ΕΠΙΛΟΓΗ (Selection)

Το πρόβλημα της επιλογής

Δίνεται μια ακολουθία S από n στοιχεία και ένας ακέραιος k, όπου $1 \leq k \leq n$. Να προσδιοριστεί το k μικρότερο στοιχείο στην S.

Δεν έχουμε ταξινομημένους αριθμούς

Υπόθεση: Η ανάπτυξη του παράλληλου αλγόριθμου θα γίνει για το SM SIMD μοντέλο.

Ορισμός: Τα στοιχεία ενός συνόλου A ικανοποιούν μια γραμμική διάταξη $<$ τότε και μόνον τότε αν

(i) για $a, b \in A$, $a < b$, $a = b$ ή $b < a$

και (ii) για $a, b, c \in A$, αν $a < b$ και $b < c$, τότε $a < c$.

Παράδειγμα: Το σύνολο των ακεραίων αριθμών ικανοποιεί μια γραμμική διάταξη.

Βαθμός Για μια ακολουθία $S = \{s_1, s_2, \ldots, s_n\}$ της οποίας τα στοιχεία είναι και στοιχεία ένος συνόλου με γραμμική διάταξη, ο

βαθμός ενός στοιχείου s_i του S είναι το πλήθος των στοιχείων του S που προηγούνται του s_i συν 1.

Παράδειγμα: $S=\{9,-4,3,-6,7,0\}$ ο βαθμός του 0 είναι 3.

Εξήγηση: Μικρότερα στοιχεία του 0 είναι τα -4,-6, δηλ. πλήθος=2->βαθμός του 0=3

Αν $s_i = s_j$ τότε το s_i προηγείται του s_j τότε και μόνον τότε αν $i<j$.

Επιλογή

Δίνεται μία ακολουθία $S=\{s_1,s_2,\dots,s_n\}$, της οποίας τα στοιχεία έχουν παρθεί από ένα γραμμικά διατεταγμένο σύνολο, και ένας ακέραιος k, $1\le k\le n$. Ζητείται ο προσδιορισμός του στοιχείου της S με βαθμό k. Το στοιχείο με βαθμό k θα συμβολίζεται με $s_{(k)}$.

Πολυπλοκότητα (κάτω όριο)

Αν τα στοιχεία του S ήταν ταξινομημένα δηλ.

$$S=\{s_{(1)},s_{(2)},\dots,s_{(n)}\}$$

τότε το $s_{(k)}$ λαμβάνεται σε ένα βήμα. Προφανώς δεν υποθέτουμε ότι έχουμε αυτή την περίπτωση ούτε επιθυμούμε να ταξινομήσουμε τα στοιχεία του S.

Λύση

Αν k=1 ή k=n, τότε εξετάζοντας όλα τα στοιχεία της S διατηρώντας κάθε φορά το μικρότερο (k=1) (ή το μεγαλύτερο (k=n)) βρίσκεται το ζητούμενο στοιχείο. Ωστόσο αυτός ο αλγόριθμος δεν μπορεί να εφαρμοστεί αν $1<k<n$. Στην καλύτερη περίπτωση λοιπόν η πολυπλοκότητα του αλγορίθμου είναι

$\Omega(n)$ (κάτω όριο)

Ένας ακολουθιακός αλγόριθμος

->στην καλύτερη περίπτωση έχει πολυπλοκότητα=1

Χαρακτηριστικά

- αναδρομικός αλγόριθμος
- διαίρει και βασίλευε (divide and conquer)
- απορρίπτει ένα πλήθος στοιχείων σε κάθε φάση

Procedure SEQUENTIAL SELECT (S, k)

Βήμα 1: **αν** $|S| \leq Q$ **τότε** ταξινόμησε την S και επέστρεψε το $k^{\sigma\tau o}$ στοιχείο
αλλιώς υποδιαίρεσε την S σε $|S|/Q$ υποακολουθίες με Q στοιχεία η κάθε μία (μέχρι Q-1 εναπομείναντα στοιχεία)
Τέλος αν.

Βήμα 2: Ταξινόμησε κάθε υποακολουθία και υπολόγισε το μέσο της.

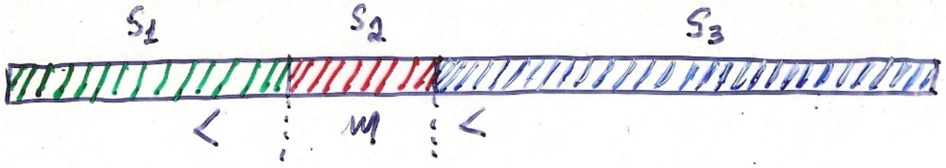
Βήμα 3: Κάλεσε την SEQUENTIAL SELECT αναδρομικά για τον υπολογισμό του m, το μέσο των $|S|/Q$ μέσων που βρέθηκαν στο βήμα 2.

