

ΠΑΡΑΛΛΗΛΟΙ ΑΛΓΟΡΙΘΜΟΙ

ΕΔΙΠ Μ. Λουκά

Τμήμα Πληροφορικής και Τηλεπικοινωνιών



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών

17 Μαρτίου 2026

Κεφάλαιο 3

Συγχώνευση (Merging)

Έστω

$$A = \{\alpha_1, \alpha_2, \dots, \alpha_r\} \quad B = \{b_1, b_2, \dots, b_s\}$$

ταξινομημένες ακολουθίες αριθμών σε μη φθίνουσα σειρά. Να σχηματισθεί η ακολουθία

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

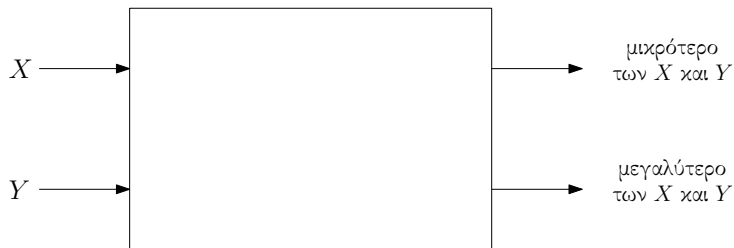
ταξινομημένη σε μη φθίνουσα σειρά. Αν $r = s = n$ (χειρότερη περίπτωση), τότε ο ακολουθιακός αλγόριθμος έχει πολυπλοκότητα $O(n)$, η οποία είναι βέλτιστη ($\Omega(n)$).

Θα μελετηθεί το πρόβλημα της συγχώνευσης σε μια ποικιλία παράλληλων υπολογιστικών μοντέλων. Με βάση το κάτω όριο απαιτείται $\Omega(n/N)$ χρόνος για οποιοδήποτε παράλληλο αλγόριθμο συγχώνευσης που χρησιμοποιεί N επεξεργαστές.

Ένα δίκτυο για συγχώνευση

(r, s) δίκτυο συγχώνευσης: ειδικού σκοπού παράλληλη αρχιτεκτονική.

- όλοι οι επεξεργαστές είναι ίδιοι, απλοί και επικοινωνούν με ένα ειδικού σκοπού δίκτυο (συγκριτές - comparators).



Σχήμα: Συγκριτής (περίπτωση $n = 1$)

Ένα δίκτυο για συγχώνευση

Χρησιμοποιώντας τους συγκριτές μπορούμε να φτιάξουμε ένα δίκτυο που να έχει σαν είσοδο δυο ταξινομημένες ακολουθίες $A = \{a_1, a_2, \dots, a_r\}$ και $B = \{b_1, b_2, \dots, b_s\}$ και να παράγει σαν έξοδο μια απλή ταξινομημένη ακολουθία $C = \{c_1, c_2, \dots, c_{r+s}\}$. Η παρουσίαση απλοποιείται από δυο υποθέσεις.

Υποθέσεις

1. $r = s = n \geq 1$
2. $n = 2^k$

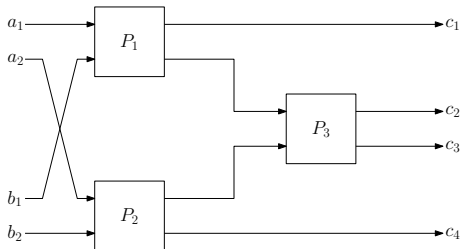
Με τη χρήση των συγκριτών θα κατασκευαστεί ένα δίκτυο.

Για $n = 1$: Αρκεί ένας συγκριτής (γιατί παράγει σαν έξοδο τις 2 εισόδους του ταξινομημένες ανάλογα).

Για $n = 2$: $A = \{\alpha_1, \alpha_2\}$ και $B = \{b_1, b_2\}$.

Για $n = 4$: $A = \{3, 5, 7, 9\}$ και $B = \{2, 4, 6, 8\}$.

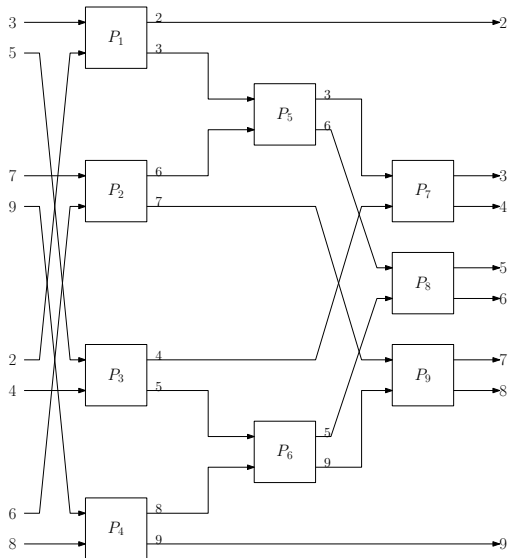
Ένα δίκτυο για συγχώνευση



Σχήμα: Περίπτωση $n = 2$ - Συγχώνευση δύο ακολουθιών με τέσσερα στοιχεία η καθεμία

Ο επεξεργαστής P_1 συγκρίνει τα $\min(A), \min(B)$. Η πάνω έξοδος του P_1 είναι το $\min(C) = c_1$. Όμοια, η κάτω έξοδος του P_2 είναι $\max(C) = c_4$. Μια επιπλέον απλή σύγκριση επιτρέπει στον P_3 να παράγει τα δυο ενδιάμεσα στοιχεία της C .

