ΑΥΤΟΡΓΑΝΟΥΜΕΝΕΣ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ (Splay Trees)

ΜΠΟΜΠΟΤΑΣ ΑΓΟΡΑΚΗΣ

mpompotas@ceid.upatras.gr

Εισαγωγή

- Μία τέτοιου είδους δομή βρίσκεται σε τυχαία αρχική κατάσταση, αλλά κατά τη διάρκεια κάθε πράξης εφαρμόζεται ένας απλός ανακατασκευαστικός κανόνας με σκοπό την βελτίωση της αποδοτικότητας των μελλοντικών πράξεων

Πλεονεκτήματα – Μειονεκτήματα

- Επίτευξη αποδοτικότητας με την επιμερισμένη έννοια
- Εφόσον προσαρμόζονται ανάλογα με τη χρήση, μπορούν να είναι πολύ πιο αποδοτικές αν το μοτίβο χρήσης είναι παράξενο
- Δεν αποθηκεύουν καμία πληροφορία ισορροπίας → Απαίτηση για μικρότερο χώρο
- Έχουν απλούς και εύκολα υλοποιήσιμους αλγόριθμους προσπέλασης και ανανέωσης
- Απαιτούν περισσότερες τοπικές αναπροσαρμογές → μεγαλύτερα έξοδα αναδιοργάνωσης

Αυτοργανούμενες γραμμικές λίστες

- Πολύ απλές λίστες που αποθηκεύουν στοιχεία
- Πράξεις :
  - $Access(x)$
  - $Insert(x)$
  - $Delete(x)$
- Για την υλοποίησή τους μπορούμε να χρησιμοποιήσουμε συνδεδεμένες λίστες ή πίνακες

Αυτοργανούμενες γραμμικές λίστες

- $pos(x_i) \rightarrow$ η θέση του στοιχείου x_i στη λίστα
- Κάθε πράξη προσπελάζει γραμμικά τα στοιχεία από το αριστερότερο άκρο έως και την εύρεση του κατάλληλου στοιχείου ή το τέλος της λίστας
- Άρα,
  - Οι $Access(x_i)$ και $Delete(x_i)$ κοστίζουν $pos(x_i)$
  - Η $Insert(x_i)$ κοστίζει $|S| + 1$.

Στρατηγικές αυτοργάνωσης

- Κανόνας μετακίνησης στην αρχή (Move to Front Rule - MFR)
  - Οι πράξεις $Access(x_i)$ και $Insert(x_i)$ μετακινούν το x στην αρχή της λίστας → Δεν μεταβάλλεται η σειρά των υπόλοιπων στοιχείων
  - Η $Delete(x_i)$ διαγράφει το στοιχείο x από τη λίστα
- Κανόνας Αντιμετάθεσης – Transposition Rule
  - Η $Access(x_i)$ εναλλάσσει το x με το προηγούμενο στοιχείο
  - Η $Insert(x_i)$ κάνει το x προτελευταίο στοιχείο της λίστας

Splay Trees

- Επιτυγχάνουν **πολύ καλή επιμερισμένη πολυπλοκότητα** για κάθε πράξη χωρίς να αποθηκεύουν τα βάρη κάθε στοιχείου.
- Στρατηγική επανοργάνωσης: **splaying**
- Κατά τη διάρκεια κάθε πράξης το εμπλεκόμενο στοιχείο μεταφέρεται στη ρίζα με διαδοχικές περιστροφές
  - οι επόμενες πράξεις που θα χρειαστούν αυτό το στοιχείο θα είναι πιο φθηνές

Splay Trees – Επανοργανωτικές πράξεις

Έστω ένα στοιχείο x ε S και p(x) ο πατέρας του x. Για να μεταφέρω το x στη ρίζα επαναλαμβάνουμε το splaying ως εξής:

Περίπτωση 1:

- Ο p(x) είναι ρίζα

Επανοργάνωση:

- Κάνουμε απλή περιστροφή και φέρνουμε το x στη ρίζα

Splay Trees – Επανοργανωτικές πράξεις

Έστω ένα στοιχείο x ε S και p(x) ο πατέρας του x. Για να μεταφέρω το x στη ρίζα επαναλαμβάνουμε το splaying ως εξής:

Περίπτωση 2:

- Ο p(x) δεν είναι ρίζα
- Οι x, p(x) είναι και οι 2 αριστερά ή δεξιά παιδιά του γονέα τους

Επανοργάνωση:

- Κάνουμε απλή περιστροφή στο p(x) με τον πατέρα του p(p(x))
- Κάνουμε άλλη μία περιστροφή της ίδιας μορφής με την πρώτη στο x και το p(x)

Splay Trees – Επανοργανωτικές πράξεις

Έστω ένα στοιχείο x ε S και p(x) ο πατέρας του x. Για να μεταφέρω το x στη ρίζα επαναλαμβάνουμε το splaying ως εξής:

Περίπτωση 3:

- Ο p(x) είναι ρίζα
- Οι x, p(x) δεν είναι του ίδιου είδους παιδιά

Επανοργάνωση:

- Εκτελούμε διπλή περιστροφή στον x