Βήμα 4: Δημιούργησε τρείς υποακολουθίες S_1, S_2, και S_3 με στοιχεία της S μικρότερα, ίσα, και μεγαλύτερα από το m, αντίστοιχα.

Βήμα 5. **αν** $|S_1| \geq k$ **τότε** {το $k^{\sigma\tau\acute{o}}$ στοιχείο της S πρέπει να είναι στο S_1}
κάλεσε την SEQUENTIAL SELECT αναδρομικά για να βρείς το $k^{\sigma\tau\acute{o}}$ στοιχείο του S_1
αλλιώς αν $|S_1|+|S_2| \geq k$ **τότε** επέστρεψε m {διότι το k στοιχείο ανήκει στο S_2}
αλλιώς κάλεσε την SEQUENTIAL SELECT αναδρομικά για να βρείς το $(k-|S_1|-|S_2|)^{\sigma\tau\acute{o}}$ στοιχείο του S_3 {διότι το k στοιχείο ανήκει στο S_3}.
τέλος αν.
τέλος αν.

• divide and conquer

Βήμα 5



Εξήγηση Βήματος 5: Αφού το m είναι μέσο των |S|/Q στοιχείων, υπάρχουν |S|/2Q στοιχεία >=m
Κάθε ένα από τα |S|/Q στοιχεία ήταν το ίδιο μέσο ενός συνόλου Q στοιχείων, δηλ. υπάρχουν Q/2 στοιχεία >= από αυτό.
Άρα υπάρχουν (|S|/2Q)x(Q/2)=|S|/4 στοιχεία του S τα οποία είναι σίγουρα >=m

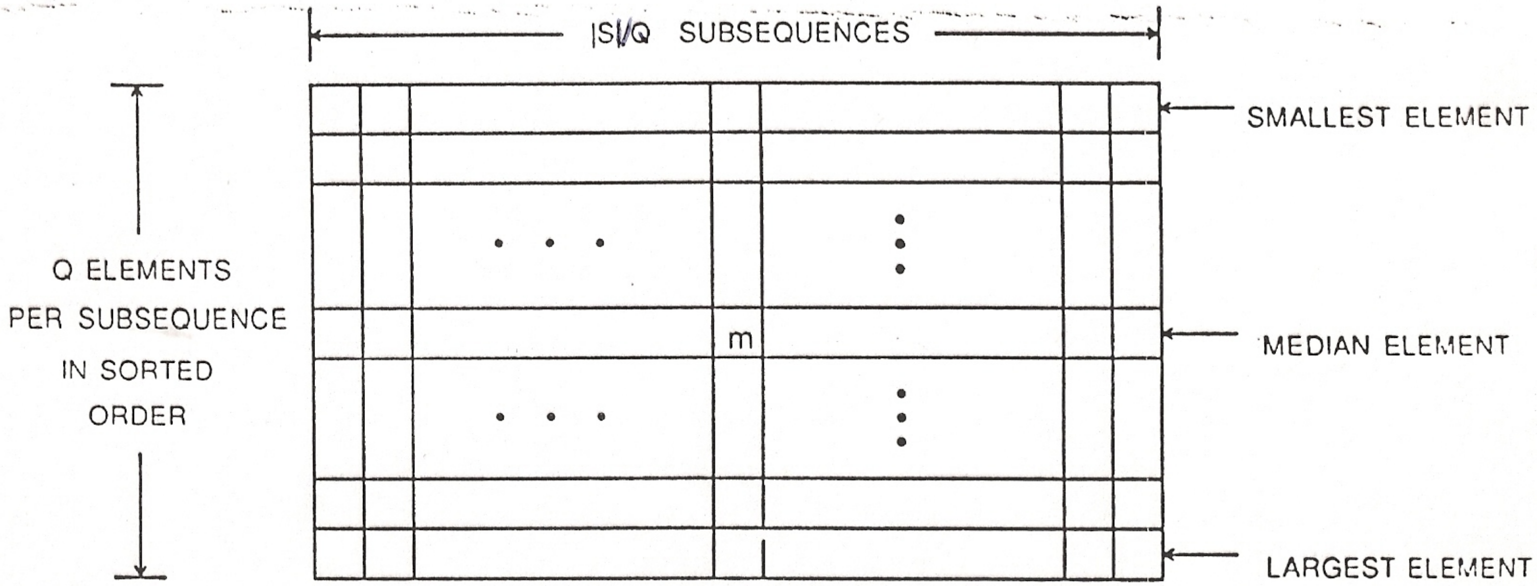


Figure 2.1 Main idea behind procedure SEQUENTIAL SELECT.

