```
end for
```

الگوریتم ادغام (وی مدل CREW

۱۸

(2.1) **for** $i = 1$ **to** $N - 1$ **do in parallel**

O(log N)

(i) P_i uses BINARY SEARCH on B' to find smallest j such that $a'_i < b'_j$

(ii) **if** j exists **then** $v_{i+j-1} \leftarrow (a'_i, i, A)$
else $v_{i+N-1} \leftarrow (a'_i, i, A)$

end if

end for

(2.2) **for** $i = 1$ **to** $N - 1$ **do in parallel**

O(log N)

(i) P_i uses BINARY SEARCH on A' to find smallest j such that $b'_i < a'_j$

(ii) **if** j exists **then** $v_{i+j-1} \leftarrow (b'_i, i, B)$
else $v_{i+N-1} \leftarrow (b'_i, i, B)$

end if

end for

الگوریتم ادغام (وی مدل CREW

۱۹

(3.1) $Q(1) \leftarrow (1, 1)$

(3.2) **for** $i = 2$ **to** N **do in parallel**

O(log s)

if $v_{2i-2} = (a'_{k'}, k, A)$ **then** processor P_i

$(r \leq s)$

(i) uses BINARY SEARCH on B to find
smallest j such that $b_j > a'_{k'}$

(ii) $Q(i) \leftarrow (k[r/n], j)$

else processor P_i

(i) uses BINARY SEARCH on A to find
smallest j such that $a_j > b'_{k'}$

(ii) $Q(i) \leftarrow (j, k[s/n])$

end if

end for

الگوریتم ادغام (وی مدل CREW

۲۰

(3.3) **for** $i = 1$ **to** N **do in parallel**

O((r+s)/N) processor P_i uses SEQUENTIAL MERGE and $Q(i) = (x, y)$ to merge two subsequences beginning at a_x and b_y and places the result in array C beginning at $x + y - 1$. The merge continues until

- (i) An element larger than or equal to the first component of v_{2i} is encountered in each of A and B (when $i \leq N - 1$)
- (ii) No elements are left in either A or B ($i = N$)

ادغام موازی

۲۱

: مثال □

- $A = [2, 3, 4, 6, 11, 12, 13, 15, 16, 20, 22, 24]$
- $B = [1, 5, 7, 8, 9, 10, 14, 17, 18, 19, 21, 23]$

(۱) مرحله □

$$A' = [4, 12, 16]$$

$$B' = [7, 10, 18]$$

(۲) مرحله □

$$V = [(4, 1, A), (7, 1, B), (10, 2, B), (12, 2, A), (16, 3, A), (18, 3, B)]$$

ادغام موازی: مرحله ۱

۲۲

A

2	3	4	6	11	12	13	15	16	20	22	24
---	---	---	---	----	----	----	----	----	----	----	----

B

1	5	7	8	9	10	14	17	18	19	21	23
---	---	---	---	---	----	----	----	----	----	----	----

$$A' = [4, 12, 16]$$

$$B' = [7, 10, 18]$$

A

2	3	4	6	11	12	13	15	16	20	22	24
---	---	---	---	----	----	----	----	----	----	----	----

B

1	5	7	8	9	10	14	17	18	19	21	23
---	---	---	---	---	----	----	----	----	----	----	----

ادغام موازی: مرحله ۲

۲۳

A

2	3	4	6	11	12	13	15	16	20	22	24
---	---	---	---	----	----	----	----	----	----	----	----

B

1	5	7	8	9	10	14	17	18	19	21	23
---	---	---	---	---	----	----	----	----	----	----	----

$$A' = [4, 12, 16]$$

$$B' = [7, 10, 18]$$

$$i=1 \Rightarrow j=1 \Rightarrow v_{1+1-1} = v_1 = (4, 1, A)$$

$$i=1 \Rightarrow j=2 \Rightarrow v_{1+2-1} = v_2 = (7, 1, B)$$

$$i=2 \Rightarrow j=3 \Rightarrow v_{2+3-1} = v_4 = (12, 2, A)$$

$$i=2 \Rightarrow j=2 \Rightarrow v_{2+2-1} = v_3 = (10, 2, B)$$

$$i=3 \Rightarrow j=3 \Rightarrow v_{3+3-1} = v_5 = (16, 3, A)$$

$$i=3 \Rightarrow j=? \Rightarrow v_{3+4-1} = v_6 = (18, 3, B)$$

$$v = [(4, 1, A), (7, 1, B), (10, 2, B), (12, 2, A), (16, 3, A), (18, 3, B)]$$

ادغام موازی: مرحله ۱-۳

۲۴

A

2	3	4	6	11	12	13	15	16	20	22	24
---	---	---	---	----	----	----	----	----	----	----	----

B

1	5	7	8	9	10	14	17	18	19	21	23
---	---	---	---	---	----	----	----	----	----	----	----

$$A' = [4, 12, 16]$$

$$B' = [7, 10, 18]$$

$$v = [(4, 1, A), (7, 1, B), (10, 2, B), (12, 2, A), (16, 3, A), (18, 3, B)]$$

$$Q(1) = (1, 1)$$

$$i=2 \Rightarrow v_{2*2-2} = v_2 = (7, 1, B) \Rightarrow j=5 \Rightarrow Q(2) = (5, 3)$$

$$i=3 \Rightarrow v_{2*3-2} = v_4 = (12, 2, A) \Rightarrow j=7 \Rightarrow Q(3) = (6, 7)$$

$$i=4 \Rightarrow v_{2*4-2} = v_6 = (18, 3, B) \Rightarrow j=10 \Rightarrow Q(4) = (10, 9)$$

ادغام موازی: مرحله M-M

۲۵

A

2	3	4	6	11	12	13	15	16	20	22	24
---	---	---	---	----	----	----	----	----	----	----	----

B

1	5	7	8	9	10	14	17	18	19	21	23
---	---	---	---	---	----	----	----	----	----	----	----

$V = [(4, 1, A), (7, 1, B), (10, 2, B), (12, 2, A), (16, 3, A), (18, 3, B)]$

$$Q(1) = (1, 1)$$

A	2	3	4	6
---	---	---	---	---

B	1	5
---	---	---



C	1	2	3	4	5	6
---	---	---	---	---	---	---

7

$$Q(2) = (5, 3)$$

A	11
---	----

B	7	8	9	10
---	---	---	---	----



C	7	8	9	10	11
---	---	---	---	----	----

12

$$Q(3) = (6, 7)$$

A	12	13	15	16
---	----	----	----	----

B	14	17
---	----	----



C	12	13	14	15	16	17
---	----	----	----	----	----	----

18

$$Q(4) = (10, 9)$$

A	20	22	24
---	----	----	----

B	18	19	21	23
---	----	----	----	----



C	18	19	20	21	21	23	24
---	----	----	----	----	----	----	----

ادغام موازی: تحلیل

۲۶

: (r = s = n) \square زمان اجرا

$$t(2n) = O\left(\frac{n}{N} + \log n\right) \xrightarrow{N=n/\log n} t(2n) = O(\log n)$$

\square تعداد پردازنده ها:

$$p(2n) = N$$

\square هزینه:

$$c(2n) = p(2n) \times t(2n) = O(n + N \log n)$$

$$N \leq \frac{n}{\log n}$$

\square توجه: شرط هزینه بهینه بودن الگوریتم ادغام موازی:

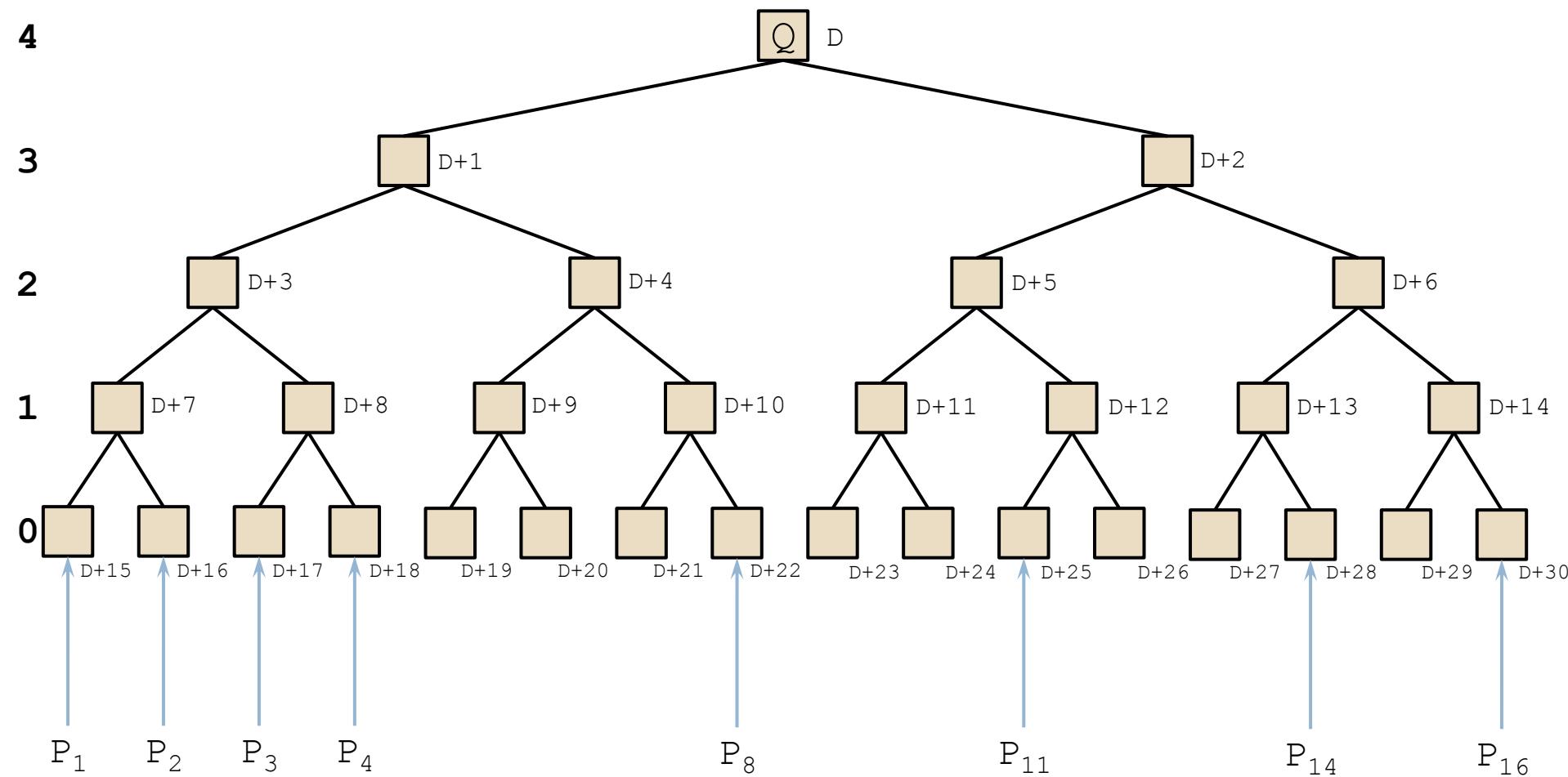
شبیه سازی ادغام موازی برای EREW

۲۷

- الگوریتم پخش چندگانه (Multiple Broadcast)
- شبیه سازی خواندن های همزمان به وسیله چند پردازنده

