Αυτορυθμιζόμενα Δυαδικά Δένδρα Αναζήτησης

Βησσαρίων Φυσικόπουλος

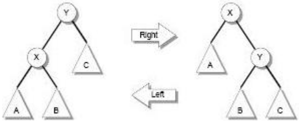
2008

Ορισμός Προβλήματος (Μοντέλο)

- Δυαδικό Δένδρο Εύρεσης T
- n κόμβοι-στοιχεία $1, 2, \ldots, n$ (συμμετρική διάταξη)
- σ: ακολουθία m αιτήσεων κάποιων στοιχείων του T
- $cost_j$ (*κόστος* αίτησης j ενός στοιχείου i) $= d(i)$ (*βάθος* του κόμβου i στο T)
- Αυθαίρετου πλήθους επαναζυγιστικές πράξεις στο T μετά από κάθε αίτηση με σταθερό κόστος
- Κόστος Αλγορίθμου: $cost_{algorithm_A} = \sum_{j=1}^{m} cost_j$
- online αλγόριθμος - άγνωστη σ, offline αλγόριθμος - γνωστή σ
- Competitive Ratio: $cost_{online}(\sigma) \leq CR \cdot cost_{offline}(\sigma) + k$
- Competitive Ratio c: c-competitive

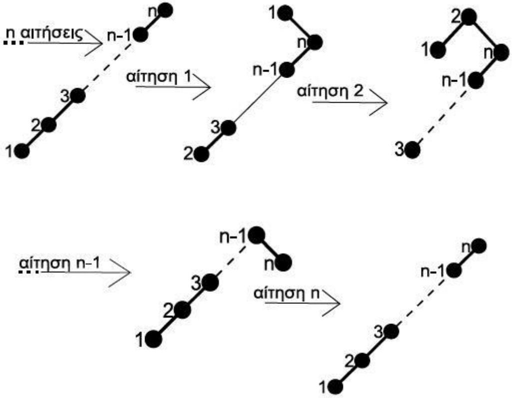
Μια πρώτη λύση

- Οποιοδήποτε Ισοζυγισμένο Δυαδικό Δένδρο Εύρεσης T είναι $O(\log n)$-competitive
- Αυτορυθμιζόμενα Δένδρα. Φυσικός αλγόριθμος: μετά από την ικανοποίηση μιας αίτησης του στοιχείου i, μετακινούμε το i στη ρίζα
- Ερώτημα: τι πράξεις θα χρησιμοποιήσει ο αλγόριθμος για να μετακινήσει ένα στοιχείο στη ρίζα;
- *Απλή λύση:* Ο αλγόριθμος *move to root* μετακινεί το στοιχείο που προσπελάστηκε στη ρίζα εκτελώντας μια σειρά από rotations



Μια πρώτη λύση (Κάτω Φράγμα)

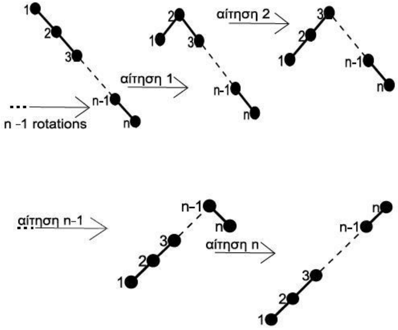
- Έστω η ακολουθία $\sigma^k = 1, 2, \ldots, n$ στο T
- Μετά τις πρώτες n αιτήσεις το T εκφυλίζεται σε ένα αριστερό μονοπάτι.
- Στη συνέχεια κάθε αίτηση έχει κόστος n.



- move to root αλγόριθμος: amortized cost n για μεγάλο k

Βέλτιστος Offline Αλγόριθμος

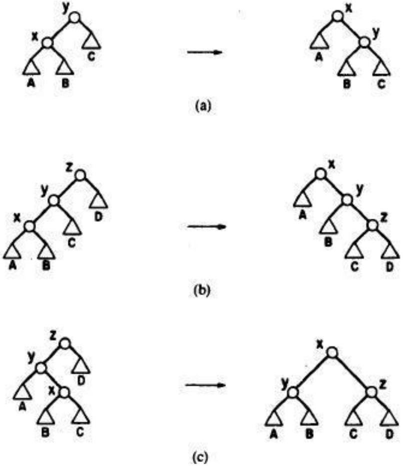
- Έστω η ακολουθία $\sigma^k = 1, 2, \ldots, n$ στο T
- Σε $n-1$ rotations ο αλγόριθμος εκφυλίζει το T σε ένα δεξί μονοπάτι.
- Στη συνέχεια κάθε αίτηση έχει κόστος 1.