Ανάλυση Πολυπλοκότητας

Βήμα 1: Ταξινόμηση της S όταν $|S| < Q$ απαιτεί σταθερό χρόνο $O(1)$. Διαφορετικά, υποδιαίρεση της S απαιτεί $c_1 n$ χρόνο

Βήμα 2: Η ταξινόμηση κάθε $|S|/Q$ υποακολουθίας (έχει Q στοιχεία) απαιτεί σταθερό χρόνο. Για όλες $c_2 n$ χρόνο

Βήμα 3: $t(n/Q)$, $t(n)$: χρόνος εκτέλεσης της SEQUENTIAL SELECT

Βήμα 4: $c_3 n$

Βήμα 5: $(|S|/2Q) \times (Q/2) = |S|/4$ στοιχεία της S (βλέπε σχήμα) είναι μεγαλύτερα ή ίσα με m. Συνεπώς $|S_1| \leq 3|S|/4$. Όμοια $|S_3| \leq 3|S|/4$. Συνεπώς μια αναδρομική κλήση της SEQUENTIAL SELECT απαιτεί $\underline{t(3n/4)}$ χρόνο.

Συνολικά: $t(n) = c_4 n + t(n/Q) + t(3n/4)$

όπου $c_4 = c_1 + c_2 + c_3$

Καθορισμός του Q

Αν το Q επιλεγεί τέτοιο ώστε

$$n/Q + 3n/4 < n$$

τότε οι δυο αναδρομικές κλήσεις στην procedure εφαρμόζονται σε φθίνουσες ακολουθίες. Οποιαδήποτε τιμή του $Q \geq 5$ ικανοποιεί την ανωτέρω ανισότητα. Για $Q = 5$ έχουμε

$$t(n) = c_4 n + t(n/5) + t(3n/4)$$

Υποθέτοντας

$$t(n) \leq c_5 n$$

έχουμε

$$t(n) \leq c_4 n + c_5 (19n/20)$$

και για $c_5 = 20 c_4$

$$t(n) \leq c_5 (n/20) + c_5 (19n/20)$$

$$= c_5 n$$

$t(n) = O(n)$ βέλτιστος χρόνος!

ΔΥΟ ΧΡΗΣΙΜΟΙ ΑΛΓΟΡΙΘΜΟΙ

Στο EREW SM SIMD μοντέλο δεν μπορούν δύο επεξεργαστές να προσπελαύνουν την ίδια τοποθεσία μνήμης ταυτόχρονα.

Μετάδοση (broadcasting)

(i) Όλοι οι επεξεργαστές χρειάζονται να διαβάσουν ένα δεδομένο από μία συγκεκριμένη τοποθεσία της κοινής μνήμης.

Επικοινωνία (όλοι με ένα)

(ii) Κάθε επεξεργαστής πρέπει να κάνει υπολογισμούς στα δεδομένα, τα οποία υπάρχουν στους άλλους επεξεργαστές και συνεπώς χρειάζεται να λάβει αυτά τα δεδομένα.

Οι δύο αυτές λειτουργίες δεν μπορούν να εκτελεστούν σε ένα βήμα στο EREW μοντέλο. και πρέπει να προσομοιωθούν.

Υπόθεση: N επεξεργαστές $P_1, P_2, \ldots, P_N$ είναι διαθέσιμοι σε ένα EREW SM SIMD υπολογιστή.

D: Είναι η τοποθεσία της μνήμης στην οποία υπάρχει το ζητούμενο δεδομένο. Αυτή τη μνήμη θα προσπελάσουν όλοι οι επεξεργαστές προκειμένου να διαβάσουν το δεδομένο
A: πίνακας αποθηκευμένος στη μνήμη μήκους N. Είναι αρχικά άδειος και χρησιμοποιείται από την συνάρτηση σαν "χώρος εργασίας" για τον διαμοιρασμό των στοιχείων της D στους επεξεργαστές. Η i-θέση του πίνακα αυτού συμβολίζεται με A(i).

O(logN)

Procedure BROADCAST (D, N, A)

Βήμα 1: Ο επεξεργαστής P_1

(i) διαβάζει την τιμή από την D,

(ii) την αποθηκεύει στην μνήμη του, και

(iii) τη γράφει στην A(1).