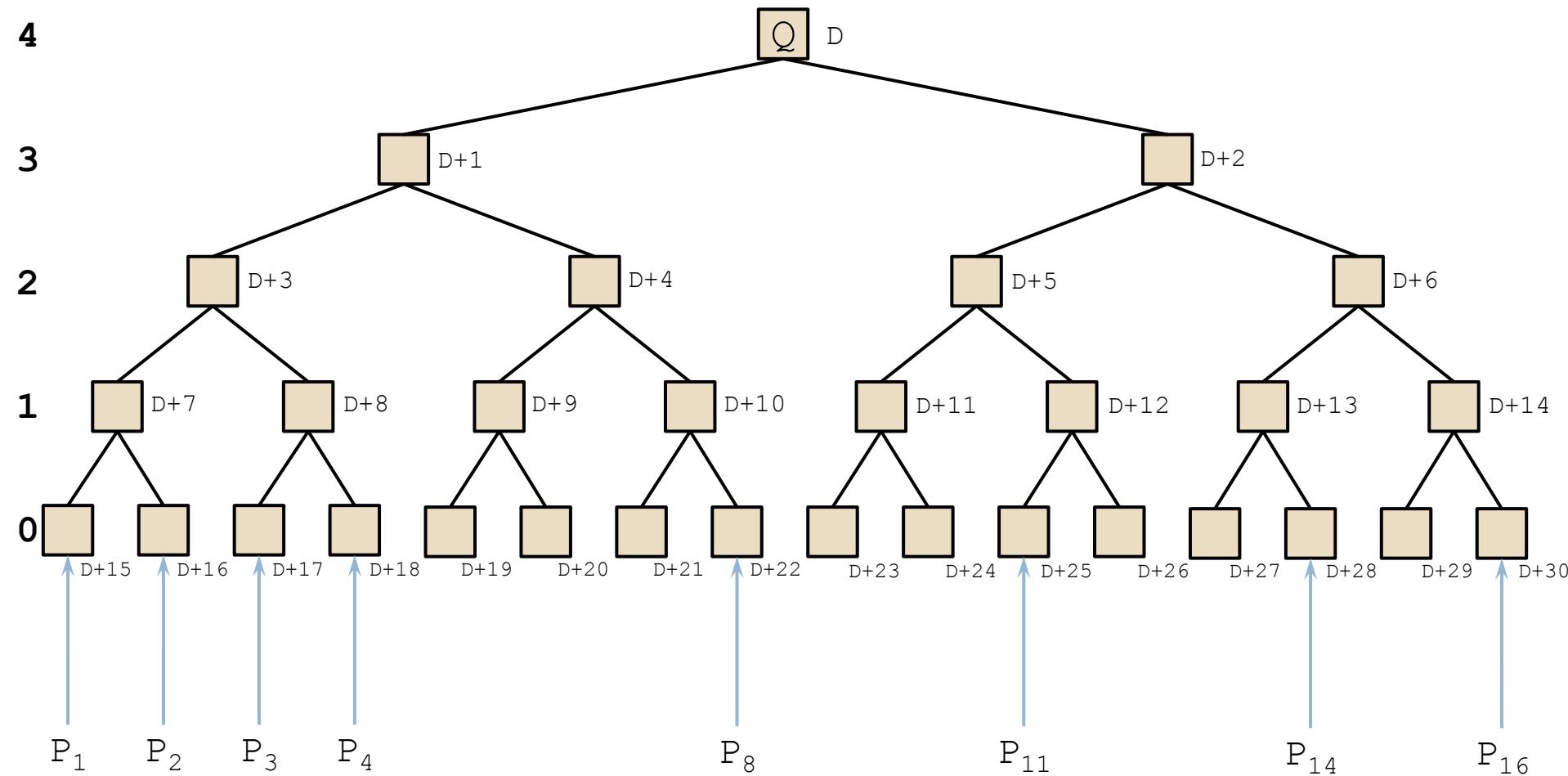
الگوریتم پخش چندگانه (شبیه سازی خواندن های همزمان)

۲۸



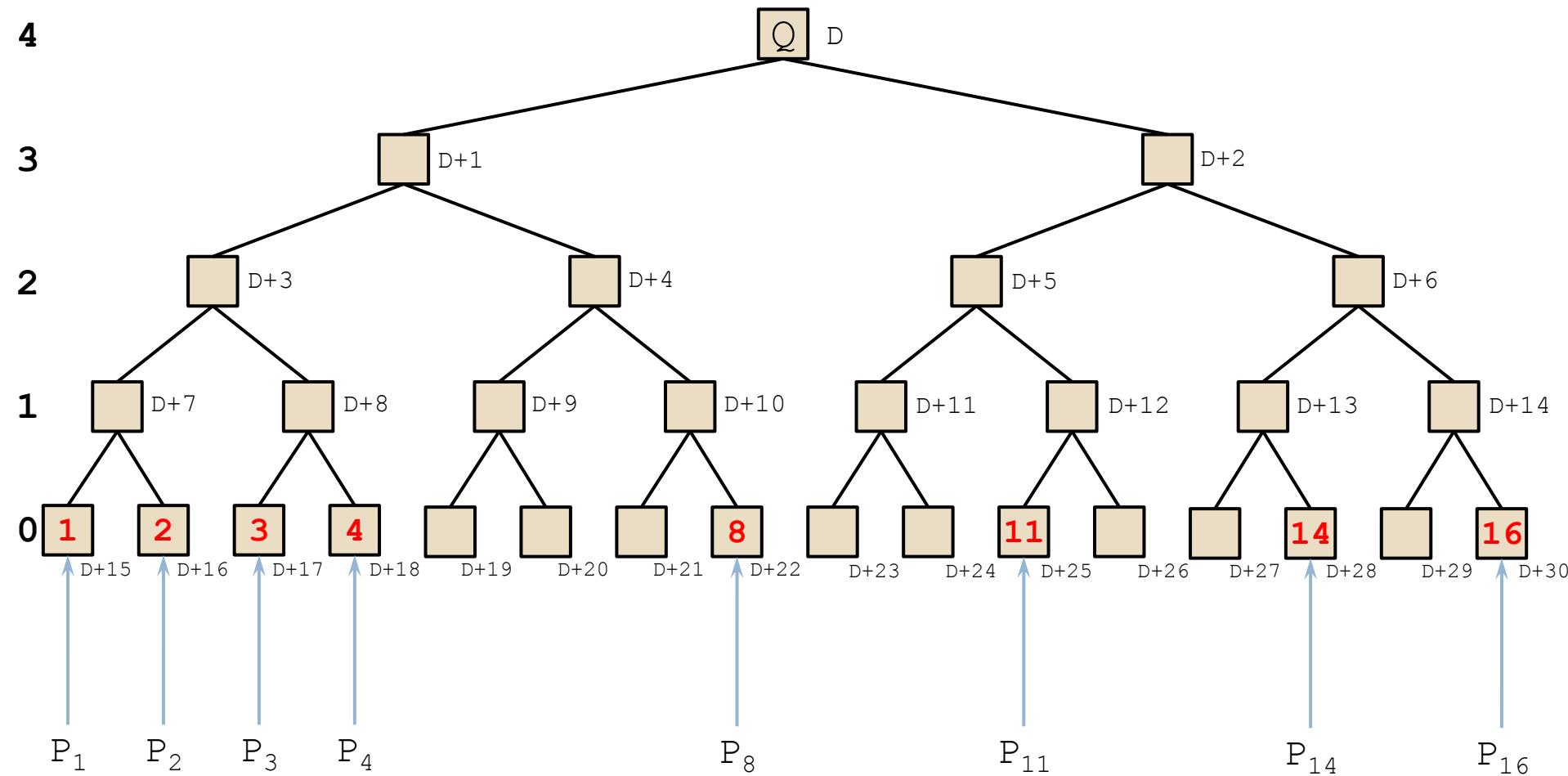
الگوریتم پخش چندگانه (شبیه سازی خواندن های همزمان)

۲۹



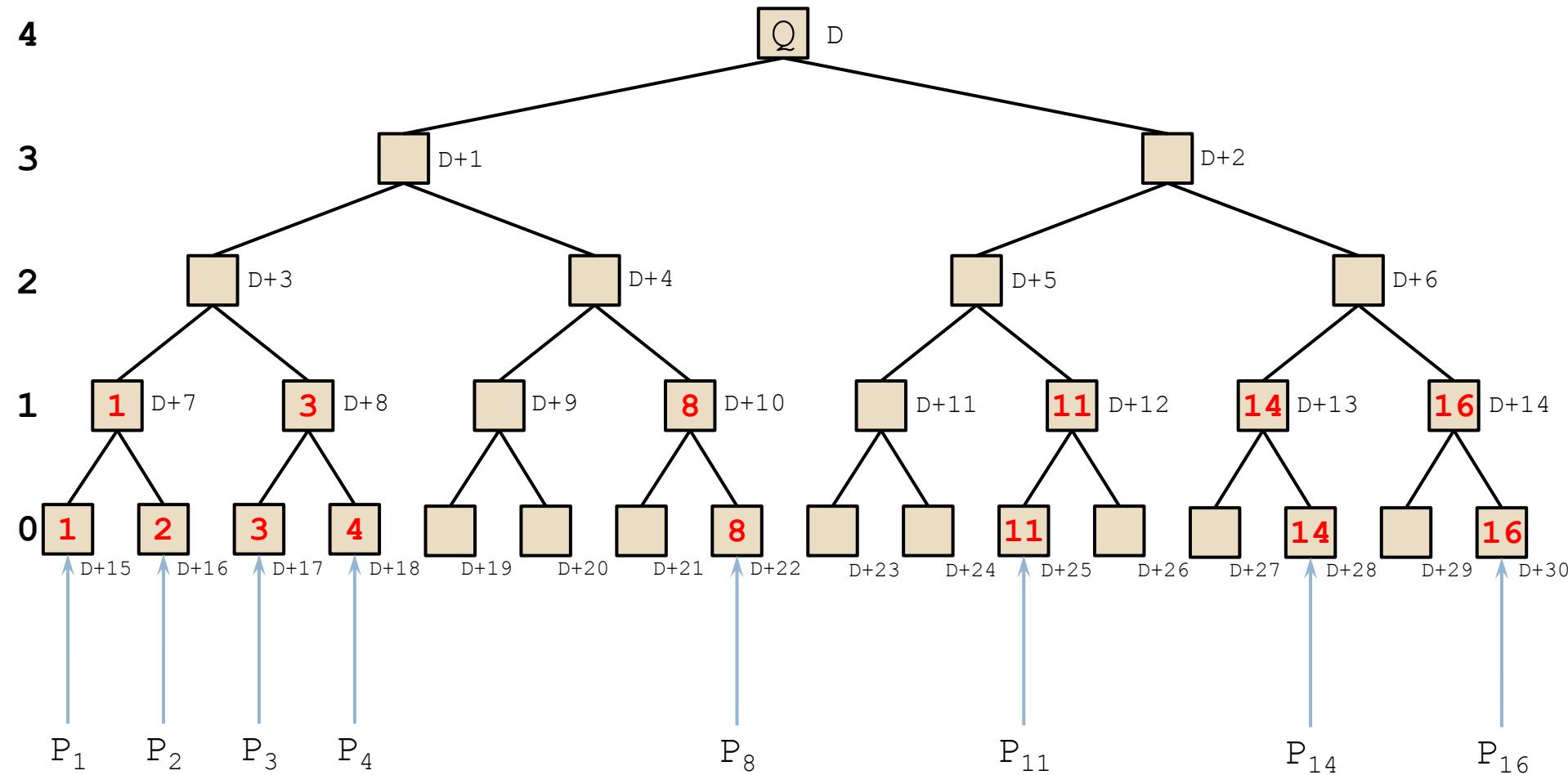
الگوریتم پفس چندگانه: گذر و به بالا (مرحله ا)