- βέλτιστος offline αλγόριθμος: amortized cost $= 2 - \frac{1}{n}$
- Συμπέρασμα: Ο αλγόριθμος *move to root* είναι $\Theta(n)$-competitive

Splay Trees (Sleator και Tarjan [2])

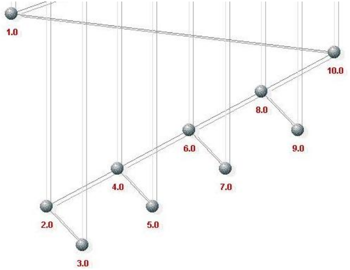
- splay πράξεις: (a) zig, (b) zig-zig, (c) zig-zag



- splaying αλγόριθμος:
  1. Εκτελεί ένα ψάξιμο στο δένδρο T για το ζητούμενο στοιχείο i
  2. Χρησιμοποιώντας splay πράξεις ανεβάζει το i στη ρίζα

Βελτίωση προηγούμενου παραδείγματος

- Η ακολουθία $\sigma^k = 1, 2, \ldots, n$ στο T αν το T είναι splay tree
- Μετά τις πρώτες n αιτήσεις το T εκφυλίζεται σε ένα αριστερό μονοπάτι.
- Αλλά μετά την αίτηση του 1 το splay δένδρο είναι πιο ισορροπημένο



Θεώρημα

(Scanning [2, 3]): *Σε ένα splay δένδρο n κόμβων, μια ακολουθία n προσπελάσεων που προσπελαύνει ένα διαφορετικό στοιχείο κάθε φορά με βάση τη συμμετρική διάταξη έχει συνολικό χρόνο προσπέλασης* $O(n)$.

Access Lemma

Ορισμοί

- Σε κάθε κόμβο i αντιστοιχίζουμε ένα θετικό βάρος w_i
- $size : s_i = \sum_{j \in T_i} w_j$
- $rank : r_i = \log s_i$
- συνάρτηση δυναμικού $\Phi = \sum_{i \in T} r_i$

Λήμμα

([2]) *Το κόστος της αίτησης στον κόμβο x είναι το πολύ* $3(r_t - r_x) + 1$ *όπου t η ρίζα του δένδρου.*

Σκιαγράφηση Απόδειξης

- *φράσσουμε το $cost_j + \Phi_j - \Phi_{j-1}$ για κάθε splay πράξη*
- *Χρησιμοποιούμε*
  1. *ιδιότητες που προκύπτουν από το σχήμα των πράξεων splay*
  2. *ιδιότητα των λογαρίθμων:* $\max_{\substack{x+y \le 1 \\ x,y \ge 0}} (\log x + \log y) = -2$
- *Συνολικό κόστος προσπέλασης ακολουθίας:* $\sum_{j=1}^{m} cost_j + \sum_{i=1}^{n} \log s_i - \sum_{i=1}^{n} \log w_i$

Balance Theorem

Θεώρημα

([2]) *Σε ένα splay δένδρο n κόμβων, μια ακολουθία m προσπελάσεων έχει συνολικό χρόνο προσπέλασης* $O((m+n)\log n + m)$.

Σκιαγράφηση Απόδειξης

- $w_i = 1/n$ *άρα* $W = \sum_{i=1}^{n} w_i = 1$
- *κόστος μιας προσπέλασης:* $3(r_t - r_x) + 1 = 3\log \frac{W}{\sum_{i \in T_x} w_i} + 1 = 3\log n + 1$
- *μέγιστη μείωση δυναμικού:* $\log s_i - \log w_i \leq \log \frac{W}{w_i} \leq n \log n$
- *συνολικό κόστος:* $m(3\log n + 1) + \Phi_m - \Phi_0 = O((m+n)\log n + m)$