Βήμα 2: για i=0 **μέχρι** (log N-1) **κάνε**

για $j=2^i+1$ **μέχρι** 2^{i+1} **κάνε παράλληλα**

Ο επεξεργαστής P_j

(i) διαβάζει την τιμή από την $A(j-2^i)$,

(ii) την αποθηκεύει στην μνήμη του, και

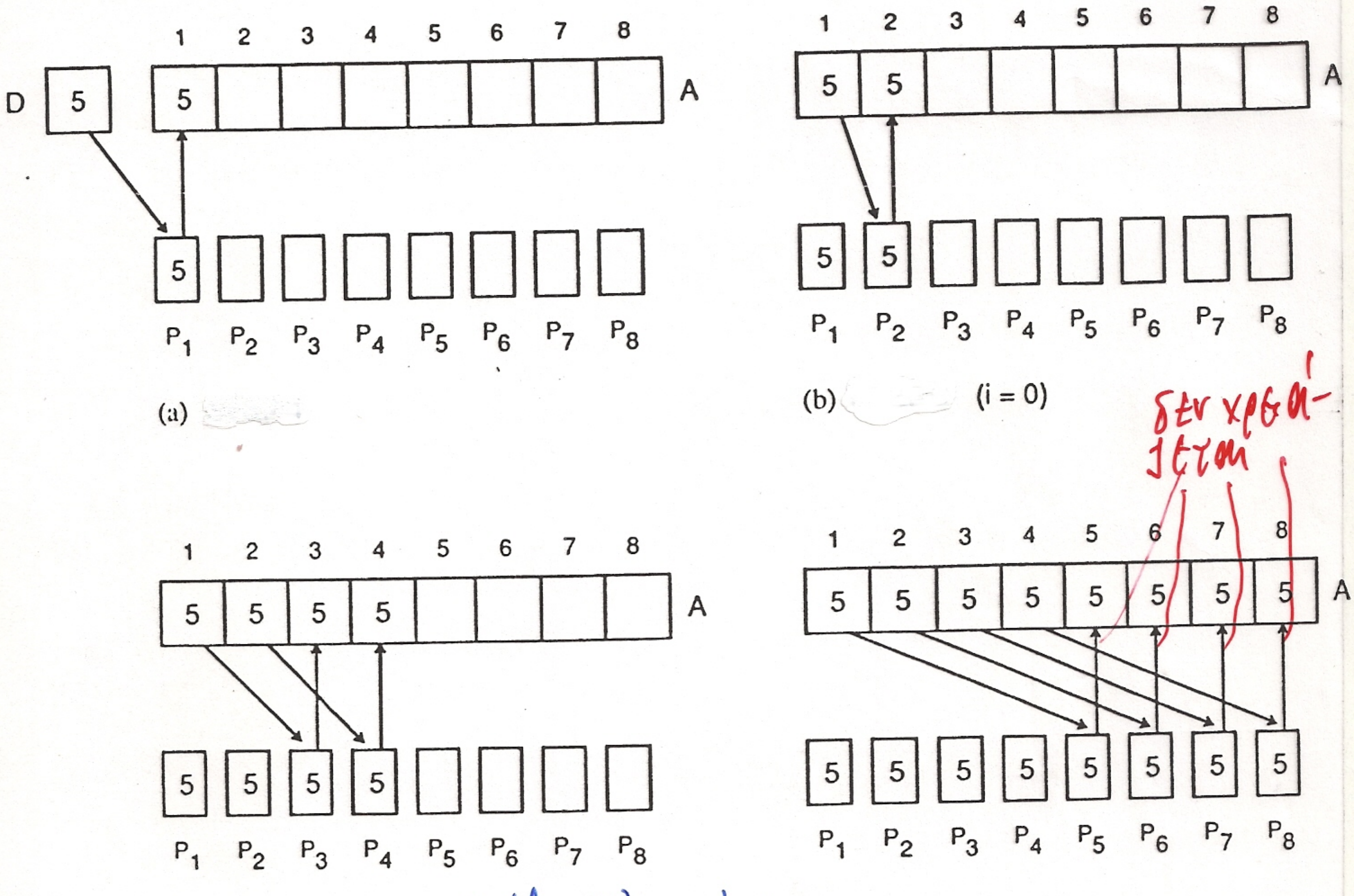
(iii) τη γράφει στην A(j).

τέλος για

τέλος για.

Τρόπος λειτουργίας της BROADCAST για N=8, D=5:

Όταν η διαδικασία σταματάει τότε όλοι οι επεξεργαστές έχουν αποθηκευμένο το D στην τοπική τους μνήμη για μετέπειτα χρήση. Αφού ο αριθμός των επεξεργαστών που έχουν διαβάσει τη D διπλασιάζεται σε κάθε επανάληψη, η διαδικασία τερματίζει σε χρόνο O(logN). Οι απαιτήσεις της μνήμης για την BROADCAST είναι ένας πίνακας μήκους N.