۳۰



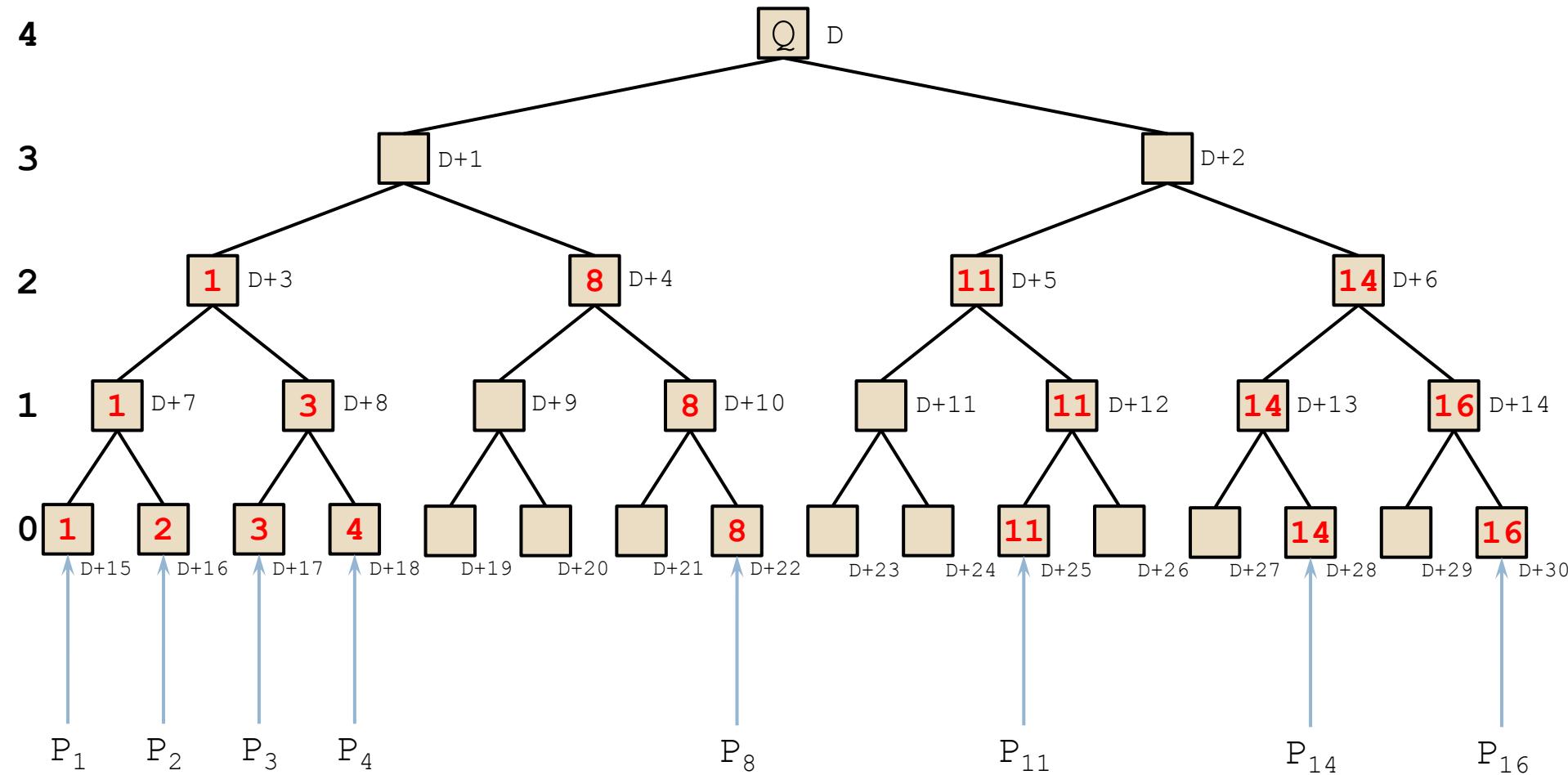
الگوریتم پفس چندگانه: گذر و به بالا (مرحله ۲)

۳۱



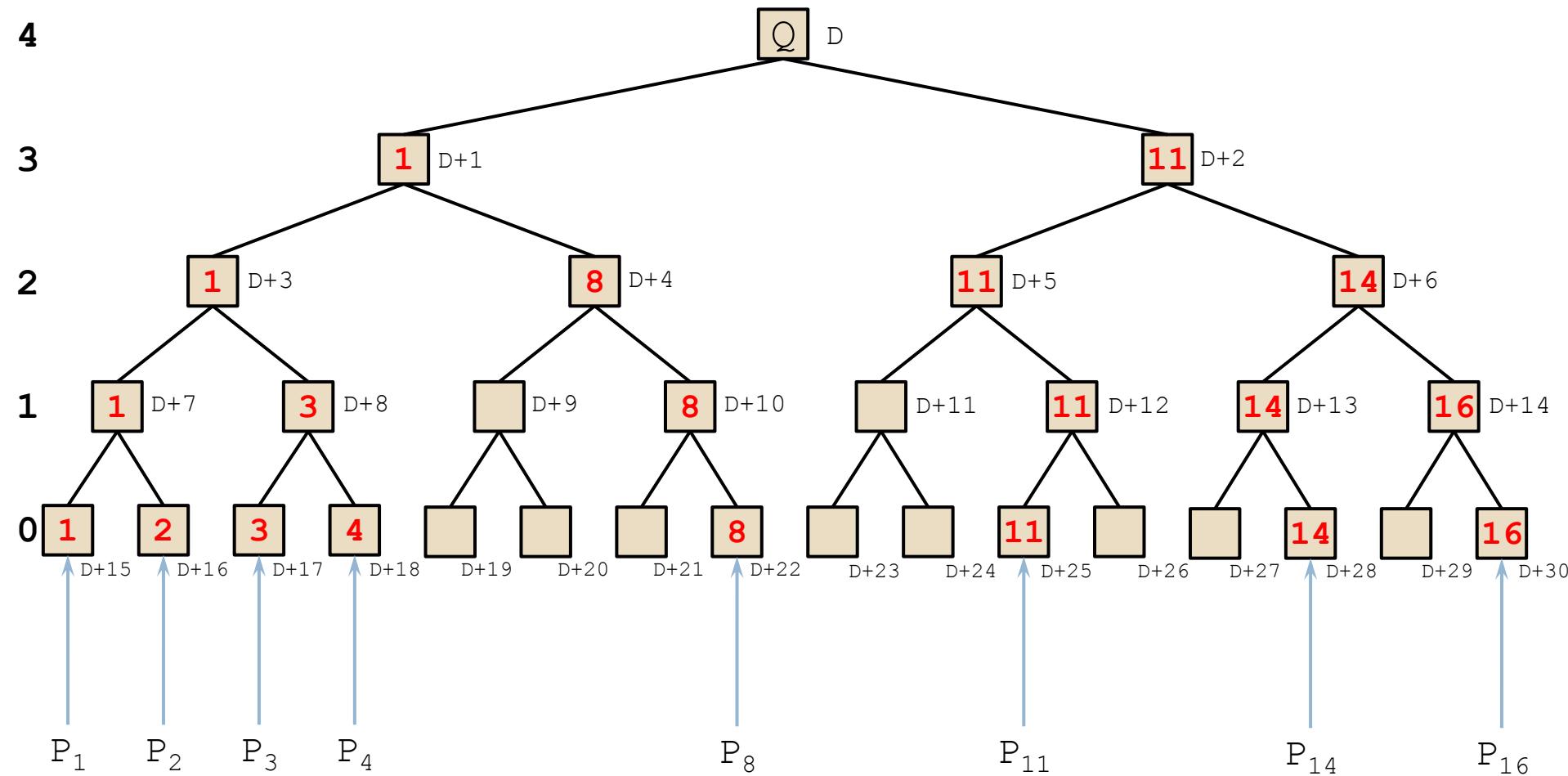
الگوریتم پفس چندگانه: گذر و به بالا (مرحله ۲)