Ένα δίκτυο για συγχώνευση



Σχήμα: Περίπτωση $n = 3$

Ένα δίκτυο για συγχώνευση

Γενικά ένα (n, n) δίκτυο συγχώνευσης σχηματίζεται από δύο $(n/2, n/2)$ άλλα δίκτυα. Τα στοιχεία στις περιπτώσεις θέσεις των A και B, δηλαδή $\{\alpha_1, \alpha_3, \alpha_5, \dots, \alpha_{n-1}\}$ και $\{b_1, b_3, b_5, \dots, b_{n-1}\}$ συγχωνεύονται στο ένα $(n/2, n/2)$ δίκτυο συγχώνευσης, δημιουργώντας την

$$\{d_1, d_2, d_3, \dots, d_n\}$$

Ταυτόχρονα για τα στοιχεία στις άρτιες θέσεις σχηματίζεται η

$$\{e_1, e_2, \dots, e_n\}$$

με τη χρήση του άλλου $(n/2, n/2)$ δικτύου. Η τελική ακολουθία $\{c_1, c_2, \dots, c_n\}$ λαμβάνεται από odd-even merging

$$c_1 = d_1, \quad c_{2n} = e_n, \quad c_{2i} = \min(d_{i+1}, e_i)$$

και

$$c_{2i+1} = \max(d_{i+1}, e_i), \quad i = 1, 2, \dots, n-1$$

Ένα δίκτυο για συγχώνευση

Στην $\{d_1, d_2, \dots, d_n\}$:

$$i \text{ στοιχεία} \leq d_{i+1} \longrightarrow 2i \text{ στοιχεία των A και B} \leq d_{i+1}$$

ή

$$c_{2i} \leq d_{i+1}$$

όμοια

$$c_{2i} \leq e_i$$

Επίσης

$$d_{i+1} \leq c_{2i+1}$$

και

$$e_i \leq c_{2i+1}$$

Επειδή

$$c_{2i} \leq c_{2i+1}$$

συνεπάγεται

$$c_{2i} = \min(d_{i+1}, e_i), \quad c_{2i+1} = \max(d_{i+1}, e_i)$$

Ένα δίκτυο για συγχώνευση

Παράδειγμα: Έστω $D = \{d_1, d_2, d_3, d_4\} = \{1, 2, 7, 10\}$, οπότε $A_1 = \{1, 7\}$ και $B_1 = \{2, 10\}$ και $E = \{e_1, e_2, e_3, e_4\} = \{3, 4, 8, 9\}$, οπότε $A_2 = \{3, 8\}$, $B_2 = \{4, 9\}$. Τότε

$$d_1 = 1, \quad e_4 = 9$$

και για $i = 1$ έχουμε

$$d_{i+1} = d_2 = 2, \quad e_i = e_1 = 3 \rightarrow \min = 2, \quad \max = 3$$

γιατί $c_{2i} = \min\{d_{i+1}, e_i\} = 2$, $c_{2i+1} = \max\{d_{i+1}, e_i\} = 3$.

Για $i = 2$ έχουμε

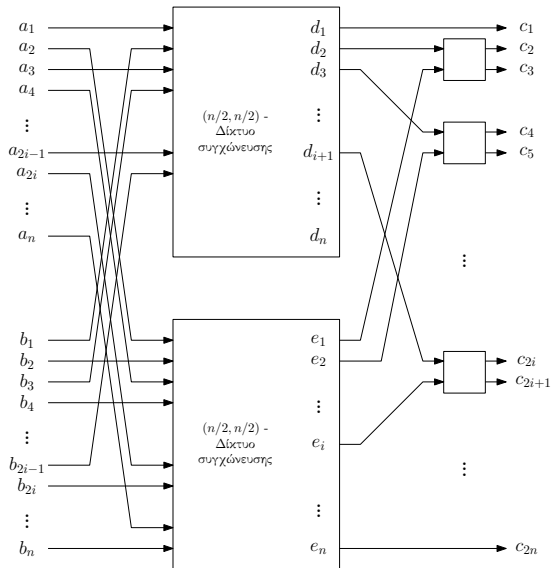
$$d_{i+1} = d_3 = 7, \quad e_i = e_2 = 4 \rightarrow \min = 4, \quad \max = 7.$$

Για $i = 3$ έχουμε

$$d_{i+1} = d_4 = 10, \quad e_i = e_3 = 8 \rightarrow \min = 8, \quad \max = 10.$$

Αρα, $C = \{1, 2, 3, 4, 7, 8, 9, 10\}$

Παράλληλη συγχώνευση



Σχήμα: Συγχώνευση odd-even

SEQUENTIAL MERGE

Διαδικασία 1: SEQUENTIAL MERGE (A,B,C)

- Βήμα 1

(1.1) $i \leftarrow 1$

(1.2) $j \leftarrow 1, \quad a_{r+1} \leftarrow \infty, \quad b_{s+1} \leftarrow \infty$