(a)

(b) (i = 0)

(c) (i = 1)

O(logN) χρόνο
N/2 μνήμη

(d) (i = 2)

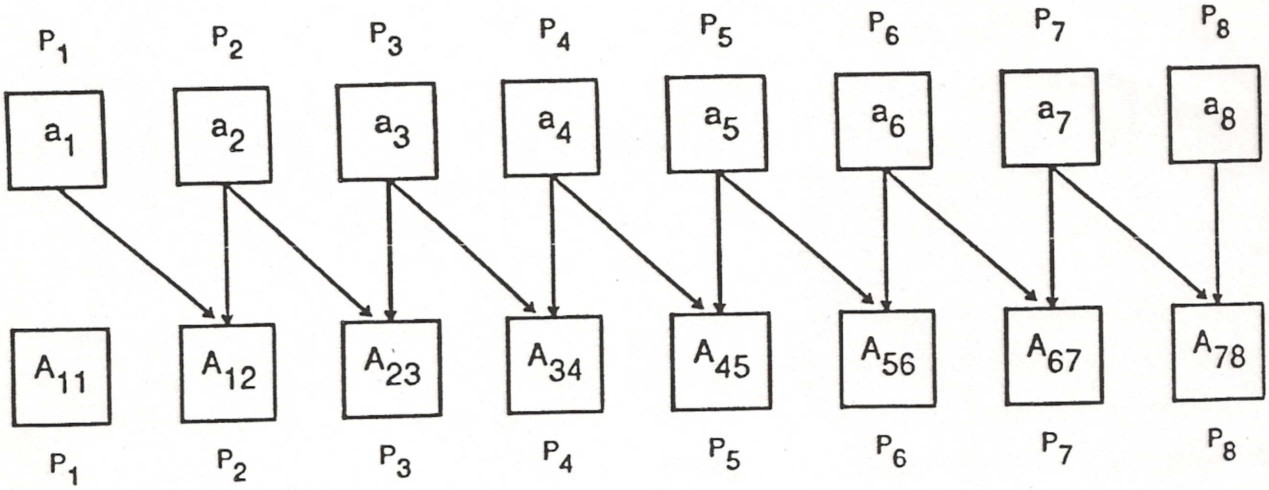
Υπολογισμός των αθροισμάτων
Υπόθεση: P_i έχει το a_i στην τοπική μνήμη.

$$P_i := a_1 + a_2 + \dots + a_i$$

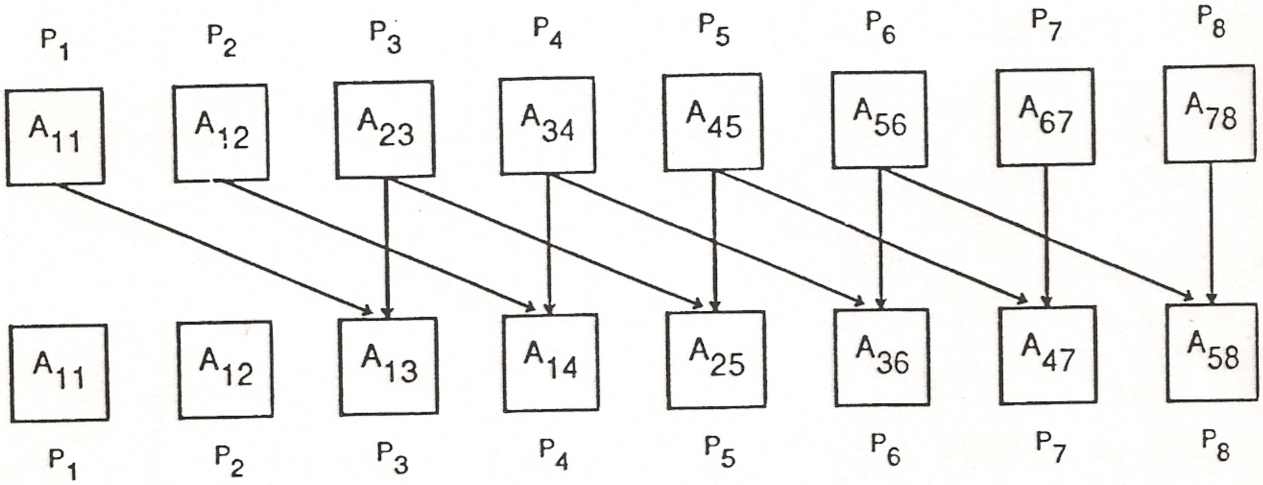
procedure ALLSUMS $(a_1, a_2, \ldots, a_N)$

for $j = 0$ **to** $\log N - 1$ **do**
  for $i = 2^j + 1$ **to** N **do in parallel**
    Processor P_i
      (i) obtains a_{i-2^j} from P_{i-2^j} through shared memory and
      (ii) replaces a_i with $a_{i-2^j} + a_i$.
  end for
end for. □