۳۲



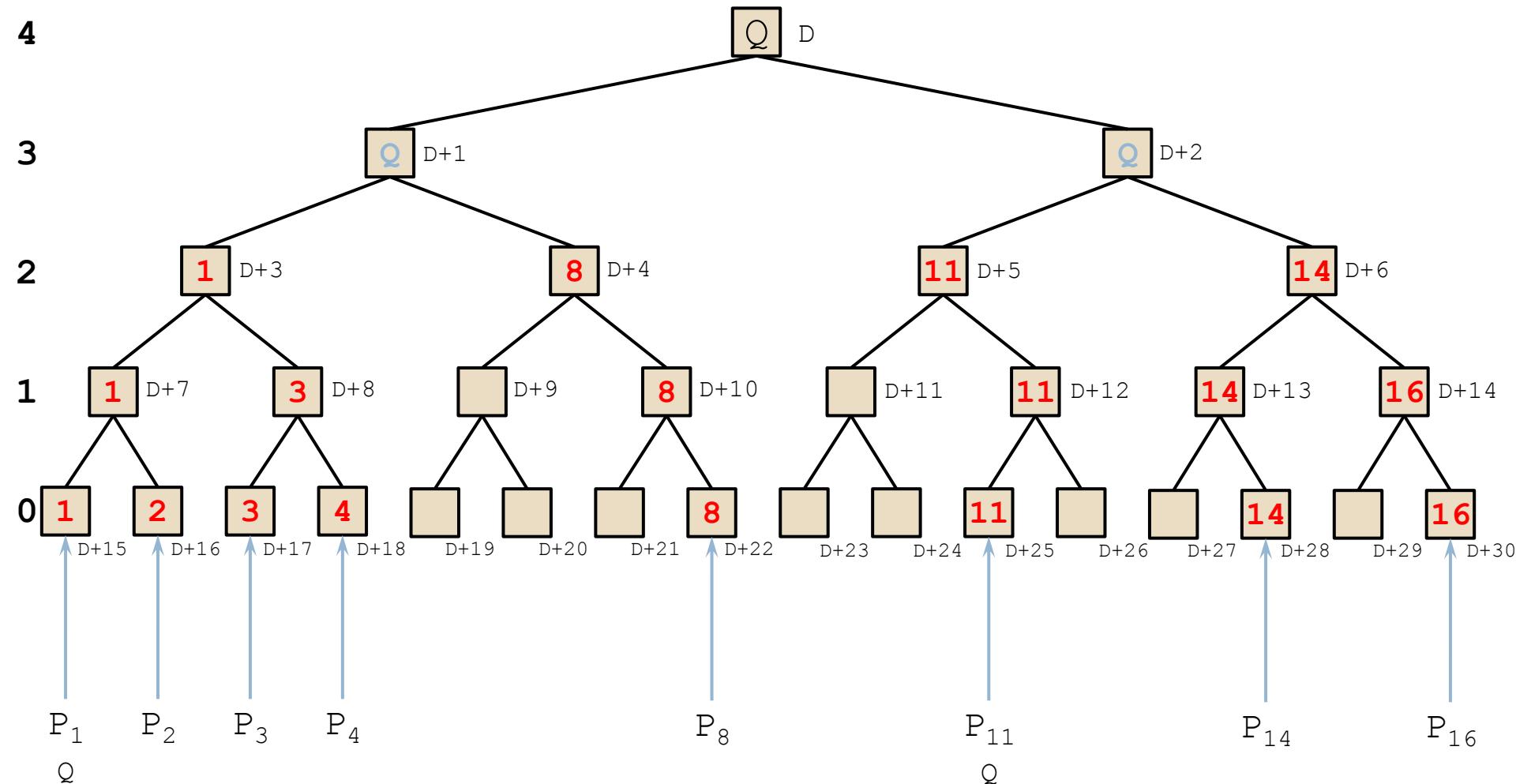
الگوریتم پفس چندگانه: گذر و به بالا (مرحله ۲)

۳۳



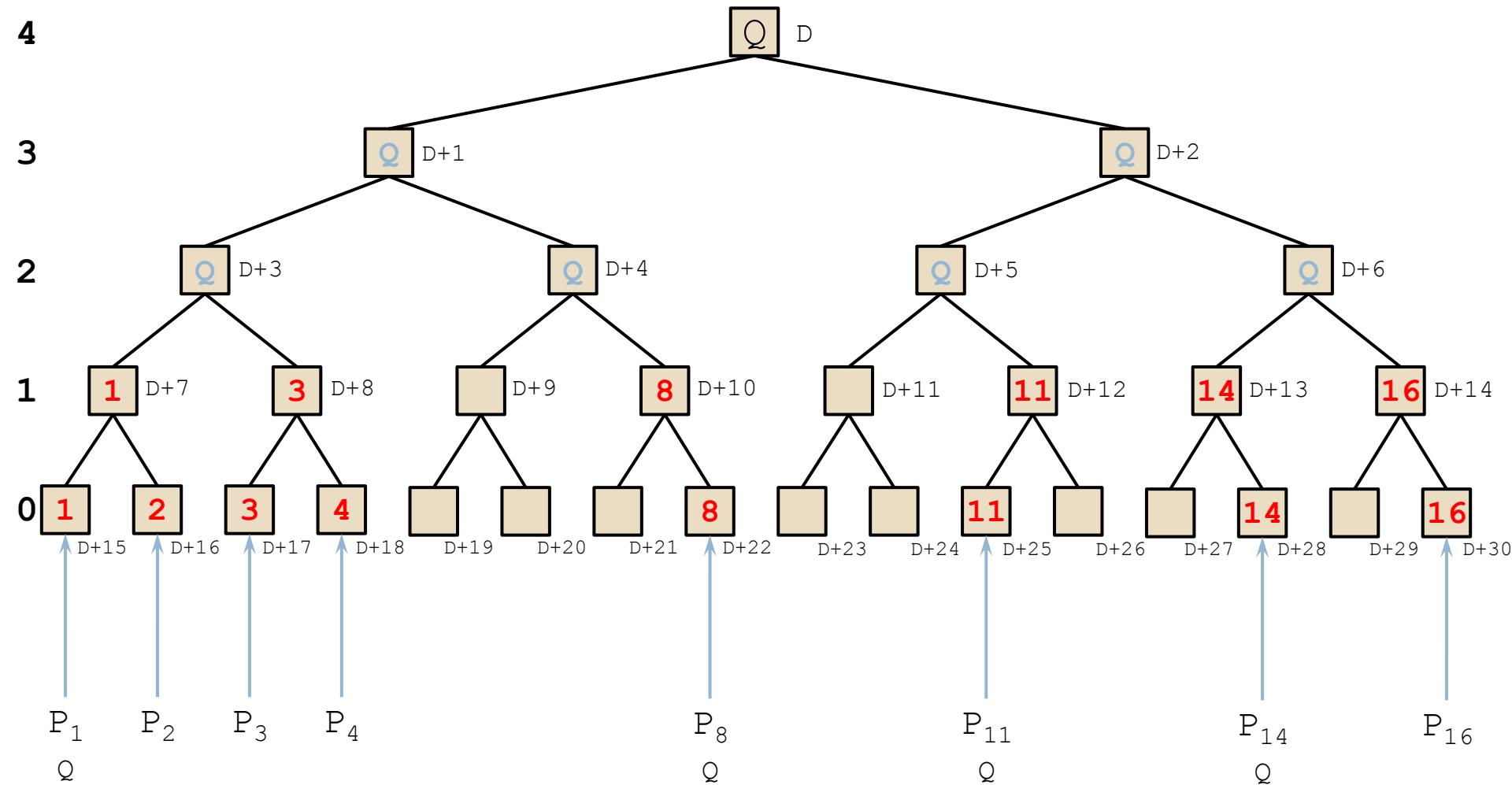
الگوریتم پفس چندگانه: گذر و به پایین (مرحله س)

۳۴



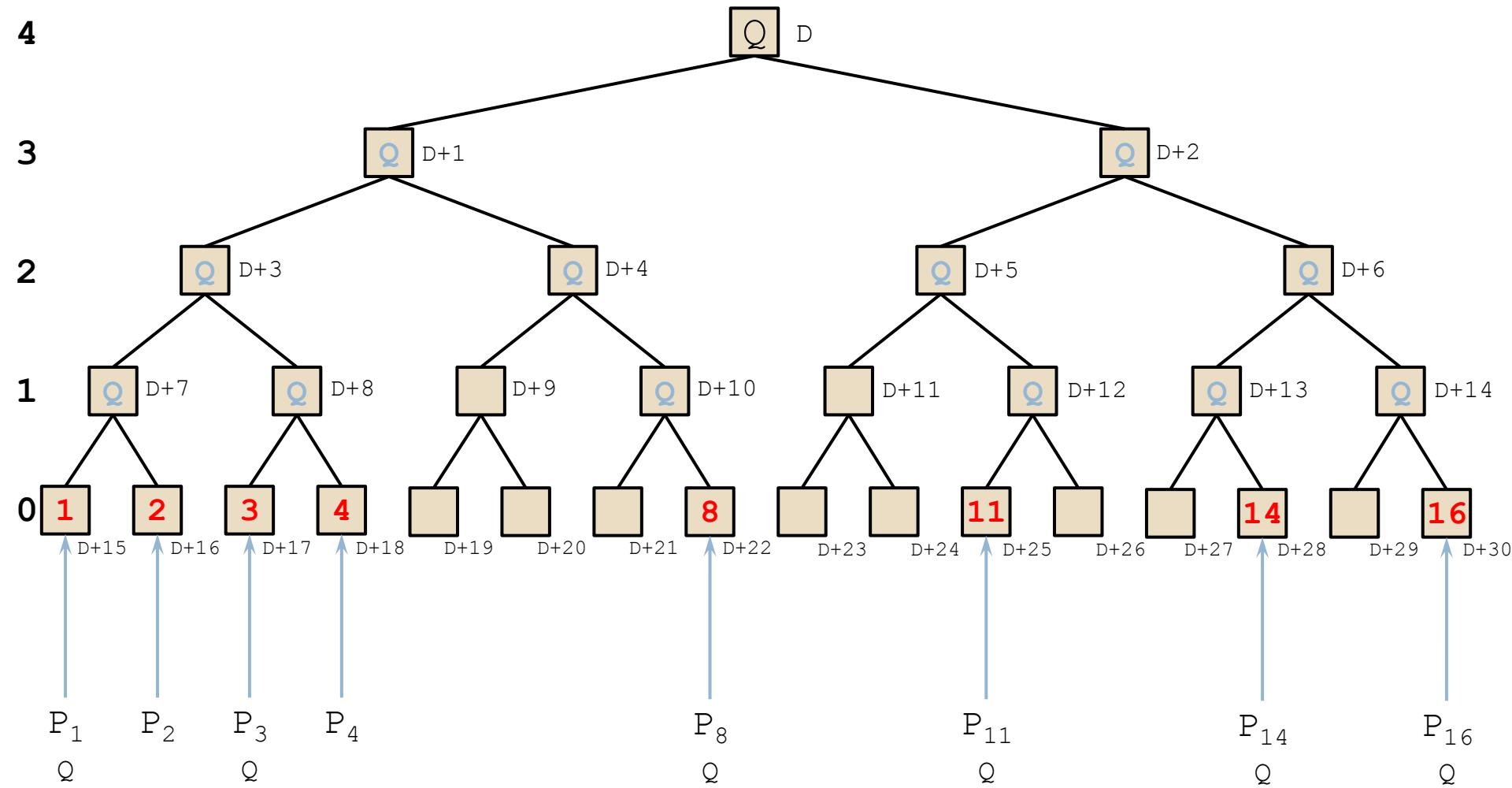
الگوریتم پفس چندگانه: گذر و به پایین (مرحله س)