- Βήμα 2

για $k = 1$ έως $r + s$ κάνε

 αν $a_i < b_j$ τότε

 (i) $c_k \leftarrow a_i$

 (ii) $i \leftarrow i + 1$

 αλλιώς

 (i) $c_k \leftarrow b_j$

 (ii) $j \leftarrow j + 1$

 τέλος

τέλος

SEQUENTIAL MERGE

Η συνάρτηση παίρνει σαν είσοδο τις ακολουθίες A και B και επιστρέφει την ακολουθία C στην έξοδο. Αφού κάθε σύγκριση οδηγεί στον ορισμό ενός στοιχείου της C , υπάρχουν ακριβώς $r + s$ τέτοιες συγκρίσεις, και στη χειρότερη περίπτωση όταν $r = s = n$ ο αλγόριθμος τρέχει σε $O(n)$ χρόνο. Αφού το $\Omega(n)$ είναι το κάτω όριο συγχώνευσης, η SEQUENTIAL MERGE είναι βέλτιστη!

Παράλληλη συγχώνευση

Υποθέτουμε ότι έχουμε στη διάθεσή μας $P_1, P_2, \dots, P_N$ επεξεργαστές, $N \leq r \leq s$ και στη χειρότερη περίπτωση όπου $r = s = n$ ο αλγόριθμος τρέχει σε $O((n/N) + \log n)$ χρόνο. Επομένως ο αλγόριθμος είναι βέλτιστου κόστους για $N \leq n / \log n$. Επίσης κάθε επεξεργαστής είναι σε θέση να εκτελεί τις εξής ακολουθιακές διαδικασίες:

1. Τη διαδικασία SEQUENTIAL MERGE
2. Τη διαδικασία BINARY SEARCH

BINARY SEARCH

Διαδικασία 2: BINARY SEARCH (S,x,k)

- Βήμα 1

(1.1) $i \leftarrow 1$

(1.2) $h \leftarrow n$

(1.3) $k \leftarrow 0$

- Βήμα 2

όσο $i \leq h$ κάνε

 (2.1) $m \leftarrow \lfloor (i + h)/2 \rfloor$

 (2.2) αν $x = s_m$ τότε

 (i) $k \leftarrow m$

 (ii) $i \leftarrow h + 1$

αλλιώς

 αν $x < s_m$ τότε

$h \leftarrow m - 1$

αλλιώς

$i \leftarrow m + 1$

τέλος

τέλος

τέλος

Πολυπλοκότητα: $O(\log n)$

Διαδικασία 3: CREW MERGE (A, B, C)

• Βήμα 1

{Επίλεξε $N - 1$ στοιχεία της A που υποδιαιρούν την ακολουθία αυτή σε N υποακολουθίες του ίδιου περίπου μεγέθους. Ονόμασε την υποακολουθία που δημιουργήθηκε από αυτά τα $N - 1$ στοιχεία A' . Δημιούργησε μια υποακολουθία B' από $N - 1$ στοιχεία της B με τον ίδιο τρόπο. Η εκτέλεση του βήματος αυτού είναι η ακόλουθη}:

για $i = 1$ έως $N - 1$ κάνε παράλληλα

 Ο επεξεργαστής P_i υπολογίζει το a'_i και b'_i από

 (1.1) $a'_i \leftarrow a_{i \lceil r/N \rceil}$

 (1.2) $b'_i \leftarrow b_{i \lceil s/N \rceil}$

τέλος

• Βήμα 2

{Συγχώνευσε τις A' και B' σε μια ακολουθία τριάδων

$V = \{v_1, v_2, \dots, v_{2N-2}\}$, όπου κάθε τριάδα αποτελείται από ένα στοιχείο της A' ή B' , ακολουθούμενο από τη θέση του στην A' ή B' , ακολουθούμενο από το όνομα της αρχικής ακολουθίας, δηλαδή, της A ή B . Αυτό γίνεται με τον ακόλουθο τρόπο:}

Συγχώνευση CREW SM SIMD

(2.1) **για** $i = 1$ **έως** $N - 1$ **κάνε παράλληλα**

(i) Ο επεξεργαστής P_i χρησιμοποιεί την BINARY SEARCH στη B' για να βρεί το μικρότερο j έτσι ώστε $a'_i < b'_j$

(ii) **αν υπάρχει** j **τότε**

| $v_{i+j-1} \leftarrow (a'_i, i, A)$

αλλιώς

| $v_{i+N-1} \leftarrow (a'_i, i, A)$

τέλος

τέλος

(2.2) **για** $i = 1$ **έως** $N - 1$ **κάνε παράλληλα**

(i) Ο επεξεργαστής P_i χρησιμοποιεί την BINARY SEARCH στην A' για να βρεί το μικρότερο j έτσι ώστε $b'_i < a'_j$

(ii) **αν υπάρχει** j **τότε**

| $v_{i+j-1} \leftarrow (b'_i, i, B)$

αλλιώς

| $v_{i+N-1} \leftarrow (b'_i, i, B)$

τέλος

τέλος

• Βήμα 3

{Κάθε επεξεργαστής συγχωνεύει και εισάγει στην C τα στοιχεία δύο υποακολουθιών, ένα από την A και ένα από την B . Οι δείκτες των δύο στοιχείων (ένα από την A και ένα από την B) από όπου κάθε επεξεργαστής πρόκειται να αρχίσει τη συγχώνευση, υπολογίζονται πρώτα και αποθηκεύονται σε έναν πίνακα Q από ταξινομημένα ζεύγη. Το βήμα αυτό εκτελείται ως εξής}:

Συγχώνευση CREW SM SIMD

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

(3.2) **για** $i = 2$ **έως** N **κάνε παράλληλα**

αν $v_{2i-2} = (a'_k, k, A)$ **τότε**

ο επεξεργαστής P_i

(i) χρησιμοποιεί την BINARY SEARCH στο B για να βρεί το μικρότερο j

τέτοιο ώστε $b_j > a'_k$

(ii) $Q(i) \leftarrow (k \lceil r/N \rceil, j)$

αλλιώς

ο επεξεργαστής P_i

(i) χρησιμοποιεί την BINARY SEARCH στο A για να βρεί το μικρότερο

j τέτοιο ώστε $a_j > b'_k$

(ii) $Q(i) \leftarrow (j, k \lceil s/N \rceil)$

τέλος

τέλος

(3.3) για $i = 1$ έως N κάνε παράλληλα

Ο επεξεργαστής P_i χρησιμοποιεί την SEQUENTIAL MERGE και τον $Q(i) = (x, y)$ για να συγχωνεύσει δύο υποακολουθίες, η μία ξεκινώντας από το a_x και η άλλη από το b_y και τοποθετεί τα αποτελέσματα της συγχώνευσης στον πίνακα C αρχίζοντας στη θέση $x + y - 1$. Η συγχώνευση συνεχίζεται μέχρις ότου

- (i) βρεθεί ένα στοιχείο μεγαλύτερο ή ίσο με το πρώτο στοιχείο του v_{2i} , σε κάθε μία από τις A και B (όταν $i \leq N - 1$)
- (ii) δεν υπάρχει κανένα στοιχείο στην A ή B (όταν $i = N$)

τέλος

Πειραματική ερμηνεία υπολογισμού των Q :

- Από το V χρησιμοποιούμε μόνο τις ζυγές (θέσεις) παρενθέσεις. Όταν έχω B στο V ψάχνω να βρω μεγαλύτερο στοιχείο από αυτό που έχω στο A , οπότε $Q(\text{θέση}, k * s/N)$
- Όταν έχω A στο V ψάχνω να βρω μεγαλύτερο στοιχείο από αυτό που έχω στο B , οπότε $Q(k * r/N, \text{θέση})$, όπου k είναι το i στα v_i , δηλ. στο $(7,1,B)$ είναι $k = 1$, στο $(12,2,A)$ είναι $k = 2$ και στο $(18,3,B)$ είναι $k = 3$.

Ανάλυση

Βήμα 1: Κάθε επεξεργαστής υπολογίζει δύο δείκτες. Συνεπώς το βήμα αυτό απαιτεί σταθερό χρόνο $O(1)$.

Βήμα 2: Το βήμα αυτό αποτελείται από δύο εφαρμογές της BINARY SEARCH σε μία ακολουθία με $N - 1$ στοιχεία. Κάθε εφαρμογή της BINARY SEARCH ακολουθείται από μία εντολή καταχώρησης. Συνεπώς απαιτείται $O(\log N)$ χρόνος.

Βήμα 3:

- Το βήμα (3.1) αποτελείται από μία καταχώρηση σταθερού χρόνου
- το βήμα (3.2) απαιτεί το πολύ $O(\log s)$ χρόνο (με $r \leq s$)
- Για το βήμα (3.3) παρατηρούμε ότι η V περιέχει $2N - 2$ στοιχεία τα οποία διαιρούν τη C σε $2N - 1$ υποακολουθίες με μέγιστο μέγεθος ίσο με $\lceil r/N \rceil + \lceil s/N \rceil$
 - ▶ Το μέγιστο μέγεθος αυτό προκύπτει αν, για παράδειγμα, ένα στοιχείο a' της A' είναι ίσο με ένα στοιχείο b' της B'
 - ▶ Τότε τα $\lceil r/N \rceil$ στοιχεία μικρότερα ή ίσα του a'_i (και μεγαλύτερα ή ίσα του a'_{i-1}) είναι επίσης μικρότερα ή ίσα του b'_j
 - ▶ Όμοια, τα $\lceil s/N \rceil$ στοιχεία μικρότερα ή ίσα του b'_j (και μεγαλύτερα του b'_{j-1}) είναι επίσης μικρότερα ή ίσα του a'_i

Συγχώνευση CREW SM SIMD

Στο βήμα αυτό κάθε επεξεργαστής δημιουργεί δύο τέτοιες υποακολουθίες της C των οποίων το ολικό μέγεθος δεν είναι μεγαλύτερο από $2(\lceil r/N \rceil + \lceil s/N \rceil)$, εκτός του P_N , ο οποίος δημιουργεί μόνο μία υποακολουθία της C . Συνεπώς η SEQUENTIAL MERGE απαιτεί το πολύ $O((r + s)/N)$ χρόνο.

