

Linear Hashing

1. Σύντομη Περιγραφή – Οδηγίες

Μετά από επιτυχημένο compilation μέσω της εντολής **javac *.java** εντός του πακέτου **mypackage** το πρόγραμμα μπορεί να εκκινηθεί. Καλείται με τη χρήση της εντολής:

java mypackage.MyHashing %FILEPATH%

Με βάση το filepath που δόθηκε ως argument γίνεται έλεγχος του αντίστοιχου δυαδικού αρχείου. Σε περίπτωση που το αρχείο δεν πληρεί τις προδιαγραφές ή το παρεχόμενο filepath είναι ελαττωματικό τότε εμφανίζεται μήνυμα σφάλματος και το πρόγραμμα τερματίζει. Διαφορετικά, γίνεται ανάγνωση των αριθμών του αρχείου. Οι αριθμοί εισάγονται σε μια δομή και διαδοχικά περνούν στον Hash Table. Οι απαραίτητες μετρήσεις απόδοσης καταγράφονται στο ενδιάμεσο. Στην συγκεκριμένη εφαρμογή δεν παρέχεται κάποιο μενού χρήστη, εκτυπώνονται ωστόσο στοιχισμένα τα αποτελέσματα των μετρήσεων, βάσει των οποίων συντάχθηκε και η αναφορά για την απόδοση του προγράμματος.

2. Διαδικασία λύσης – Παρατηρήσεις

Η υλοποίηση του πίνακα κατακερματισμού ακολουθεί σε μεγάλο βαθμό τον κώδικα που διατίθεται από τις σημειώσεις στον ιστοχώρο του μαθήματος. Οι βασικές κλάσεις για το hashing είναι:

- **HashBucket:** Η κλάση που δίνει υπόσταση στην έννοια της θέσης του Hash Table. Περιέχει έναν πίνακα με ακέραια κλειδιά και ένα αντικείμενο τύπου HashBucket προς υλοποίηση της αλυσίδας θέσεων σε περιπτώσεις overflow. Διαθέτει μεθόδους που αντιστοιχούν σε πράξεις εισαγωγής, διαγραφή, αναζήτησης, συγχώνευσης και διαχωρισμού.
- **LinearHashing:** Η wrapper κλάση της εφαρμογής που υλοποιεί τον πίνακα με τις θέσεις καταχώρησης κλειδιών (HashBucket) και παρέχει πρόσβαση στις λειτουργίες που είχαν προδιαγραφεί. Περιέχει επίσης παραμέτρους που αφορούν σε χαρακτηριστικά του πίνακα (χωρητικότητα, προδιαγραφές, current κατάσταση).

Αναλυτικότερα για:

- **Εισαγωγή τυχαίου κλειδιού**
Η κλάση Linear Hashing είναι υπεύθυνη για το hash ενός HashBucket (index στον πίνακα) με βάση το κλειδί. Στην συνέχεια ο HashBucket ο οποίος επιλέχθηκε να δεχθεί το κλειδί, αναλαμβάνει να εισάγει το κλειδί εντός του. Ελέγχεται αν το κλειδί υπάρχει ήδη, διατρέχοντας τα ήδη υπάρχοντα. Σε περίπτωση που βρεθεί ότι η εισαγωγή πρόκειται να είναι διπλότυπη το κλειδί αγνοείται και δεν γίνεται καμία πράξη. Το κλειδί εισάγεται στην πρώτη επόμενη διαθέσιμη θέση του κεφαλικού Bucket ή σε κάποιου OverflowBucket (υπάρχοντα ή μη), με αναδρομικό τρόπο.

- **Αναζήτηση τυχαίου κλειδιού**

Η κλάση Linear Hashing φροντίζει να παράξει ένα index που αντιστοιχεί σε Bucket με βάση το κλειδί που αναζητείται. Το Bucket στη συνέχεια αφού αναζητήσει όλα τα κλειδιά το κεφαλικού πίνακα και δεν βρει αποτέλεσμα (αν βρει η μέθοδος επιστρέφει) συνεχίζει αναδρομικά με την ίδια λογική στους overflow buckets, εφόσον υπάρχουν.

- **Διαγραφή τυχαίου κλειδιού**

Η κλάση Linear Hashing φροντίζει να παράξει ένα index που αντιστοιχεί σε Bucket με βάση το κλειδί που αναζητείται προς διαγραφή. Ο πίνακας των κλειδιών εντός του Bucket διατρέχεται και εφόσον το επιθυμητό κλειδί βρεθεί ο πίνακας συρρικνώνεται κατά 1 κλειδί και η κλάση του LinearHashing ενημερώνεται ως προς τις παραμέτρους χωρητικότητας και κατάστασης. Η LH αναλαμβάνει να κάνει merge τα κατάλληλα Buckets σε περίπτωση που απαιτείται με βάση το κριτήριο. Αν το κλειδί δεν βρεθεί στο κεφαλικό Bucket τότε η αναζήτηση προς διαγραφή συνεχίζει στα overflow buckets αναδρομικά εφόσον υπάρχουν.

3. Πηγές κώδικα & αλγορίθμων

<http://courses.ece.tuc.gr/>

<https://www.geeksforgeeks.org/>