۳۵



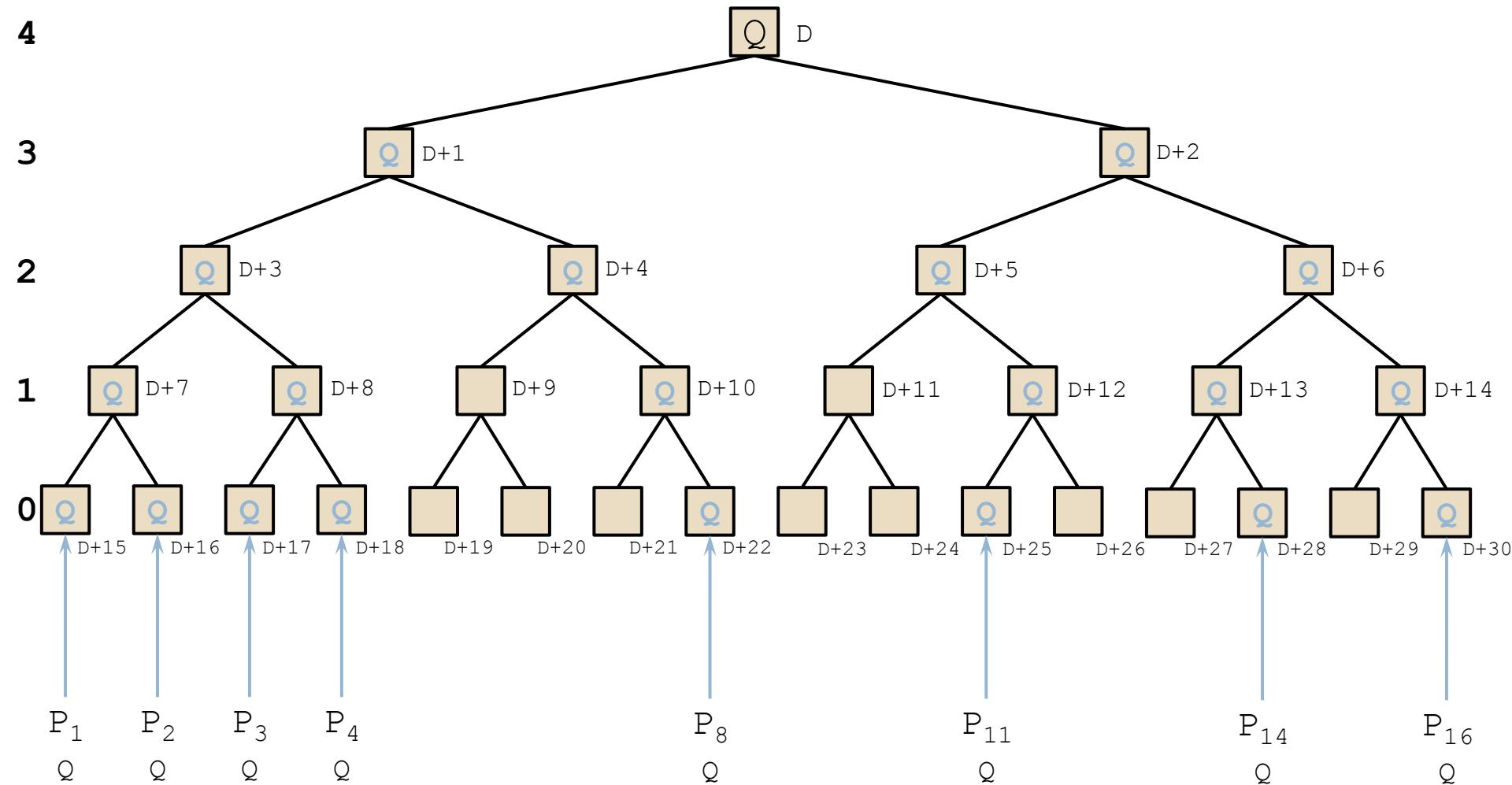
الگوریتم پفس چندگانه: گذر و به پایین (مرحله س)

۳۶



الگوریتم پفس چندگانه: گذر و به پایین (مرحله س)

۳۷



الگوریتم ادغام موازی برای EREW

۳۸

□ تحلیل:

$$t(2n) = O(\log N)O\left(\frac{n}{N} + \log n\right) = O\left(\frac{n}{N} \log N + \log^2 n\right)$$

□ زمان اجرا

$$p(2n) = N$$

□ تعداد پردازنده ها:

$$c(2n) = p(2n) \times t(2n) = O(n \log n + N \log^2 n)$$

□ هزینه:

□ توجه: این الگوریتم هزینه بهینه نیست.

یک الگوریتم هزینه بهینه برای EREW

۳۹

□ سنگ بنای این الگوریتم، یک الگوریتم ترتیبی به منظور یافتن میانه دو دنباله مرتب است.

□ مسأله یافتن میانه دو دنباله مرتب:

□ ورودی: دنباله های مرتب A و B

□ خروجی: میانه دنباله مرتب $A \cdot B$

$$A = [a_1, a_2, \dots, a_r]$$

$$B = [b_1, b_2, \dots, b_s]$$

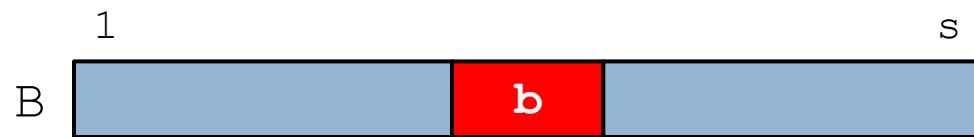
$$A = [2, 5, 7, 10]$$

$$B = [1, 4, 8, 9]$$

□ مثال:

یافتن میانه دو دنباله مرتب

۴۰

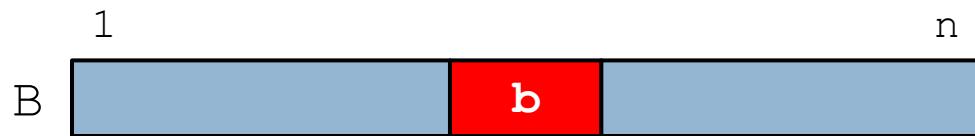


یافتن میانه دو دنباله مرتب

۴۱



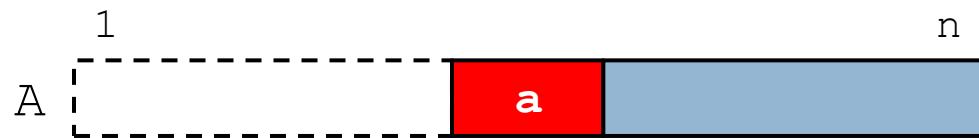
($a = b$) \square حالت اول



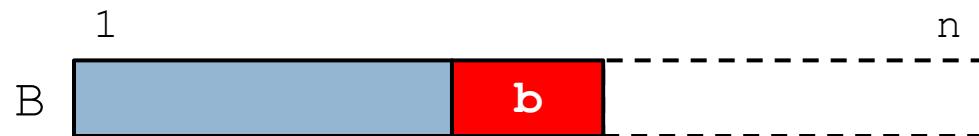
در این حالت میانه همان a است. \square

یافتن میانه دو دنباله مرتب

۴۲

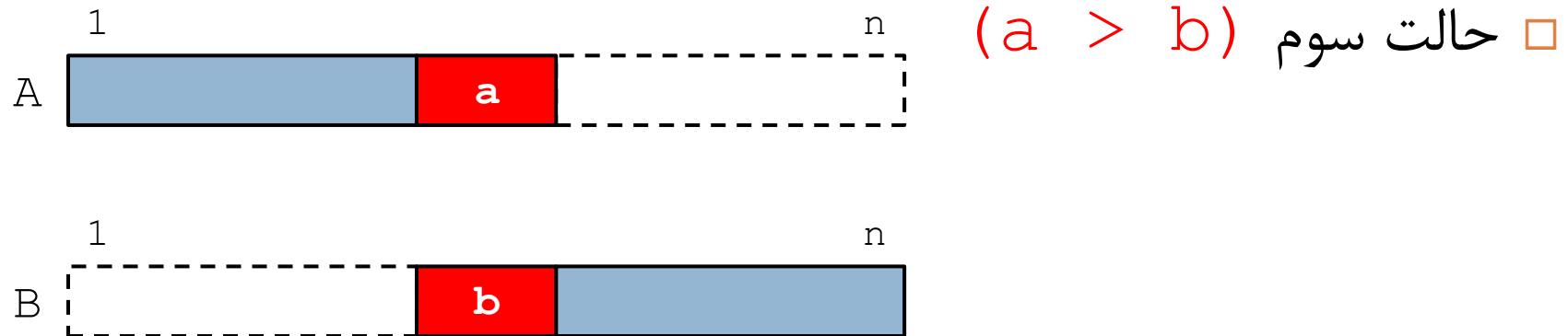


(a < b) \square حالت دوم



یافتن میانه دو دنباله مرتب

۴۳



یافتن میانه دو دنباله مرتب: مثال

۴۴

مثال 

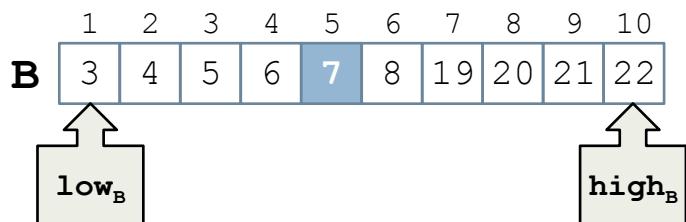
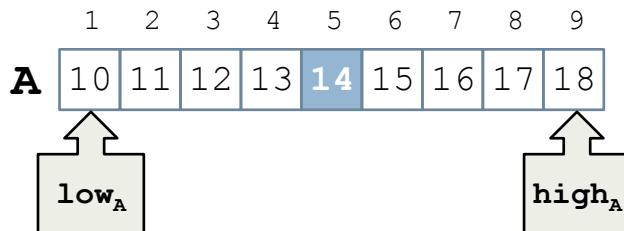
1	2	3	4	5	6	7	8	9	
A	10	11	12	13	14	15	16	17	18

1	2	3	4	5	6	7	8	9	10	
B	3	4	5	6	7	8	19	20	21	22

یافتن میانه دو دنباله مرتب: مثال

۴۵

مثال □

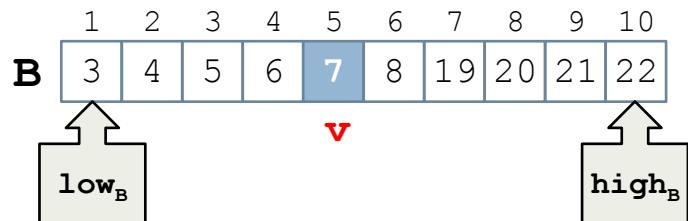
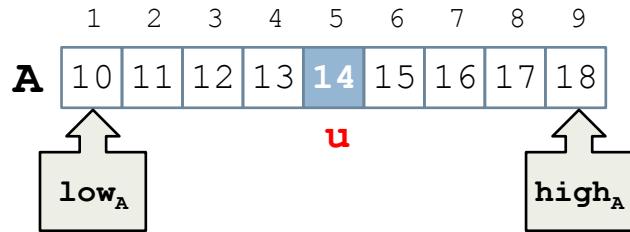


```
lowA ← 1  
lowB ← 1  
highA ← r  
highB ← s  
nA ← r  
nB ← s
```