Στην χειρότερη περίπτωση $r = s = n$ και επειδή $n \geq N$, ο χρόνος εκτέλεσης του αλγορίθμου ταυτίζεται με το χρόνο εκτέλεσης του βήματος 3. Συνεπώς

$$t(2n) = O((n/N) + \log n)$$

Επειδή $p(2n) = N$, $c(2n) = p(2n) \cdot t(2n) = O(n + N \log n)$.

Ο αλγόριθμος έχει βέλτιστο κόστος αν και μόνο αν

$$N \leq n / \log n$$

.

Παράδειγμα

Έστω $N = 4$, $A = \{2, 3, \textcircled{4}, 6, 11, \textcircled{12}, 13, 15, \textcircled{16}, 20, 22, 24\}$
 $B = \{1, 5, \textcircled{7}, 8, 9, \textcircled{10}, 14, 17, \textcircled{18}, 19, 21, 23\}$, δηλαδή $r = s = 12$

Βήμα 1: $A' = \{4, 12, 16\}$, $B' = \{7, 10, 18\}$ με $\lceil r/N \rceil = \lceil s/N \rceil = 3$

Βήμα 2: Συγχώνευση των A' , B' και δημιουργία της
 $V = \{(4, 1, A), (7, 1, B), (10, 2, B), (12, 2, A), (16, 3, A), (18, 3, B)\}$

Βήμα 3: $Q(1) = (1, 1)$, $Q(2) = (5, 3)$, $Q(3) = (6, 7)$, $Q(4) = (10, 9)$

Συγχώνευση CREW SM SIMD

Βήμα 3.3: Ο επεξεργαστής P_1 αρχίζει από τα στοιχεία $a_1 = 2$ και $b_1 = 1$ και συγχωνεύει όλα τα στοιχεία των A και B μικρότερα του 7, δημιουργώντας την υποακολουθία $\{1,2,3,4,5,6\}$ της C .

Ομοίως ο επεξεργαστής P_2 αρχίζει από τα στοιχεία $a_5 = 11$ και $b_3 = 7$ και συγχωνεύει όλα τα στοιχεία μικρότερα του 12, δημιουργώντας την $\{7,8,9,10,11\}$.

Ο επεξεργαστής P_3 αρχίζει από τα στοιχεία $a_6 = 12$ και $b_7 = 14$ και δημιουργεί την $\{12,13,14,15,16,17\}$.

Τέλος, ο P_4 αρχίζει από τα $a_{10} = 20$ και $b_9 = 18$ και δημιουργεί την $\{18, 19, 20, 21, 22, 23, 24\}$. Η τελική ακολουθία C είναι συνεπώς η

$$C = \{1, 2, 3, \textcircled{4}, 5, 6, \textcircled{7}, 8, 9, \textcircled{10}, 11, \textcircled{12}, 13, 14, 15, \textcircled{16}, 17, \textcircled{18}, 19, 20, \\ 21, 22, 23, 24\}$$

Ένας καλύτερος αλγόριθμος για το EREW μοντέλο

Ο αλγόριθμος συγχωνεύει δύο ακολουθίες

$$A = \{\alpha_1, \alpha_2, \dots, \alpha_r\}, \quad B = \{b_1, b_2, \dots, b_s\}$$

σε μία ακολουθία

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

χρησιμοποιώντας N επεξεργαστές

$$P_1, P_2, \dots, P_N, \quad 1 \leq N \leq r + s$$

Ένας καλύτερος αλγόριθμος για το EREW μοντέλο

Εύρεση του μέσου δύο ταξινομημένων ακολουθιών

Δεδομένων δύο ταξινομημένων ακολουθιών $A = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ και $B = \{b_1, b_2, \dots, b_s\}$ όπου $r, s \geq 1$ συμβολίζουμε με $A \cdot B$ την ακολουθία μήκους $m = r + s$ που προκύπτει από τη συγχώνευση των A και B .

Ζητείται να βρεθεί το μέσο, δηλαδή το $\lceil m/2 \rceil$ στοιχείο της $A \cdot B$ χωρίς να σχηματιστεί (βρεθεί) η $A \cdot B$.

Ένας καλύτερος αλγόριθμος για το EREW μοντέλο

Ο αλγόριθμος επιστρέφει ένα ζεύγος (α_x, b_y) που ικανοποιεί τις ιδιότητες :

1. Ένα από τα α_x ή b_y είναι το μέσον της $A \cdot B$ δηλαδή το α_x ή το b_y είναι μεγαλύτερο από $\lceil m/2 \rceil - 1$ στοιχεία και μικρότερο από $\lfloor m/2 \rfloor$ στοιχεία.
2. Αν το α_x είναι το μέσο τότε το b_y είτε είναι

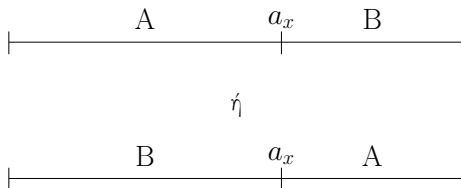
- (i) το μεγαλύτερο στοιχείο του B μικρότερο ή ίσο του α_x ή
- (ii) το μικρότερο στοιχείο του B μεγαλύτερο ή ίσο του α_x

Διαφορετικά αν το b_y είναι το μέσον, τότε α_x είναι είτε :

- (i) το μεγαλύτερο στοιχείο της A και μικρότερο ή ίσο του b_y ή
- (ii) το μικρότερο στοιχείο της A μεγαλύτερο ή ίσο του b_y .