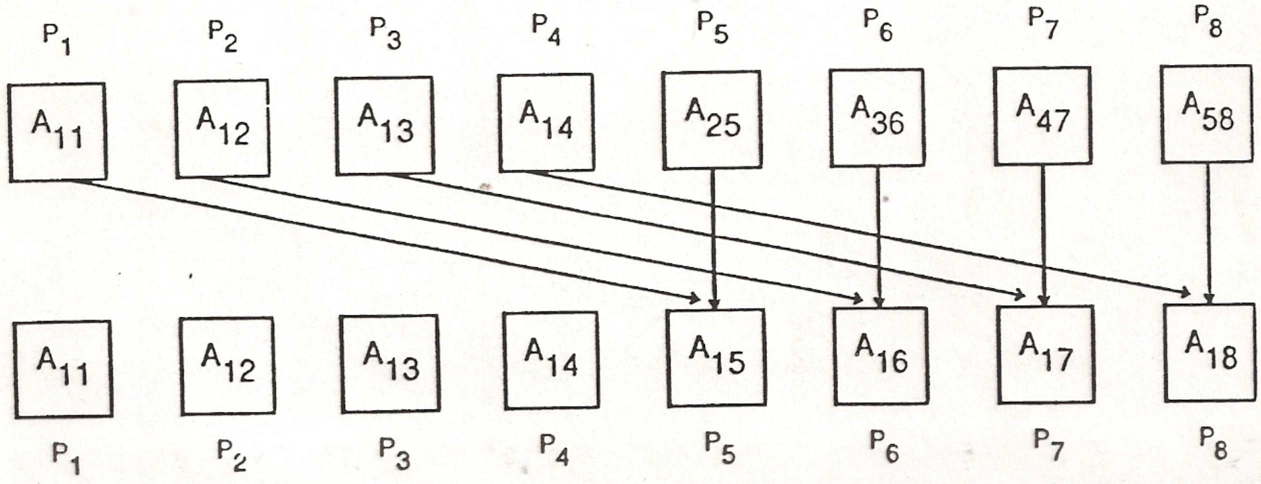
$A_{ij} = a_i + a_{i+1} + \cdots + a_j$ $\quad O(\log N)$



(a) j = 0



(b) j = 1



(c) j = 2

όταν έχει ολοκληρωθεί η διαδικασία, το a_i έχει αντικατασταθεί από το P_i για 1<=i<=N. Η διαδικασία απαιτεί O(logN) χρόνο αφού ο αριθμός των επεξεργαστών έχει τελειώσει τους υπολογισμούς σε κάθε βήμα.

ΕΝΑΣ ΑΛΓΟΡΙΘΜΟΣ ΓΙΑ ΤΗΝ ΠΑΡΑΛΛΗΛΗ ΕΠΙΛΟΓΗ (EREW SM SIMD)

Υποθέσεις:

1. $S = \{s_1, s_2, \dots, s_n\}$, k, $1 \le k \le n$
2. $P_1, P_2, \dots, P_N$ επεξεργαστές
3. Κάθε επεξεργαστής έχει λάβει το n και έχει υπολογίσει το x από την $N = n^{1-x}$, όπου $0 < x < 1$.
4. Κάθε ένας από τους n^{1-x} επεξεργαστές μπορεί να αποθηκεύσει n^x στοιχεία στην τοπική μνήμη του
5. Κάθε επεξεργαστής μπορεί να εκτελέσει τις SEQUENTIAL SELECT, BROADCAST και ALLSUMS.
6. M είναι ένας μονοδιάστατος πίνακας στην κοινή μνήμη μήκους N.

Procedure PARALLEL SELECT (S, k)

Βήμα 1: **αν** $|S| \leq 4$ **τότε** ο P_1 χρησιμοποιεί το πολύ πέντε συγκρίσεις για να επιστρέψει το $k^{στο}$ στοιχείο
αλλιώς
(i) Η ακολουθία S υποδιαιρείται σε $|S|^{1-x}$ υποακολουθίες S_i μήκους $|S|^x$ η κάθε μία, όπου $1 \leq i \leq |S|^{1-x}$
(ii) Η υποακολουθία S_i ανατίθεται στον επεξεργαστή P_i.
τέλος αν.