یافتن میانه دو دنباله مرتب: مثال

۴۶

مثال □



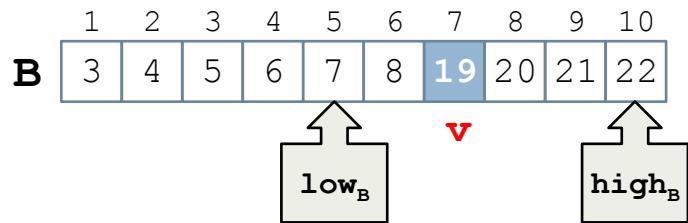
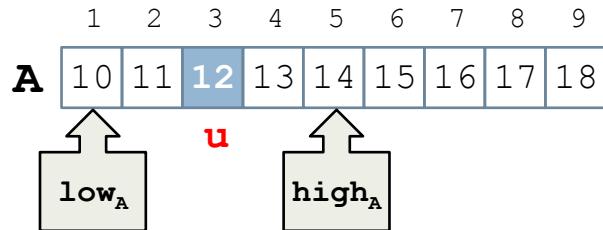
$$w = \min(4, 5) = 4$$

```
while (nA > 1 and nB > 1)
    u ← lowA + ⌈(highA - lowA + 1)/2⌉
    v ← lowB + ⌈(highB - lowB + 1)/2⌉
    w ← min([nA/2], [nB/2])
    nA ← nA - w
    nB ← nB - w
    if (au ≥ bv)
        then highA ← highA - w
              lowB ← lowB + w
    else lowA ← lowA + w
          highB ← highB - w
```

یافتن میانه دو دنباله مرتب: مثال

۴۷

مثال □



$$w = \min(2, 3) = 2$$

```

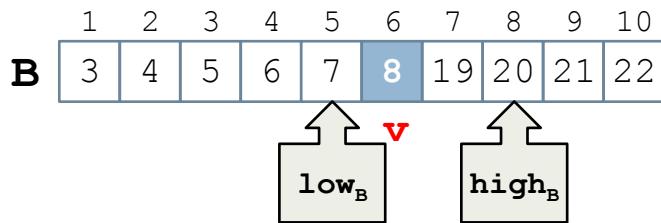
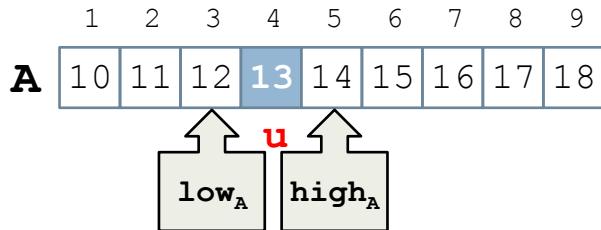
while ( $n_A > 1$  and  $n_B > 1$ )
     $u \leftarrow low_A + \lceil (high_A - low_A + 1) / 2 \rceil$ 
     $v \leftarrow low_B + \lceil (high_B - low_B + 1) / 2 \rceil$ 
     $w \leftarrow \min(\lfloor n_A / 2 \rfloor, \lfloor n_B / 2 \rfloor)$ 
     $n_A \leftarrow n_A - w$ 
     $n_B \leftarrow n_B - w$ 
    if ( $a_u \geq b_v$ )
        then  $high_A \leftarrow high_A - w$ 
                 $low_B \leftarrow low_B + w$ 
        else  $low_A \leftarrow low_A + w$ 
                 $high_B \leftarrow high_B - w$ 

```

یافتن میانه دو دنباله مرتب: مثال

۴۸

مثال □



$$w = \min(1, 2) = 1$$

```

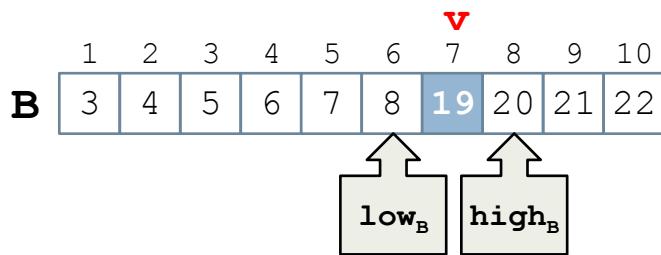
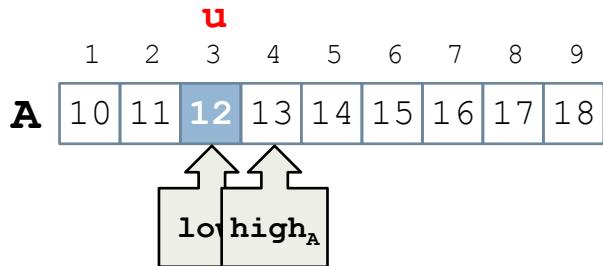
while ( $n_A > 1$  and  $n_B > 1$ )
     $u \leftarrow low_A + \lceil (high_A - low_A + 1) / 2 \rceil$ 
     $v \leftarrow low_B + \lceil (high_B - low_B + 1) / 2 \rceil$ 
     $w \leftarrow \min(\lfloor n_A / 2 \rfloor, \lfloor n_B / 2 \rfloor)$ 
     $n_A \leftarrow n_A - w$ 
     $n_B \leftarrow n_B - w$ 
    if ( $a_u \geq b_v$ )
        then  $high_A \leftarrow high_A - w$ 
                 $low_B \leftarrow low_B + w$ 
        else  $low_A \leftarrow low_A + w$ 
                 $high_B \leftarrow high_B - w$ 

```

یافتن میانه دو دنباله مرتب: مثال

۴۹

مثال □



$$w = \min(1, 1) = 1$$

(a_4 , b_6)

```

while ( $n_A > 1$  and  $n_B > 1$ )
     $u \leftarrow low_A + \lceil (high_A - low_A + 1) / 2 \rceil$ 
     $v \leftarrow low_B + \lceil (high_B - low_B + 1) / 2 \rceil$ 
     $w \leftarrow \min([n_A/2], [n_B/2])$ 
     $n_A \leftarrow n_A - w$ 
     $n_B \leftarrow n_B - w$ 
    if ( $a_u \geq b_v$ )
        then  $high_A \leftarrow high_A - w$ 
                 $low_B \leftarrow low_B + w$ 
        else  $low_A \leftarrow low_A + w$ 
                 $high_B \leftarrow high_B - w$ 

```

O(log n)

سید ناصر (ضوی) - الگوریتم های موازی

الگوریتم سریع ادغام برای EREW

۵۰

□ ویژگی ها:

- تعداد پردازنده ها یک تابع زیرخطی از اندازه ورودی و تطبیقی است.
- بسیار سریع تر از ادغام ترتیبی است و زمان اجرای آن با تعداد پردازنده ها رابطه عکس دارد.
- دارای هزینه بهینه است.

□ در این الگوریتم:

- $1 \leq N \leq r + s,$
- $N = 2^k$

الگوریتم سریع ادغام برای EREW

۵۱

□ مرحله ۱:

□ افزایش هر یک از دنباله های مرتب A و B به N قسمت (شامل قسمت های تنهی)، به طوری که:

$$\blacksquare |A_i| + |B_i| = (r + s) / N$$

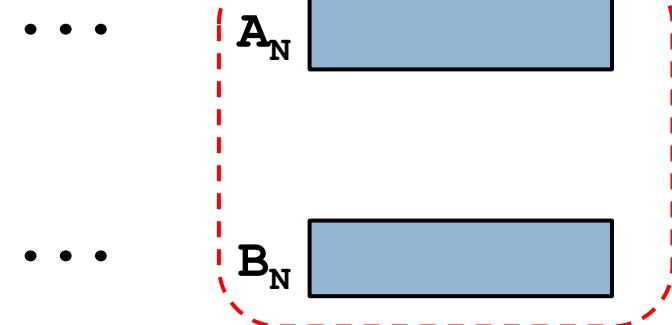
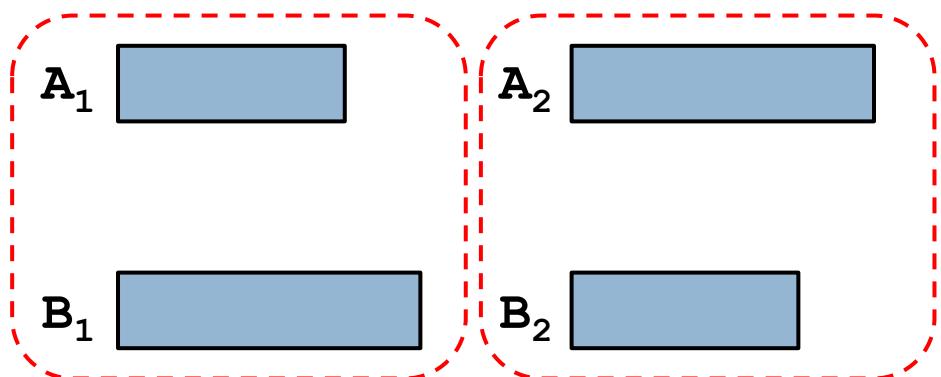
□ تمام عناصر $A_{i+1} \cdot B_{i+1}$ کوچکتر یا مساوی عناصر $A_i \cdot B_i$ هستند.