3. Αν περισσότερα του ενός ζεύγη ικανοποιούν τα 1 και 2, τότε ο αλγόριθμος επιστρέφει το ζεύγος για το οποίο το $x + y$ είναι το μικρότερο.

Ένας καλύτερος αλγόριθμος για το EREW μοντέλο



Σχήμα: Εύρεση μέσου

Το (α_x, b_y) λέγεται μεσαίο ζεύγος της $A \cdot B$ συνεπώς x και y είναι οι δείκτες του μεσαίου ζεύγους. Ας σημειωθεί ότι α_x είναι το μέσον της $A \cdot B$ αν

(i) $\alpha_x > b_y$ και $x + y - 1 = \lceil m/2 \rceil - 1$ ή

(ii) $\alpha_x < b_y$ και $m - (x + y - 1) = \lfloor m/2 \rfloor$

Διαφορετικά το b_y είναι το μέσον της $A \cdot B$.

Ένας καλύτερος αλγόριθμος για το EREW μοντέλο

Διαδικασία 4: TWO-SEQUENCE MEDIAN (A, B, x, y)

- Βήμα 1

(1.1) χαμηλό_B $\leftarrow 1$

(1.2) χαμηλό_B $\leftarrow 1$

(1.3) ψηλό_A $\leftarrow r$

(1.4) ψηλό_B $\leftarrow s$

(1.5) $n_A \leftarrow r$

(1.6) $n_B \leftarrow s$

Ένας καλύτερος αλγόριθμος για το EREW μοντέλο

• Βήμα 2

όσο $n_A > 1$ και $n_B > 1$ κάνε

$$(2.1) u \leftarrow \text{χαμηλό}_A + \lceil (\psi\eta\lambda\acute{o}_A - \text{χαμηλό}_A - 1)/2 \rceil$$

$$(2.2) v \leftarrow \text{χαμηλό}_B + \lceil (\psi\eta\lambda\acute{o}_B - \text{χαμηλό}_B - 1)/2 \rceil$$

$$(2.3) w \leftarrow \min(\lfloor n_A/2 \rfloor, \lfloor n_B/2 \rfloor)$$

$$(2.4) n_A \leftarrow n_A - w$$

$$(2.5) n_B \leftarrow n_B - w$$

$$(2.6) \text{αν } a_u \geq b_v \text{ τότε}$$

$$(i) \psi\eta\lambda\acute{o}_A \leftarrow \psi\eta\lambda\acute{o}_A - w$$

$$(ii) \text{χαμηλό}_B \leftarrow \text{χαμηλό}_B + w$$

αλλιώς

$$(i) \text{χαμηλό}_A \leftarrow \text{χαμηλό}_A + w$$

$$(ii) \psi\eta\lambda\acute{o}_B \leftarrow \psi\eta\lambda\acute{o}_B - w$$

τέλος

τέλος

• **Βήμα 3** Επίστρεψε σαν x και y τους δείκτες του ζεύγους από

$\{a_{u-1}, a_u, a_{u+1}\} \times \{b_{v-1}, b_v, b_{v+1}\}$ που ικανοποιούν τις ιδιότητες 1-3 ενός μεσαίου ζεύγους.

Ένας καλύτερος αλγόριθμος για το EREW μοντέλο

Η TWO-SEQUENCE MEDIAN επιστρέφει τους δείκτες του ζεύγους μέσω των (α, b_y) και όχι το ίδιο το ζεύγος.

Παράδειγμα

$$A = \{10, 11, 12, 13, 14, 15, 16, 17, 18\}$$

$$B = \{3, 4, 5, 6, 7, 8, 19, 20, 21, 22\}$$

$$low_A = low_B = 1, \quad high_A = n_A = 9, \quad high_B = n_B = 10$$

Ένας καλύτερος αλγόριθμος για το EREW μοντέλο

Βήμα 2

1η επανάληψη

$$u = v = 5, \quad w = \min(4, 5) = 4, \quad n_A = 5, \quad n_B = 6$$

Επειδή $\alpha_5 > b_5$, $high_A = low_B = 5$

2η επανάληψη

$$u = 3, v = 7, w = \min(2, 3) = 2, n_A = 3, n_B = 4$$

Επειδή $\alpha_3 < b_7$, $low_A = 3$ και $high_B = 8$

Ένας καλύτερος αλγόριθμος για το EREW μοντέλο

3η επανάληψη

$$u = 4, v = 6, w = \min(1, 2) = 1 \quad n_A = 2, n_B = 3$$

Επειδή $\alpha_4 > b_6$, $high_A = 4$ $low_B = 6$

4η επανάληψη

$$u = 3, v = 7, w = \min(1, 1) = 1 \quad n_A = 1, n_B = 2$$

Επειδή $\alpha_3 < b_7$, $low_A = 4$ και $high_B = 7$

Ένας καλύτερος αλγόριθμος για το EREW μοντέλο

Βήμα 3

Έχουμε 9 ζεύγη $\{11, 12, 13\} \times \{8, 19, 20\}$

Αυτά που ικανοποιούν τις δύο πρώτες ιδιότητες είναι τα εξής δύο:

$$(\alpha_4, b_6) = (13, 8) \quad \text{και} \quad (\alpha_4, b_7) = (13, 19)$$

οπότε επιστρέφεται το $(4, 6)$.

