ΠΑΡΑΛΛΗΛΟΙ ΑΛΓΟΡΙΘΜΟΙ

ΕΔΙΠ Μαρία Λουκά

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών

13 Μαΐου 2020

Κεφάλαιο 5
Αναζήτηση (Searching)

Αναζήτηση (Searching)

Η αναζήτηση είναι μια από τις πλέον βασικές πράξεις στην περιοχή των υπολογισμών. Χρησιμοποιείται σε οποιαδήποτε εφαρμογή απαιτείται να βρεθεί αν ένα στοιχείο ανήκει σε μια λίστα ή πιο γενικά ανάκληση πληροφοριών από ένα αρχείο που σχετίζονται με αυτό το στοιχείο.

Το πρόβλημα της αναζήτησης διατυπώνεται ως εξής:
Δίνεται μια ακολουθία $S = \{s_1, s_2, ..., s_n\}$ ακεραίων αριθμών και ένας ακέραιος αριθμός x. Ζητείται να βρεθεί αν $x = s_k$ για κάποιο s_k στην S.

Αναζήτηση (Searching)

Στους ακολουθιακούς υπολογισμούς το πρόβλημα λύνεται με τη σάρωση της ακολουθίας S και τη σύγκρισή του x με τα διαδοχικά στοιχεία της μέχρις ότου είτε ένας ακέραιος είναι ίσος με x ή τελειώσουν τα στοιχεία της ακολουθίας.

Η μέθοδος αυτή είναι η σειριακή ή ακολουθιακή αναζήτηση και δίνεται από την procedure SEQUENTIAL SEARCH. Στη χειρότερη περίπτωση ο υπολογιστικός χρόνος του υποπρογράμματος είναι $O(n)$, το οποίο είναι βέλτιστο.

Αναζήτηση (Searching)

Διαδικασία 1: SEQUENTIAL SEARCH (S, x, k)

- **Βήμα 1**

(1.1) $i \leftarrow 1$

(1.2) $k \leftarrow 0$

- **Βήμα 2**

όσο $i \leq n$ *και* $k = 0$ **κάνε**

    αν $s_i = x$ **τότε**

        $k \leftarrow i$

    τέλος

    $i \leftarrow i + 1$

τέλος

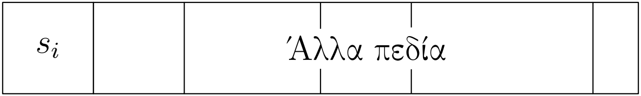
Αναζήτηση (Searching)

κάθε στοιχείο της S πρέπει να ελεγχθεί (όταν το x δε βρίσκεται στην S). Στην περίπτωση που η S είναι ταξινομημένη σε μη φθίνουσα διάταξη, τότε η BINARY SEARCH επιστρέφει τη θέση ενός στοιχείου της S που είναι ίσο με το x (ή 0 αν δεν υπάρχει τέτοιο στοιχείο σε $O(\log n)$ χρόνο.

Αναζήτηση σε ταξινομημένη ακολουθία

Υποθέτουμε ότι η ακολουθία $S = s_1, s_2, ..., s_n$ είναι ταξινομημένη σε μη φθίνουσα σειρά, δηλαδή, $s_1 \leq s_2 \leq ... \leq s_n$.

Το πιο σύνηθες πρόβλημα στις εφαρμογές είναι ένα αρχείο με n εγγραφές, οι οποίες είναι ταξινομημένες με βάση το κλειδί s (πεδίο). Χάριν απλότητας υποθέτουμε ότι τα s_i είναι διαφορετικά.



Σχήμα: Δομή εγγραφής σε αρχείο προς αναζήτηση

EREW Αναζήτηση

Υποθέτουμε ότι έχουμε N επεξεργαστές σε ένα EREW SM SIMD υπολογιστή για την αναζήτηση ενός δεδομένου στοιχείου του x.

1. Η τιμή του x γίνεται γνωστή σε όλους τους επεξεργαστές με την BROADCAST σε $O(\log N)$ χρόνο
2. η S χωρίζεται σε N υποακολουθίες με $\frac{n}{N}$ στοιχεία η κάθε μια και στον P_i επεξεργαστή καταχωρούνται τα στοιχεία $\{s_{(i-1)(\frac{n}{N})+1}, s_{(i-1)(\frac{n}{N})+2}, \ldots, s_{i(\frac{n}{N})}\}$
3. Όλοι οι επεξεργαστές εκτελούν την BINARY SEARCH στις υποακολουθίες τους, που συμβαίνει $O(\log(\frac{n}{N}))$ χρόνο στη χειρότερη περίπτωση. Επειδή τα στοιχεία της S είναι όλα διαφορετικά, το πολύ ένας επεξεργαστής θα εντοπίσει ένα s_k ίσο με το x και επιστρέφει το k
4. Συνολικός χρόνος που απαιτείται από αυτόν τον $EREW$ αλγόριθμο αναζήτησης: $O(\log N) + O(\log(\frac{n}{N}))$, που είναι $O(\log n)$.
   Επειδή ο χρόνος αυτός είναι ίσος με αυτόν της ακολουθιακής BINARY SEARCH <u>δεν επιτυγχάνεται αύξηση της ταχύτητας με τη μέθοδο αυτή!</u>