Βήμα 2: **για** i=1 **μέχρι** $|S|^{1-x}$ **κάνε παράλληλα**
(2.1) {Ο P_i αποκτά το μέσο m_i, δηλ., το $\lceil |S_i|/2 \rceil$ ιοστό στοιχείο, της υποακολουθίας του }
SEQUENTIAL SELECT (S_i, $\lceil |S_i|/2 \rceil$)
(2.2) Ο P_i αποθηκεύει το m_i στο M(i)
τέλος για.

Βήμα 3: {Το υποπρόγραμμα καλείται αναδρομικά για να αποκτηθεί το μέσο m του M}
PARALLEL SELECT (M, $\lceil |M|/2 \rceil$).

Βήμα 4: Η ακολουθία S υποδιαιρείται σε τρία υποακολουθίες:
$L=\{s_i \in S: s_i < m\}$,
$E=\{s_i \in S: s_i = m\}$, και
$G=\{s_i \in S: s_i > m\}$.

Βήμα 5: **αν** $|L| \geq k$ **τότε** PARALLEL SELECT (L, k)
αλλιώς αν $|L|+|E| \geq k$ **τότε** επέστρεψε m
αλλιώς PARALLAL SELECT (G, k-|L|-|E|)
τέλος αν
τέλος αν.

Ανάλυση

Βήμα 1: Για την εκτέλεση του βήματος αυτού αποστέλλεται η διεύθυνση αρχής A της ακολουθίας S στην κοινή μνήμη, το μέγεθος $|S|$ και η τιμή k. Οι τιμές αυτές μεταδίδονται σε όλους τους επεξεργαστές με την BROADCAST σε $O(\log n^{1-x})$ χρόνο. Αν $|S| \leq k$, τότε ο P_1 επιστρέφει το k-οστό στοιχείο σε σταθερό χρόνο. Διαφορετικά ο P_i υπολογίζει τη διεύθυνση του πρώτου και του τελευταίου στοιχείου της S_i από τους

$$A + (i-1)n^x \quad \text{και} \quad A + in^x - 1$$

σε σταθερό χρόνο. Συνεπώς το βήμα 1 απαιτεί $c_1 \log n$ μονάδες χρόνου για κάποια σταθερά c_1.

Βήμα 2: Η SEQUENTIAL SELECT βρίσκει το μέσο μιας ακολουθίας μήκους n^x σε $c_2 n^x$ μονάδες χρόνου για κάποια σταθερά c_2.

Βήμα 3: Καλείται η PARALLEL SELECT για μία ακολουθία μήκους n^{1-x}. Συνεπώς το παρόν βήμα απαιτεί $t(n^{1-x})$ χρόνο.

Βήμα 4: Η ακολουθία S χωρίζεται στις υποακολουθίες L, E και G ως εξής:

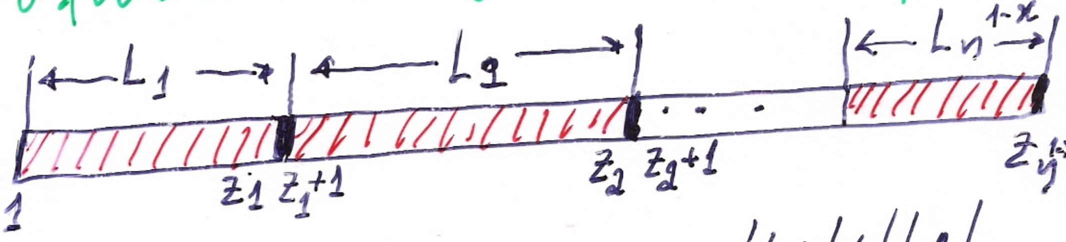
(i) Το m μεταδίδεται σε όλους τους επεξεργαστές σε $O(\log n^{1-x})$ χρόνο με τη χρήση της BROADCAST.

(ii) Κάθε επεξεργαστής P_i χωρίζει την S_i σε τρεις υποακολουθίες L_i, E_i και G_i με στοιχεία μικρότερα, ίσα και μεγαλύτερα από το m, αντίστοιχα. Αυτή η εργασία μπορεί να γίνει σε χρόνο γραμμικό ως προς το μέγεθος της S_i, δηλαδή σε $O(n^x)$ χρόνο.

(iii) Οι υποακολουθίες L_i, E_i και G_i συγχωνεύονται για να σχηματίσουν τις L, E και G. Στη συνέχεια δείχνεται πως αυτή η εργασία μπορεί να γίνει για της L_i. Όμοια και με τον ίδιο χρόνο μπορεί να γίνει για τη συγχώνευση των E_i και G_i, αντίστοιχα. Θέτοντας $a_i = |L_i|$, υπολογίζεται το άθροισμα

$$z_i = \sum_{j=1}^{i} a_j , \quad 1 \le i \le n^{1-x}$$

Τα αθροίσματα αυτά υπολογίζονται σε $O(\log n^{1-x})$ χρόνο από τους n^{1-x} επεξεργαστές. χρησιμοποιώντας την ALLSUMS. Όλοι οι επεξεργαστές συγχωνεύουν της L_i υποακολουθίες τους για τον σχηματισμό της L.