□ مرحله ۲:

□ ادغام تمام زوج های A_i و B_i به طور همزمان و قرار دادن نتیجه در دنباله C

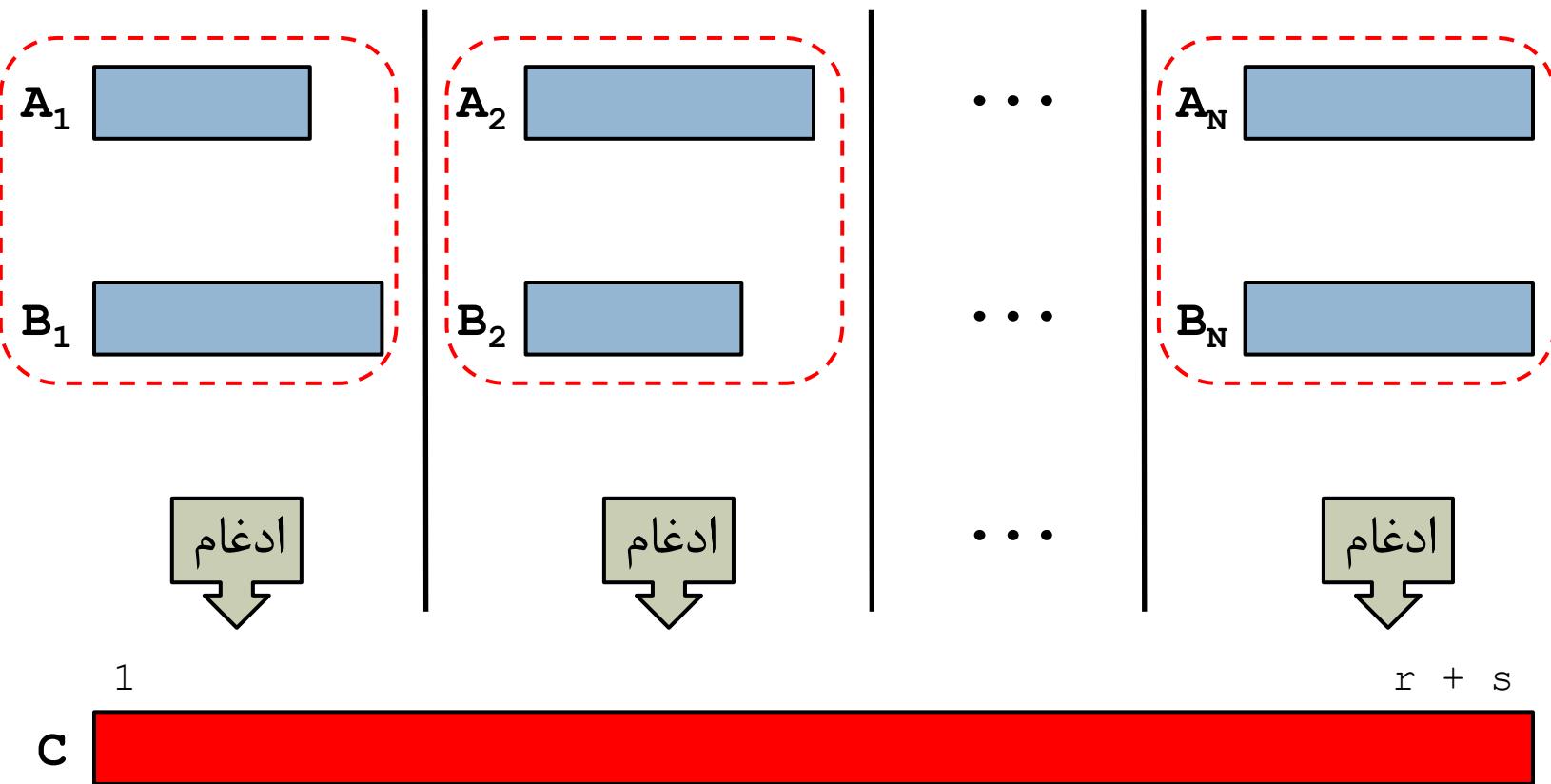
الگوریتم سریع ادغام برای EREW مرحله ۱

۵۲



الگوریتم سریع ادغام برای EREW: مرحله ۲

۵۳



الگوریتم سریع ادغام برای EREW

۵۴

Step 1:

- (1.1) Processor P_1 obtains the quadruple $(1, r, 1, s)$
- (1.2) **for** $j = 1$ **to** $\log N$ **do**
- O(logN · log n)** **for** $i = 1$ **to** 2^{j-1} **do in parallel**
- Processor P_i having received the quadruple (e, f, g, h)
- (1.2.1) {Find the median pair of two sequences}
- O(log n)** TWO-SEQUENCE-MEDIAN($a[e, f], b[g, h], x, y$)
- (1.2.2) {Compute four pointers p_1, p_2, q_1 and q_2 as follows:}
- O(1)** **if** a_x is median
- then** (i) $p_1 \leftarrow x$
- (ii) $q_1 \leftarrow x + 1$
- (iii) **if** $b_y \leq a_x$ **then** (a) $p_2 \leftarrow y$
- (b) $q_2 \leftarrow y + 1$
- else** (a) $p_2 \leftarrow y - 1$
- (b) $q_2 \leftarrow y$
- end if**

الگوریتم سریع ادغام برای EREW

۵۸

```
    else (i)  $p_2 \leftarrow y$ 
              (ii)  $q_2 \leftarrow y + 1$ 
              (iii) if  $a_x \leq b_y$  then (a)  $p_1 \leftarrow x$ 
                                (b)  $q_1 \leftarrow x + 1$ 
              else (a)  $p_1 \leftarrow x - 1$ 
                                (b)  $q_1 \leftarrow x$ 
        end if
    end if
    (1.2.3) Communicate the quadruple  $(e, p_1, g, p_2)$  to  $P_{2i-1}$ 
    (1.2.4) Communicate the quadruple  $(q_1, f, q_2, h)$  to  $P_{2i}$ 
end for
end for
```

الگوریتم سریع ادغام برای EREW

۵۶

Step 2: **for** $i = 1$ **to** N **do in parallel**

O(n / N)

Processor P_i having received the quadruple (a, b, c, d)

$$(2.1) \quad w \leftarrow 1 + ((i - 1)(r + s)) / N$$

$$(2.2) \quad z \leftarrow \min\{i(r + s) / N, (r + s)\}$$

$$(2.3) \quad \text{SEQUENTIAL-MERGE}(A[a, b], B[c, d], C[w, z])$$

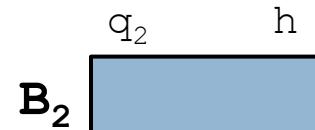
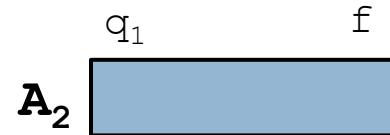
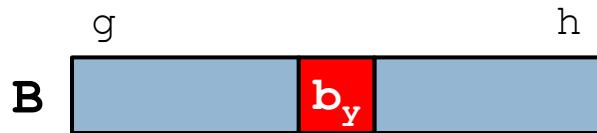
end for

حالت اول)

۵۷

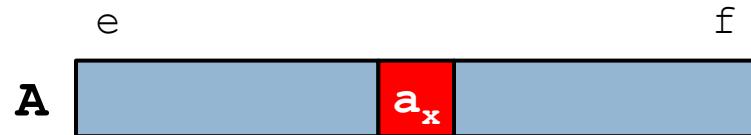


$$(a_x \geq b_y) \quad a_x : \text{میانه}$$

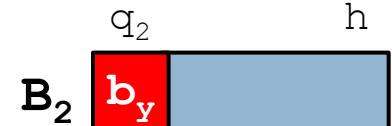
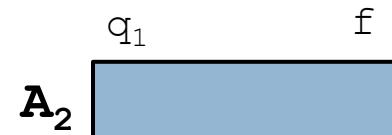
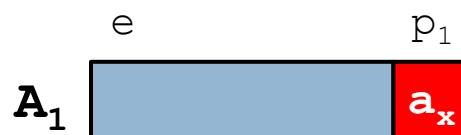
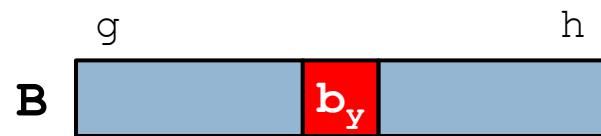


حالت دوچرخه

۵۸



$(a_x < b_y) \quad a_x : \text{میانه}$

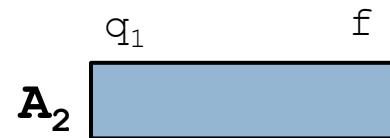
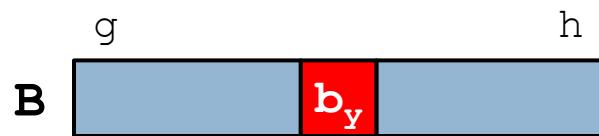


حالت سوم

۵۹



$$(b_y \geq a_x) \quad b_y : \text{میانه}$$

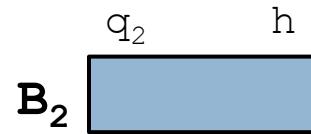
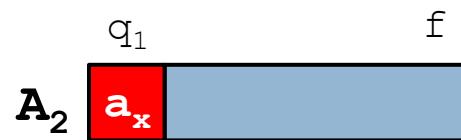
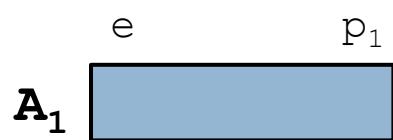
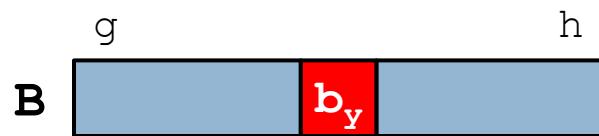


حالات پهلو (۵)

۶۰



$$(b_y < a_x) \quad b_y : \text{میانه}$$



الگوریتم سریع ادغام برای EREW: مثال

۶۱

$N = 4$ □

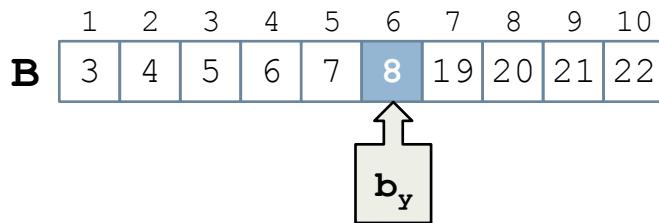
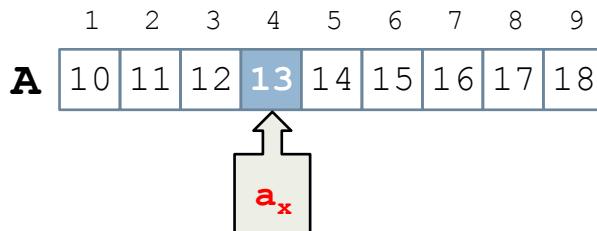
	1	2	3	4	5	6	7	8	9
A	10	11	12	13	14	15	16	17	18