CREW Αναζήτηση

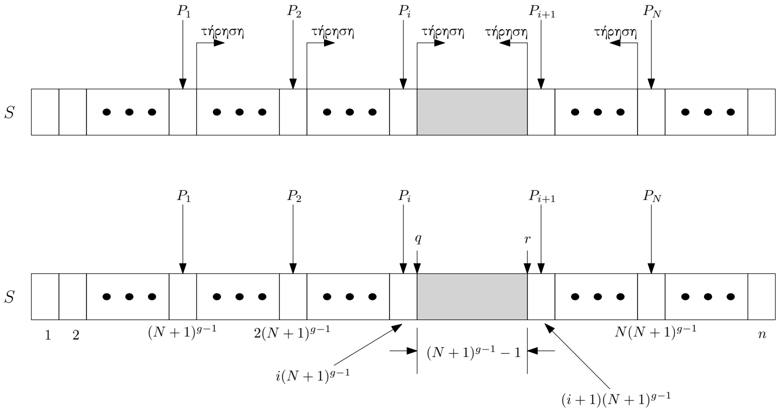
Ο αλγόριθμος για τον EREW υπολογιστή μπορεί να χρησιμοποιηθεί και στην παρούσα περίπτωση με τη διαφορά ότι τώρα όλοι οι επεξεργαστές μπορούν να διαβάσουν το x ταυτόχρονα σε σταθερό χρόνο και στη συνέχεια προχωρούν στην εκτέλεση της BINARY SEARCH για τις υπακολουθίες τους.

- Απαιτούμενος χρόνος: $O(\log(\frac{n}{N}))$ στη χειρότερη περίπτωση
- μικρότερος από τον ακολουθιακό χρόνο της BINARY SEARCH
- Είναι δυνατόν να επιτύχουμε καλύτερο χρόνο με την παραλληλοποίηση της δυαδικής αναζήτησης.

CREW Αναζήτηση

Στην παράλληλη αναζήτηση υπάρχουν διαθέσιμοι N επεξεργαστές και συνεπώς χρησιμοποιείται N+1-ιοστή αναζήτηση. Σε κάθε φάση, η ακολουθία χωρίζεται σε N+1 υποακολουθίες με ίσο πλήθος στοιχείων και οι N επεξεργαστές ελέγχουν ταυτόχρονα τα στοιχεία που βρίσκονται στο σύνορο μεταξύ διαδοχικών υποακολουθιών. Κάθε επεξεργαστής συγκρίνει το στοιχείο s του S με το x.

CREW Αναζήτηση



Σχήμα: Εξαγωγή του αριθμού σταδίων που απαιτούνται για αναζήτηση ακολουθίας

1. Αν $s > x$, τότε απορρίπτονται όλα τα στοιχεία μεγαλύτερα του s.
2. Αν $s < x$, τότε απορρίπτονται όλα τα στοιχεία μικρότερα του s.

CREW Αναζήτηση

- κάθε επεξεργαστής χωρίζει την ακολουθία σε δύο τμήματα. Το ένα τμήμα περιέχει όλα τα στοιχεία τα οποία απορρίπτονται καθώς σίγουρα δεν περιέχουν ένα στοιχείο ίσο με x και το άλλο τμήμα που περιλαμβάνει εκείνα τα στοιχεία τα οποία πιθανά να περιέχουν ένα στοιχείο ίσο με x.
- Το πρώτο τμήμα απορρίπτεται ενώ το δεύτερο διατηρείται. Η διαδικασία συρρικνώνει αρχική ακολουθία σε μια υποακολουθία που είναι η τομή όλων των τμημάτων που διατηρούνται, δηλαδή η υποακολουθία που βρίσκεται μεταξύ δύο στοιχείων που εξετάζονται σε αυτή τη φάση.
- Αυτή η υποακολουθία (γραμμοσκιασμένη στο Σχήμα) υφίσταται την ίδια διαδικασία στην επόμενη φάση. Η αναζήτηση συνεχίζεται μέχρις ότου είτε ένα στοιχείο βρεθεί ίσο με x ή απορριφθούν όλα τα στοιχεία της S.
- Επειδή κάθε φάση χρησιμοποιεί μια ακολουθία της οποίας το μήκος είναι $\frac{\text{μήκος προηγούμενης}}{N+1} - 1$, απαιτούνται $O(\log_{N+1}(n+1))$ φάσεις.