Συγχώνευση στο μοντέλο EREW

Δεδομένων δύο ταξινομημένων ακολουθιών $A = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ και $B = \{b_1, b_2, \dots, b_s\}$ υποθέτουμε την ύπαρξη N επεξεργαστών $P_1, P_2, \dots, P_N$, όπου N είναι μια δύναμη του 2 και $1 \leq N \leq r + s$. Ο αλγόριθμος που θα αναπτυχθεί στη συνέχεια συγχωνεύει τις A και B σε μία ταξινομημένη ακολουθία $C = \{c_1, c_2, \dots, c_{r+s}\}$ σε δυο φάσεις ως εξής:

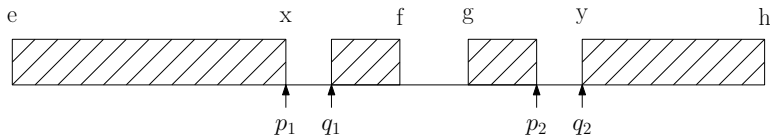
Φάση 1: Κάθε μια από τις δύο ακολουθίες A και B χωρίζονται σε N (πιθανές κενές) υποακολουθίες $A_1, A_2, \dots, A_N$ και $B_1, B_2, \dots, B_N$ τέτοιες ώστε

- (i) $|A_i| + |B_i| = (r + s)/N$ για $1 \leq i \leq N$
- (ii) όλα τα στοιχεία στην $A_i \cdot B_i$ είναι μικρότερα ή ίσα από όλα τα στοιχεία στην $A_{i+1} \cdot B_{i+1}$ για $1 \leq i \leq N$

Συγχώνευση στο μοντέλο EREW

(I) Περίπτωση όπου

- ▶ α_x είναι το μεσαίο στοιχείο
- ▶ $\alpha_x \leq b_y$



Σχήμα: $\alpha_x \leq b_y$

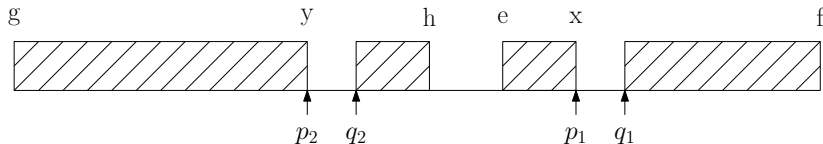
Παρατήρηση

- Οι υποακολουθίες με τα μικρότερα στοιχεία στέλνονται σε ένα επεξεργαστή P_{2i-1} , ενώ οι άλλες με τα μεγαλύτερα στοιχεία στέλνονται στον επεξεργαστή P_{2i}
- Το b_y συμπεριλαμβάνεται στην υπακολουθία με τα μεγαλύτερα στοιχεία

Συγχώνευση στο μοντέλο EREW

(II) Περίπτωση όπου

- ▶ α_x είναι το μεσαίο στοιχείο
- ▶ $b_y \leq \alpha_x$



Σχήμα: $b_y \leq \alpha_x$

Συγχώνευση στο μοντέλο EREW

Διαδικασία 5: EREW MERGE (A, B, C)

• Βήμα 1

(1.1) Ο επεξεργαστής P_1 λαμβάνει την τετράδα $(1, r, 1, s)$

(1.2) για $j = 1$ έως $\log N$ κάνε

για $i = 1$ έως 2^{j-1} κάνε παράλληλα

Ο επεξεργαστής P_i έχοντας λάβει την τετράδα (e, f, g, h)

(1.2.1) {Βρίσκει το μεσαίο ζεύγος των δύο ακολουθιών

TWO-SEQUENCE MEDIAN ($A[e, f], B[g, h], x, y$)}

(1.2.2) {Υπολογίζει τέσσερις δείκτες p_1, p_2, q_1 και q_2 ως εξής}

αν α_x είναι το μεσαίο τότε

(i) $p_1 \leftarrow x$

(ii) $q_1 \leftarrow x + 1$

(iii) αν $b_y \leq \alpha_x$ τότε

(a) $p_2 \leftarrow y$

(b) $q_2 \leftarrow y + 1$

αλλιώς

(a) $p_2 \leftarrow y - 1$

(b) $q_2 \leftarrow y$

τέλος

αλλιώς

τέλος

(i) $p_2 \leftarrow y$

(ii) $q_2 \leftarrow y + 1$

(iii) αν $\alpha_x \leq b_y$ τότε

(a) $p_1 \leftarrow x$

(b) $q_1 \leftarrow x + 1$

αλλιώς

(a) $p_1 \leftarrow x - 1$

(b) $q_1 \leftarrow x$

τέλος

(1.2.3) Μεταδίδει τις τετράδες (e, p_1, g, p_2) στον P_{2i-1}

(1.2.4) Μεταδίδει τις τετράδες (q_1, f, q_2, h) στον P_{2i}

τέλος

τέλος

Συγχώνευση στο μοντέλο EREW

- Βήμα 2

για $i = 1$ έως N κάνε παράλληλα

 Ο επεξεργαστής P_i έχοντας λάβει την τετράδα (a, b, c, d)

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

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

 (2.3) SEQUENTIAL MERGE $(A[a, b], B[c, d], C[w, z])$

τέλος

- Όλα τα ζεύγη A_i και B_i , $1 \leq i \leq N$ συγχωνεύονται ταυτόχρονα και τοποθετούνται στην C .