Static Optimality Theorem

Θεώρημα

([2]) *Σε ένα splay δένδρο n κόμβων, αν κάθε κόμβος i προσπελαύνεται τουλάχιστον μια φορά με συχνότητα $f(i)$ τότε μια ακολουθία m προσπελάσεων έχει συνολικό χρόνο προσπέλασης.*

$$O\left(m + \sum_{i=1}^{n} f(i) \log\left(\frac{m}{f(i)}\right)\right)$$

Σκιαγράφηση Απόδειξης

- $w_i = f(i)/m$ *άρα* $W = 1$
- *κόστος μιας προσπέλασης:* $3\log(m/f(i)) + 1$
- *μέγιστη μείωση δυναμικού:* $\sum_{i=1}^{n} \log \frac{m}{f(i)}$
- *συνολικό κόστος:* $\sum_{i=1}^{n} 3f(i) \log \frac{m}{f(i)} + \sum_{i=1}^{n} \log \frac{m}{f(i)} + m$

Static Finger Theorem

Θεώρημα

([2]) *Σε ένα splay δένδρο n κόμβων, μια ακολουθία m προσπελάσεων $i_1, i_2, \ldots, i_m$, όπου έχουμε αριθμήσει τα στοιχεία από 1 μέχρι n με βάση τη συμμετρική διάταξη και s ένα σταθερό στοιχείο, έχει συνολικό χρόνο προσπέλασης*

$$O(n \log n + m + \sum_{k=1}^{m} \log(|i_k - s| + 1))$$

Σκιαγράφηση Απόδειξης

- *s ένα σταθερό στοιχείο*
- $w_i = \frac{1}{(|i-s|+1)^2}$ *άρα*
  $W = \sum_{i=1}^{n} \frac{1}{((|i-s|+1)^2} \leq 2\sum_{k=1}^{\infty} \frac{1}{k^2} = O(1)$
- *κόστος k προσπέλασης:* $6\log(|i_k - s| + 1) + 1$
- *μέγιστη μείωση δυναμικού:* $2\sum_{i=1}^{n} \log n = 2n \log n$ *αφού* $w_i \leq 1/n^2$

Working Set Theorem (1)

Αρχικά κάποιοι ορισμοί...

- Έστω $1, \ldots, m$ αιτήσεις
- j αίτηση, στοιχείο i
- $t(j)$= πλήθος διαφορετικών στοιχείων που προσπελάστηκαν στο χρονικό διάστημα μεταξύ των 2 τελευταίων προσπελάσεων του στοιχείου i
- π.χ. για την ακολουθία $1, 3, 2, 4, 3, 5, 6, \ldots$ $t(7) = 5$ και $t(5) = 2$

Θεώρημα

([2]) *Σε ένα splay δένδρο n κόμβων, μια ακολουθία m προσπελάσεων όπου έχουμε αριθμήσει τα στοιχεία από 1 μέχρι n με βάση τη συμμετρική διάταξη έχει συνολικό χρόνο προσπέλασης*

$$O(n \log n + m + \sum_{j=1}^{m} \log(t(j) + 1))$$

Working Set Theorem (2)

Σκιαγράφηση Απόδειξης

- *βάρη:* $1, \frac{1}{4}, \frac{1}{9}, \ldots, \frac{1}{k^2}, \ldots, \frac{1}{n^2}$
- *πιο πρόσφατα προσπελάσιμος κόμβος βάρος* 1
- *Έστω j αίτηση, i στοιχείο, ανακατατάξεις στα βάρη ως εξής:*
  - $w'_k = \frac{1}{\sqrt{\frac{1}{w_k}+1}}$ *για κάθε κόμβο k που $w_k > w_i$*
  - $w_i = 1$
- *Για κάθε κόμβο i*
  - $s_i = \sum_{i=1}^{n} 1/i^2$, *αν i ρίζα*
  - s_i *μειώνεται ή παραμένει σταθερό, αν i εσωτερικός κόμβος*
  - *Συμπέρασμα: οι ανακατατάξεις στα βάρη μειώνουν ή διατηρούν σταθερό το δυναμικό*

Working Set Theorem (3)

Σκιαγράφηση Απόδειξης

(συνέχεια...)