Στη συνέχεια αναπτύσσουμε πιο αναλυτικά τη μέθοδο.

CREW Αναζήτηση

Έστω g ο μικρότερος ακέραιος τέτοιος ώστε $n \leq (N+1) - 1$, δηλαδή $g = \lceil \log(n+1)/\log(N+1) \rceil$.
Θα δείξουμε με επαγωγή ότι g φάσεις είναι ικανές για την αναζήτηση μιας ακολουθίας μήκους n.

- Η ανωτέρω πρόταση είναι αληθής για $g = 0$.
- Υποθέτουμε ότι η πρόταση είναι αληθής για την αναζήτηση μιας ακολουθίας μήκους $(N+1)^{g-1} - 1$.
- Για την αναζήτηση μιας ακολουθίας μήκους $(N+1)^g - 1$ ο επεξεργαστής $P_i, i = 1, 2, ..., N$ συγκρίνει το x με το s_j, όπου $j = i(N+1)^{g-1}$. Μετά τη σύγκριση, η ακολουθία για αναζήτηση έχει μήκος $(N+1)^{g-1} - 1$ και είναι η γραμμοσκιασμένη, πράγμα που αποδεικνύει την πρόταση.

CREW Αναζήτηση

Η υποακολουθία αυτή προσδιορίζεται ως εξής: Κάθε επεξεργαστής P_i χρησιμοποιεί μια μεταβλητή c_i, η οποία λαμβάνει την τιμή $left$ ή $right$ ανάλογα με το τμήμα της ακολουθίας που αποφασίζει ο P_i να διατηρήσει βρίσκεται στα αριστερά ή δεξιά του στοιχείου που συγκρίθηκε με το x κατά τη διάρκεια αυτής της φάσης.

CREW Αναζήτηση

Αρχικά, η τιμή κάθε c_i είναι αυθαίρετη, με σταθερές τις τιμές των $c_0 = right$ και $c_{N+1} = left$. Μετά τη σύγκριση μεταξύ του x και ενός στοιχείου s_{j_i} της S, ο P_i καταχωρεί μια τιμή στο c_i (εκτός εαν $s_{j_i} = x$). Αν $c_i \neq c_{i-1}$ για κάποιο $i, 1 \leq i \leq N$, τότε η ακολουθία που πρόκειται να αναζητηθεί θα αρχίζει από το s_q και θα τελειώνει στο s_r, όπου:

$$q = (i-1)(N+1)^{g-1} - 1 \quad \text{και} \quad r = i(N+1)^{g-1} - 1$$

Ένας μόνον επεξεργαστής ενημερώνει τα q και r στην κοινή μνήμη και όλοι οι υπόλοιποι επεξεργαστές μπορούν ταυτόχρονα να διαβάσουν τις ενημερωμένες τιμές σε σταθερό χρόνο.

CREW Αναζήτηση

Διαδικασία 2: CREW SEARCH (S, x, k)

- **Βήμα 1** {Αρχικές τιμές στους δείκτες των ακολουθιών που θα σαρωθούν}

(1.1) $q \leftarrow 1$
(1.2) $r \leftarrow n$

- **Βήμα 2** {Αρχικές τιμές για τα αποτελέσματα και το μέγιστο αριθμό σταδίων}

(2.1) $k \leftarrow 0$
(2.2) $g \leftarrow \left\lceil \frac{\log(n+1)}{\log(N+1)} \right\rceil$
(2.3) $c_0 = right$
(2.4) $c_{NH} = left$

• **Βήμα 3**

όσο $(q \leq r$ ***και*** $k = 0)$ **κάνε**

(3.1) $j_0 \leftarrow q - 1$

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

(i) $j_i \leftarrow (q-1) + i(N+1)^{g-1}$

{Ο επεξεργαστής P_i συγκρίνει το x με το s_j και προσδιορίζει το μέρος της ακολουθίας που θα διατηρηθεί}

(ii) **αν** $j_i \leq r$ **τότε**

αν $s_j i = x$ **τότε**

$k \leftarrow j_i$

αλλιώς

αν $s_{j_i} > x$ **τότε**

$c_i \leftarrow left$

αλλιώς

$c_i \leftarrow right$

τέλος

τέλος

αλλιώς

(a) $j_i \leftarrow r + 1$

(b) $c_i \leftarrow left$

{Σημαίνει ότι $j_i > r$ δείχνει εκτός της ακολουθίας και γίνεται διορθωτική ενέργεια}