$z_1 = a_1 = |L_1|$, $z_2 = a_1 + a_2 = |L_1| + |L_2|$

Σχηματισμός της L

Ο επεξεργαστής p_i αναγράφει την L_i στην L $(z_0=0)$ ξεκινώντας από τη θέση $z_{i-1}+1$. Η εργασία αυτή μπορεί να γίνει σε $O(n^x)$ χρόνο. Συνεπώς ο χρόνος που απαιτείται για το παρόν βήμα είναι $c_3 n^x$ για κάποια σταθερά c_3.

Βήμα 5 Το μέγεθος της L που χρειάζεται στο βήμα αυτό έχει υπολογιστεί στο βήμα 4 και είναι z_n^{1-x}. Η ίδια παρατήρηση ισχύει για τα μεγέθη των E και G.
Στη συνέχεια πρέπει να υπολογιστεί ο χρόνος που απαιτείται για κάθε μία από τις δύο αναδρομικές κλήσεις της PARALLEL SELECT.
Επειδή το m είναι το μέσο του M, έπεται ότι $n^{1-x}/2$ στοιχεία είναι μεγαλύτερα από αυτό. Επίσης κάθε στοιχείο του M είναι μικρότερο από $n^x/2$ τουλάχιστον στοιχεία της S.
Άρα $\frac{n^{1-x}}{2} \cdot \frac{n^x}{2} = \frac{n}{4}$ της L είναι τουλάχιστον μεγαλύτερα ή ίσα του m. Συνεπώς $|L| \le \frac{3n}{4}$.
Όμοια, $|G| \le \frac{3n}{4}$. Τελικά, το παρόν βήμα απαιτεί το πολύ $t(3n/4)$ χρόνο.

Η προηγούμενη ανάλυση έχει σαν αποτέλεσμα την ακόλουθη αναδρομική εξίσωση για το συνολικό χρόνο

$$t(n) = c_1 \log n + c_2 n^x + t(n^{1-x}) + c_3 n^x + t(3n/4)$$

της οποίας η λύση είναι $t(n) = O(n^x)$ για $n > 4$. Επίσης $c(n) = p(n) \cdot t(n) = n^{1-x} \cdot O(n^x) = O(n)$.
Το κόστος αυτό είναι βέλτιστο λόγω $\Omega(n)$.

Παράδειγμα

$S = \{3, 14, 16, 20, 8, 31, 22, 12, 33, 1, 4, 9, 10, 5, 13, 7, 24, 2, 14, 26, 18, 34, 36, 25, 14, 27, 32, 35, 33\}$

$n = 29$, $k = 21$, $N = 5$. Συνεπώς

$|S|^{1-x} = 5$ ή $29^{1-x} = 5$ η $1-x = 0.47796$

Σχήμα 2.4(α)

$|L| = 11$, $|E| = 3$, $|L| + |E| < k$ αναδρομική

κλήση με $S = G$ και $k = 21 - (11+3) = 7$.

Επειδή $|G| = 15$ χρησιμοποιείται

$15^{1-x} = 3.6485$ δηλ. 3 επεξεργαστές

Σχήμα 2.4(β)

Επειδή $|L| = 6$, $|E| = 1$, $|L| + |E| = 7 = k$

άρα $m = 26$

- $p(n) < n^{1/2}$, προσαρμοστικός
- $t(n)$ μικρό, προσαρμοστικό
- $c(n) = p(n) \times t(n)$ βέλτιστο κόστος όταν

  $c(n) = \Omega(n)$ του ακολουθιακού αλγορίθμου

S

3	14	16	20	8	31	22	12	33	1	4	9	10	5	13	7	24	2	14	26	18	34	36	25	14	27	32	35	33

(a)

P_1

3	14	16	20	8	31

P_2

22	12	33	1	4	9

P_3

10	5	13	7	24	2

P_4

14	26	18	34	36	25

P_5

14	27	32	35	33

(b)

M

14	9	7	25	32

(c)

L | E | G

3	8	12	1	4	9	10	5	13	7	2	14	14	14	16	20	31	22	33	24	26	18	34	36	25	27	32	35	33

(d)

16	20	31	22	33

24	26	18	34	36

25	27	32	35	33

(e)

M

22	26	32

(f)

L | E | G

16	20	22	24	18	25	26	31	33	34	36	27	32	35	33

(g)