- $W = \sum_{i=1}^{n} 1/k^2 = O(1)$
- *άρα* $w_i = \frac{1}{(t(j)+1)^2}$
- *συνολικό κόστος των προσπελάσεων*

$$O(\sum_{j=1}^{m} \log(t(j)+1))$$

- *μείωση δυναμικού είναι το πολύ*

$$\sum_{i=1}^{n} \log \frac{W}{w_i} = \sum_{i=1}^{n} \log(t(j)+1)^2 = O(n \log n)$$

Ιδιότητες Splay Tree

1. Ισοζυγισμένο Δυαδικό Δένδρο Αναζήτησης

   [Balance Theorem]

2. Βέλτιστο Στατικό Δυαδικό Δένδρο (θεωρία πληροφορίας)

   [Static Optimality Theorem]

3. Finger Δένδρο Αναζήτησης

   [Static Finger Theorem]

   - Finger Δένδρο Αναζήτησης: αναζήτηση ενός στοιχείου που απέχει απόσταση d από την αρχή ή το τέλος της λίστας γίνεται σε χρόνο $O(\log d)$.

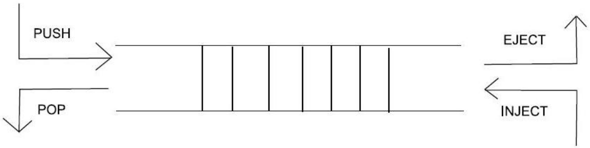
4. Χρόνος Προσπέλασης Στοιχείου $i = \log(1 + workingset(i))$

   [Working Set Theorem]

   - working set(i) = πλήθος διαφορετικών στοιχείων που προσπελάστηκαν από την τελευταία προσπέλαση του i
   - Όταν ένα στοιχείο προσπελαστεί το κόστος της επόμενης προσπέλασης του αυξάνει λογαριθμικά με το πλήθος των διαφορετικών στοιχείων που προσπελαύνονται

Deque (1)

- Deque (διπλή ουρά)



- i θέση ουράς$=i$ στοιχείο T(συμμετρική διάταξη)
- πράξεις deque : PUSH, POP, INJECT, EJECT

  POP Εφαρμόζουμε splay πράξεις στο πιο αριστερό φύλλο του T και τον αφαιρούμε από το T.

  PUSH Ο κόμβος που εισάγεται γίνεται η νέα ρίζα του T με αριστερό παιδί κενό και δεξί παιδί την παλιά ρίζα.

Deque (2)

Εικασία

(Deque [4]) *Το κόστος της εκτέλεσης μιας ακολουθίας m πράξεων διπλής ουράς σε ένα οποιοδήποτε δυαδικό δένδρο n κόμβων χρησιμοποιώντας splay πράξεις είναι $O(m+n)$.*

Θεώρημα

(Restricted Deque[4]) *Μια ακολουθία m οποιωνδήποτε πράξεων POP, PUSH και INJECT σε μια διπλή ουρά υλοποιημένη με ένα splay δένδρο που περιέχει αρχικά n κόμβους έχει συνολικό χρόνο εκτέλεσης $O(n+m)$.*

Dynamic Finger

Ο Cole στο [1] αποδεικνύει μια απο τις 3 εικασίες του [2]

Θεώρημα

(Dynamic Finger [2]) *Σε ένα splay δένδρο n κόμβων, μια ακολουθία m προσπελάσεων έχει συνολικό χρόνο προσπέλασης*

$$O(m + n + \sum_{j=1}^{m-1} \log(|i_{j+1} - i_j| + 1))$$

όπου i_j το στοιχείο $i \in [1, m]$ που προσπελαύνεται κατά την j αίτηση.

Ανοικτά Προβλήματα

Εικασία

(Dynamic Optimality [2]) *Το splay δένδρο είναι έχει λόγο ανταγωνισμού* $O(1)$.

Εικασία

(Traversal [2]) *Έστω* T_1, T_2 *δύο δυαδικά δένδρα αναζήτησης με* n *ακριβώς ίδιους κόμβους το καθένα. Έστω ότι προσπελαύνουμε τους κόμβους του* T_1 *με βάση την προδιάταξη τους στο* T_2, *και εφαρμόζουμε σε αυτούς τις splay πράξεις τότε ο συνολικός χρόνος προσπέλασης είναι* $O(n)$.

Αναφορές

R. Cole.
On the dynamic finger conjecture for splay trees. part ii: The proof.
30:44–85, 2000.

D. D. Sleator and R. E. Tarjan.
Self-adjusting binary search trees.
J. Assoc. Comput. Mach., 32(3):652–686, 1985.

R. Sundar.
Twists, turns, cascades, deque conjecture, and scanning theorem.
In *FOCS*, pages 555–559, 1989.

R. E. Tarjan.
Sequential access in splay trees takes linear time.
Combinatorica, 5(4):367–378, 1985.

Τέλος...