τέλος

{υπολογίζονται οι δείκτες της ακολουθίας που θα σαρωθεί στην επόμενη επανάληψη}

(iii) **αν** $c_i \neq c_{i-1}$ **τότε**

(a) $q \leftarrow j_{i-1} + 1$

(b) $r \leftarrow j_i - 1$

τέλος

(iv) **αν** $(i = N$ ***και*** $c_i \neq c_{i+1})$ **τότε**

$q \leftarrow j_i + 1$

τέλος

τέλος

(3.3) $g \leftarrow g - 1$

τέλος

CREW Αναζήτηση

Ανάλυση

Τα βήματα 1, 2, 3.1 και 3.2 εκτελούνται από έναν επεξεργαστή, έστω τον P_1 σε σταθερό χρόνο. Το βήμα 3.2 εκτελείται επίσης σε σταθερό χρόνο. Όπως αποδείχθηκε, το βήμα 3 εκτελείται το πολύ σε g επαναλήψεις. Συνεπώς η CREW SEARCH εκτελείται σε $O\left(\frac{\log(n+1)}{\log(N+1)}\right)$ χρόνο, δηλαδή $t(n) = O(\log_{N+1}(n+1))$ συνεπώς $c(n) = O(\log_{N+1}(n+1))$ το οποίο δεν είναι βέλτιστο.

CREW Αναζήτηση

Παράδειγμα

Έστω $S = \{1, 4, 6, 9, 10, 11, 13, 14, 15, 18, 20, 23, 32, 45, 51\}$ η ακολουθία, η οποία πρόκειται να αναζητηθεί χρησιμοποιώντας ένα CREW SM SIMD υπολογιστή.

1. Υποθέτουμε ότι $N = 3, x = 45$. Αρχικά $q = 1, r = 15, k = 0$ και $g = 2$. Στην πρώτη επανάληψη του βήματος 3, ο P_1 υπολογίζει $j_1 = 4$ και συγκρίνει το s_4 με το x. Επειδή $9 < 45$, $c_1 = right$. Ταυτόχρονα οι P_2 και P_3 συγκρίνουν τα s_8 και s_{12} με το x, αντίστοιχα. Επειδή $14 < 45$ και $23 < 45$, $c_2 = right$ και $c_3 = right$. Αλλά $c_3 \neq c_4$, συνεπώς $q = 13$ και το r παραμένει αμετάβλητο. Η νέα ακολουθία που θα αναζητηθεί ξεκινά από το s_{13} μέχρι το s_{15} με $g = 1$. Στη δεύτερη επανάληψη, ο P_1 υπολογίζει $j_1 = 12 + 1$ και συγκρίνει το s_{13} με το x. Επειδή $32 < 45$, $c_1 = right$. Ταυτόχρονα, ο P_2 συγκρίνει το s_{14} με το x και επειδή είναι ίσα, θέτει $k = 14$. Επίσης, ο P_3 συγκρίνει το s_{15} με το x. Επειδή $51 > 45$, $c_3 = left$. Τώρα, $c_3 \neq c_2$. Συνεπώς, $q = 12 + 2 + 1 = 15$ και $r = 12 + 3 - 1 = 14$ και η διαδικασία σταματά με $k = 14$.

CREW Αναζήτηση

2. Έστω ότι $x = q$ με $N = 3$. Στην πρώτη επανάληψη, ο P_1 συγκρίνει το s_4 με το x και επειδή είναι ίσα, θέτει $k = 4$.
3. Έστω ότι $N = 2$ και $x = 21$. Αρχικά, $g = 3$. Στην πρώτη επανάληψη ο P_1 υπολογίζει $j_1 = 9$ και συγκρίνει το s_9 με το x, επειδή $15 < 21, c_1 = right$. Ταυτόχρονα ο P_2 υπολογίζει $j_2 = 16$ και $c_2 = left$. Τώρα, $c_2 \neq c_1$, συνεπώς $q = 10$ και $r = 15$ έτσι η ακολουθία που θα αναζητηθεί θα είναι από s_{10} μέχρι s_{15} και $g = 2$.
   Στη δεύτερη επανάληψη, ο P_1 υπολογίζει $j_1 = q + 3$ και συγκρίνει το s_{12} με το x, επειδή $51 > 21, c_2 = left$. Τώρα $c_1 \neq c_0$ και συνεπώς r_{11} και το q παραμένει αμετάβλητο. Στην τελική επανάληψη, $q = 1$ και ο P_1 υπολογίζει $j_1 = q + 1$, συγκρίνει το s_{10} με το x. Επειδή $18 < 21,\ c_1 = right$. Ταυτόχρονα, ο P_2 υπολογίζει $j_2 = q + 2$ και συγκρίνει το s_{11} με το x, επειδή $20 < 21, c_2 = right$. Τώρα, $c_2 \neq c_3$ και $q = 12$. Επειδή $q > r$ η διαδικασία σταματά ανεπιτυχώς και επιστρέφει $k = 0$.

CREW Αναζήτηση