Συγχώνευση στο μοντέλο EREW

Παράδειγμα

$$A = \{10, 11, 12, 13, 14, 15, 16, 17, 18\},$$

$$B = \{3, 4, 5, 6, 7, 8, 19, 20, 21, 22\}, N = 4$$

$$r = 9, s = 10$$

Βήμα 1.1: Ο P_1 λαμβάνει (1,9,1,10).

Βήμα 1.2: Ο P_1 προσδιορίζει το μεσαίο ζεύγος των A και B που είναι το (4, 6). Διατηρεί το (1, 4, 1, 6) και μεταδίδει το (5, 9, 7, 10) στον P_2 . Στη δεύτερη επανάληψη, ο P_1 υπολογίζει τους δείκτες του μεσαίου ζεύγους των $A[1, 4] = \{10, 11, 12, 13\}$ και $B[1, 6] = \{3, 4, 5, 6, 7, 8\}$ που είναι το (1,5). Ταυτόχρονα ο P_2 κάνει το ίδιο με τις $A[5, 9] = \{14, 15, 16, 17, 18\}$ και $B[7, 10] = \{19, 20, 21, 22\}$ και λαμβάνει το ζεύγος (9, 7). Ο P_1 κρατά το (1, 0, 1, 5) και μεταδίδει το (1, 4, 6, 6) στον P_2 . Όμοια ο P_2 μεταδίδει το (5, 9, 7, 6) στον P_3 και το (10, 9, 7, 10) στον P_4 .

Συγχώνευση στο μοντέλο EREW

Βήμα 2: Οι P_1, P_2, P_3 και P_4 δημιουργούν ταυτόχρονα την $C[1, 19]$ ως εξής:

- Η τελευταία τετράδα που έλαβε ο P_1 είναι η $(1, 0, 1, 5)$ και ο P_1 υπολογίζει $w = 1, z = 5$ και αντιγράφει την $B[1, 5] = \{3, 4, 5, 6, 7\}$ στην $C[1, 5]$.
- Όμοια ο P_2 , έχοντας λάβει τελευταία την $(1, 4, 6, 6)$, υπολογίζει $w = 6$ και $z = 10$ και συγχωνεύει τις $A[1, 4]$ και $B[6, 6]$ για το σχηματισμό της $C[6, 10] = \{8, 10, 11, 12, 13\}$.
- Ο P_3 έχοντας λάβει την τελευταία $(5, 9, 7, 6)$ υπολογίζει $w = 11$ και $z = 15$ και αντιγράφει την $A[5, 9] = \{14, 15, 16, 17, 18\}$ στην $C[11, 15]$.
Τέλος ο P_4 έχοντας λάβει στο τέλος την $(10, 9, 7, 10)$ υπολογίζει $w = 16, z = 19$ και αντιγράφει την $B[7, 10] = \{19, 20, 21, 22\}$ στην $C[16, 19]$.

Συγχώνευση στο μοντέλο EREW

Ανάλυση

Βήμα 1.1: Ο P_1 διαβάζει από τη μνήμη σε σταθερό χρόνο.

Βήμα 1.2: Κατά τη διάρκεια της j -ιστής επανάληψης κάθε επεξεργαστής υπολογίζει το μεσαίο ζεύγος από $(r + s)/2^{j-1}$ στοιχεία, μέσω της TWO-SEQUENCE MEDIAN, σε $O(\log[(r + s)/2^{j-1}])$ χρόνο, ή $O(\log(r + s))$. Τα άλλα δύο βήματα στο βήμα 1.2 απαιτούν σταθερό χρόνο.

Επειδή εκτελούνται $\log N$ επαναλήψεις του βήματος 1.2 το Βήμα 1 απαιτεί $O(\log N \cdot \log(r + s))$ χρόνο.

Στο βήμα 2 κάθε επεξεργαστής συγχωνεύει το πολύ $(r + s)/N$ στοιχεία. Η εργασία αυτή γίνεται με την SEQUENTIAL MERGE και απαιτεί $O((r + s)/N)$ χρόνο.

Συγχώνευση στο μοντέλο EREW

Συνολικά

$$O((r + s)/N) + \log N \cdot \log(r + s)$$

Στη χειρότερη περίπτωση $r = s = n$ και

$$t(2n) = O(n/N + \log^2 n)$$

με κόστος

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

Επειδή $\Omega(n)$ είναι το κάτω φράγμα του αριθμού των πράξεων για συγχώνευση το ανωτέρω κόστος είναι βέλτιστο όταν $N \leq n/\log^2 n$.

Σημείωση: Η $\log^2 n$ είναι αύξουσα συνάρτηση σχεδόν flat δηλ. έχουμε βέλτιστο κόστος για μεγάλο πλήθος επεξεργατών. Ο αλγόριθμος αυτός αποτελεί σπάνια περίπτωση βέλτιστου κόστους με μεγάλο πλήθος επεξεργαστών.