	1	2	3	4	5	6	7	8	9	10
B	3	4	5	6	7	8	19	20	21	22

الگوریتم سریع ادغام برای EREW: مثال

۶۲

P₁

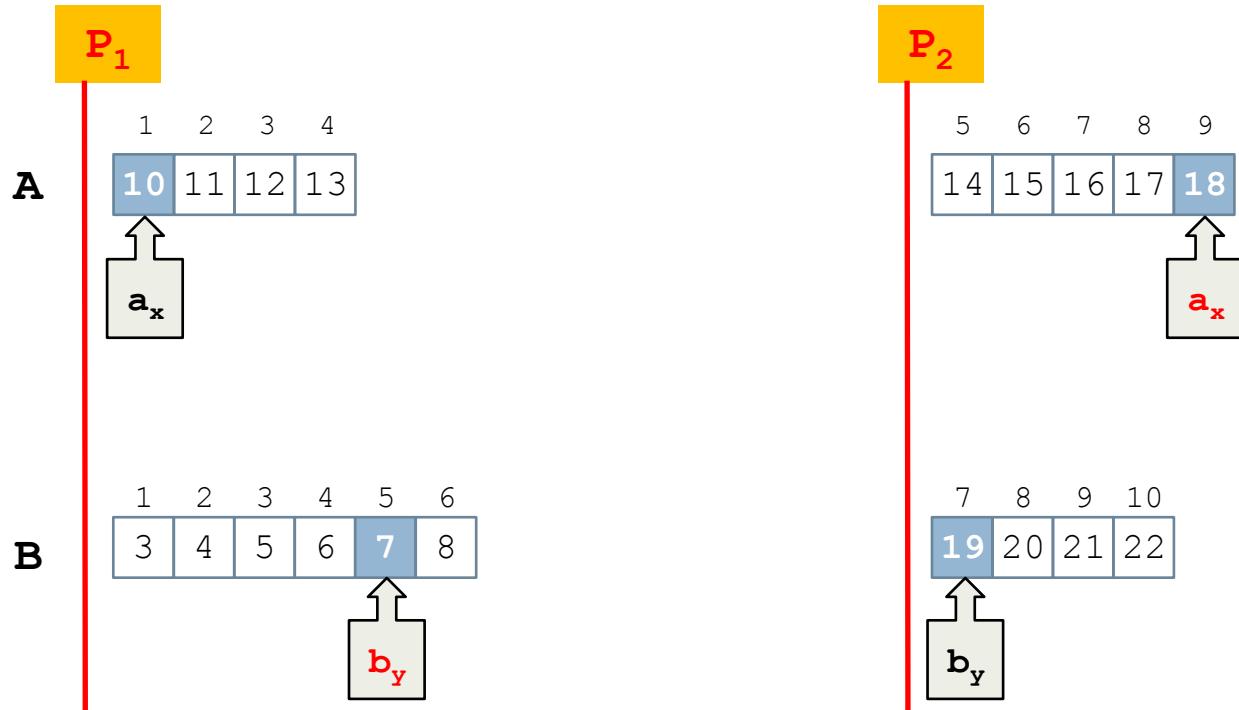


N = 4 تعداد پردازنده ها: □

```
if ax is median  
then (i) p1 ← x  
      (ii) q1 ← x + 1  
      (iii) if by ≤ ax then (a) p2 ← y  
                                         (b) q2 ← y + 1  
      else (a) p2 ← y - 1  
             (b) q2 ← y  
end if
```

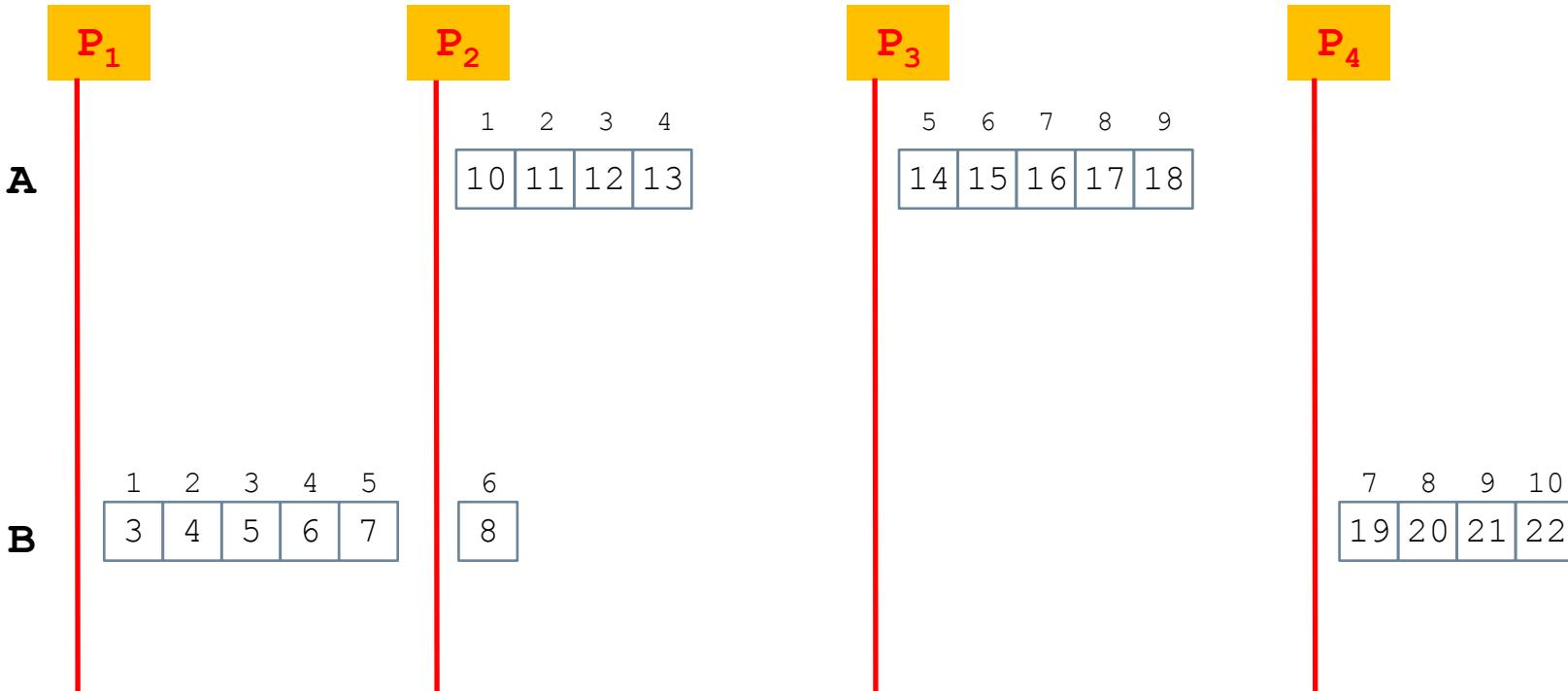
الگوریتم سریع ادغام برای EREW: مثال

۶۳



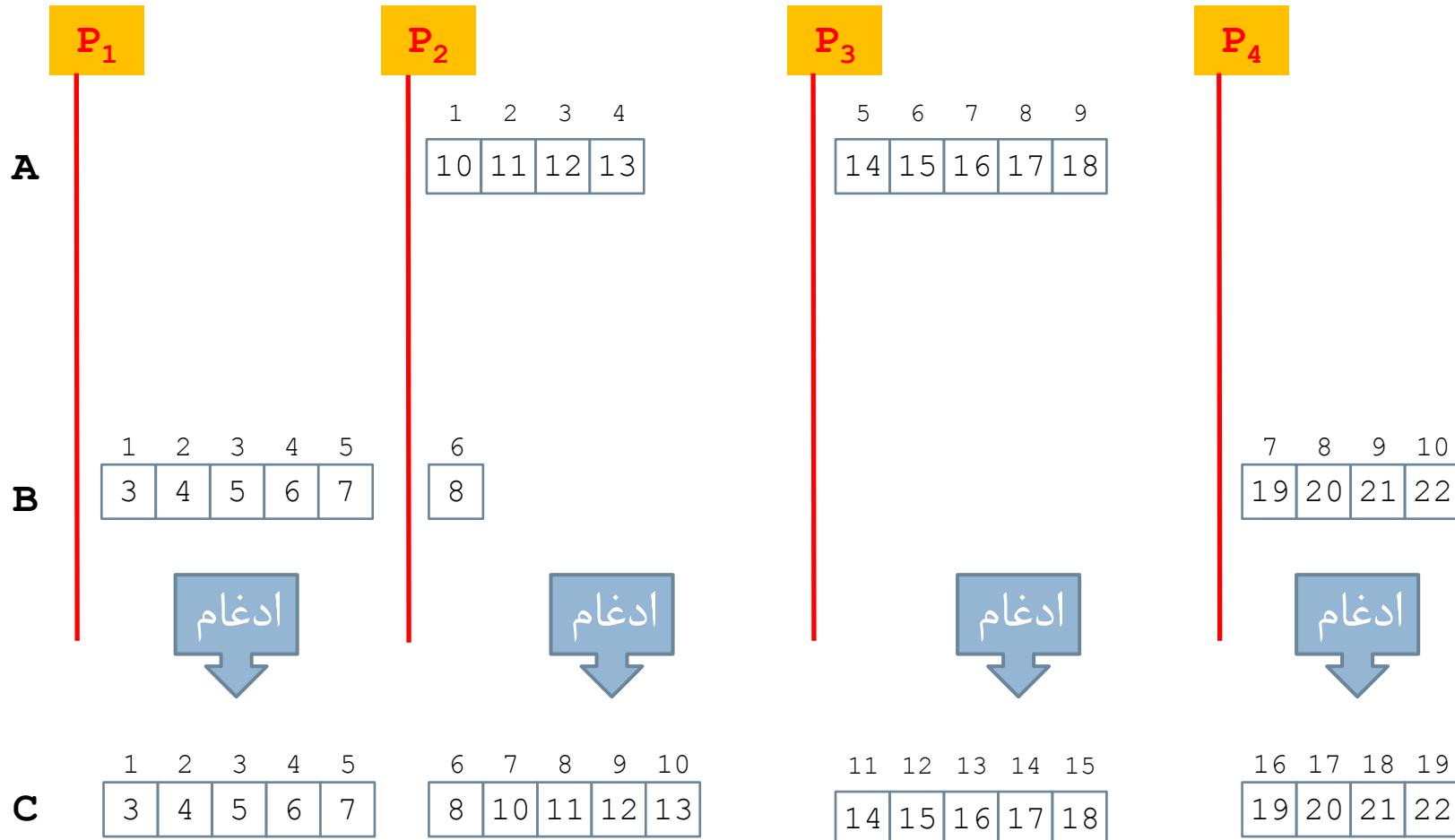
الگوریتم سریع ادغام برای EREW: مثال

۶۴



الگوریتم سریع ادغام برای EREW: مثال

۶۵



الگوریتم سریع ادغام برای تملیل EREW:

۶۶

$$t(2n) = O\left(\frac{n}{N} + \log^2 n\right) \xrightarrow{N=n/\log^2 n} t(2n) = O(\log^2 n)$$

: زمان اجرا \square

تعداد پردازنده ها: \square

$$p(2n) = N$$

$$c(2n) = p(2n) \times t(2n) = O(N + N \log^2 n)$$

: هزینه \square

$$N \leq \frac{n}{\log^2 n}$$

توجه: شرط هزینه بهینه بودن الگوریتم ادغام موازی: \square