



Εργαστήριο 8

Κατακερματισμός

Στην ενότητα αυτή θα μελετηθούν:

Robin Hood Hashing

Διδάσκων: Γιώργος Χατζηπολλάς

Robin Hood Hashing

Η τεχνική κατακερματισμού Robin Hood

- Αποτελεί μορφή κατακερματισμού **ανοικτής διεύθυνσης** η οποία προσπαθεί να εξισορροπήσει τους χρόνους αναζήτησης των στοιχείων χρησιμοποιώντας μια πιο **δίκαιη στρατηγική χειρισμού των συγκρούσεων**.
 - μειώνει τον μέσο όρο του "προβλεπτικού μήκους" αναζήτησης, περιορίζοντας το πόσο μακριά από την αρχική του θέση έχει μετακινηθεί το κάθε στοιχείο λόγω συγκρούσεων
- Είναι αποδοτική για περιπτώσεις που θέλουμε να ελαχιστοποιήσουμε το χρόνο αναζήτησης στοιχείων σε ένα γεμάτο πίνακα κατακερματισμού
 - Αποδεικνύεται ότι το αναμενόμενο κόστος αναζήτησης χειρότερης περίπτωσης είναι φραγμένο από $\Theta(\log \log n)$

Robin Hood Hashing

- Σε κάθε θέση του πίνακα κατακερματισμού εισάγεται πέραν του κλειδιού και μια ακέραια τιμή, έστω `probeLength`, η οποία καθορίζει το πόσο μακριά από την αρχική του θέση (που ορίζεται από τη συνάρτηση κατακερματισμού) έχει εισαχθεί το κλειδί.
- Κατά την εισαγωγή ενός κλειδιού και σε περίπτωση σύγκρουσης, εφαρμόζουμε τη στρατηγική του Ρομπέν των Δασών
 - Το στοιχείο που είναι μειονεκτικά, δηλαδή βρίσκεται σε μεγαλύτερη απόσταση από την αρχική του θέση, κρατεί τη θέση και αναζητά καινούρια θέση το στοιχείο που πλεονεκτεί.

Εισαγωγή κλειδιού στον πίνακα κατακερματισμού

1. **Εύρεση Θέσης:** Υπολογίζουμε τη θέση του νέου κλειδιού σύμφωνα με τη συνάρτηση κατακερματισμού.
2. **Έλεγχος για Σύγκρουση:** Αν η θέση είναι κατειλημμένη, συγκρίνουμε τα `probeLength` των δύο κλειδιών.
 1. Αν το νέο κλειδί έχει μεγαλύτερο `probeLength` από το υπάρχον, αλλάζουν θέσεις. Το νέο κλειδί καταλαμβάνει τη θέση και το παλιό συνεχίζει να ψάχνει για νέα θέση.
 2. Διαφορετικά, το νέο κλειδί συνεχίζει τη διαδικασία στην επόμενη θέση του πίνακα.
3. **`maxProbeLength`:** Για βελτιστοποίηση της αναζήτησης ενός κλειδιού, μας ενδιαφέρει η μέγιστη απόσταση κλειδιού από την αρχική θέση. Αν η καινούρια εισαγωγή τοποθετηθεί σε απόσταση `ProbeLength > maxProbeLength` θέτουμε `maxProbeLength = ProbeLength`

Αναζήτηση κλειδιού στον πίνακα κατακερματισμού

Η αναζήτηση είναι παρόμοια με την εισαγωγή ενός κλειδιού.

1. Υπολογίζουμε τη θέση του κλειδιού σύμφωνα με τη συνάρτηση κατακερματισμού.
 1. Αν το κλειδί βρίσκεται στη θέση αυτή την επιστρέφουμε
 2. Αν δε βρίσκεται εκεί, το αναζητούμε στην επόμενη θέση
2. Η διαδικασία αναζήτησης (δεδομένου ότι δεν έχουμε αφαιρέσεις κλειδιών) σταματά όταν είτε εντοπίσουμε κάποιο κενό κελί είτε όταν ξεπεράσουμε το `maxProbeLength`

Παράδειγμα Εισαγωγών/Αναζητήσεων

Πίνακας κατακερματισμού μεγέθους 7

Συνάρτηση κατακερματισμού $\text{index} = \text{key} \bmod 7$

Εισαγωγή 10: $[_, _, _, (10, 0), _, _, _]$ $\text{maxProbeLength} = \text{mPL} = 0$

Εισαγωγή 22: $[_, (22, 0), _, (10, 0), _, _, _]$ $\text{mPL} = 0$

Εισαγωγή 31: $[_, (22, 0), _, (10, 0), (31, 1), _, _]$ $\text{mPL} = 1$

Εισαγωγή 4: $[_, (22, 0), _, (10, 0), (31, 1), (4, 1), _]$ $\text{mPL} = 1$

Αναζήτηση 7: Η θέση 0 είναι κενή, το κλειδί 7 δεν υπάρχει

Αναζήτηση 17: Η θέση 3 είναι κατειλημμένη, η θέση 4 είναι κατειλημμένη, το $\text{maxProbeLength} = 1$ αποκλείεται το 17 να έχει τοποθετηθεί πιο μακριά

Αναζήτηση 4: Η θέση 4 είναι κατειλημμένη, η θέση 4 περιέχει το κλειδί 4

Εισαγωγή 15: $[_, (22, 0), (15, 1), (10, 0), (31, 1), (4, 1), _]$ $\text{mPL} = 1$

Εισαγωγή 16: $[_, (22, 0), (15, 1), (16, 1), (31, 1), (10, 2), (4, 2)]$ $\text{mPL} = 2$

Υλοποίηση Robin Hood Hashing

Δημιουργείστε τις κλάσεις *RobinHoodHashing* και *Element*.

- Χαρακτηριστικά κλάσης *RobinHoodHashing*
 - `table`: ο πίνακας κατακερματισμού για τη φύλαξη των *Elements*
 - `capacity`: η χωρητικότητα του πίνακα
 - `size`: το πλήθος των στοιχείων στον πίνακα
 - `maxProbeLength`: η μέγιστη απόσταση κλειδιού από την αρχική θέση φύλαξης
- Χαρακτηριστικά κλάσης *Element*
 - `key`: είναι το κλειδί που θα φυλαχτεί
 - `probeLength`: η απόσταση από την αρχική θέση φύλαξης του κλειδιού
 - Για την εργασία → ο `trieNode` στον οποίο μας οδηγεί ο συγκεκριμένος χαρακτήρας
- Μέθοδοι
 - `constructor`: Αρχικό μήκος πίνακα 5
 - `insert`
 - `search`
 - `rehash`: Θα εκτελείται επανακατακερματισμός όταν το πλήθος των κλειδιών ξεπεράσει το 90% της χωρητικότητας. Οι καινούριες χωρητικότητες θα είναι 11, 19 και 29.