

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΡΧΕΙΩΝ

Ραφαήλ Τσιριβάκος 2013030199
LAB20243880

3η άσκηση
Γραμμικός Κατακερματισμός (Linear Hashing)

Κατασκεύασα την κλάση «Γραμμικός Κατακερματισμός» (Linear Hashing) στην κεντρική μνήμη. Η κλάση υποστηρίζει τις πράξεις εισαγωγής τυχαίου κλειδιού, αναζήτησης τυχαίου κλειδιού και διαγραφής τυχαίου κλειδιού. Ο πίνακας κατακερματισμού αρχικά έχει μέγεθος $M=100$ θέσεων και η κάθε θέση (Bucket) έχει χωρητικότητα 10 κλειδιά (SLOTS). Η χωρητικότητα των σελίδων υπερχειλίσσης είναι 10 κλειδιά. Το κριτήριο διάσπασης σελίδας (page split) σε πράξεις εισαγωγής στοιχείων εξαρτάται από την τιμή του «συντελεστή πληρότητας» u και είναι $u > 50\%$ ή $u > 80\%$. Το κριτήριο συγχώνευσης σελίδων (page merging) σε πράξεις διαγραφής ορίζεται $u < 50\%$.

Η πράξη της εισαγωγής

Για την πράξη της εισαγωγής χρησιμοποιώ αρχικά τη μέθοδο `addNum` στην με την οποία βάζω στον πίνακα `myArray` τις τιμές του αρχείου, ανά 100. Στη συνέχεια με την μέθοδο `put` βάζω ένα ένα τα στοιχεία αυτά από τον πίνακα στον πίνακα κατακερματισμού, η οποία παίρνει σαν είσοδο το κλειδί που θέλω να εισάγω. Εκεί υπολογίζεται η θέση που θα μπει μέσω των συναρτήσεων. Εφόσον, υπάρχει χώρος στην κατάλληλη θέση, γίνεται η εισαγωγή, διαφορετικά δημιουργείται σελίδα υπερχειλίσσης και μπαίνει εκεί. Στην συνέχεια γίνεται έλεγχος του συντελεστή πληρότητας και αν ικανοποιείται το κριτήριο που εφαρμόζεται, δηλαδή αν είναι μεγαλύτερος του 50% ή του 80%, γίνεται διάσπαση στην σελίδα που πρέπει να γίνει και μοιράζονται τα κλειδιά της. Η σελίδα που διασπάται είναι αρχικά η πρώτη και σειρά μετά έχει η επόμενη. Σε αυτή τη μέθοδο υπολογίζονται και οι παράμετροι των συναρτήσεων κατακερματισμού.

Με το τρέξιμο του προγράμματος δημιουργείται ένα δυαδικό δέντρο έρευνας και εισάγονται τα πρώτα 100 κλειδιά εκεί και στον πίνακα κατακερματισμού. Μετά πατώντας το κουμπί «@» γίνεται εισαγωγή των επόμενων 100. Ταυτόχρονα γίνεται η εισαγωγή των ίδιων κλειδιών στο δέντρο. Κάθε φορά εμφανίζεται ο πίνακας με όλα τα κλειδιά που έχουν εισαχθεί. Επίσης εμφανίζεται ο μέσος αριθμός συγκρίσεων ανά εισαγωγή κλειδιού στον πίνακα.

Με τη χρήση της μεθόδου `print` εμφανίζω τον πίνακα κατακερματισμού, δηλαδή ανά σειρά το κάθε bucket, μαζί με το overflow bucket, εφόσον έχει.

Η πράξη της αναζήτησης

Για την πράξη της αναζήτησης χρησιμοποιώ τη μέθοδο `search`. Εκεί υπολογίζω αρχικά τη σελίδα που πρέπει να βρίσκεται το κλειδί το στο οποίο γίνεται η αναζήτηση. Έπειτα ψάχνω στη συγκεκριμένη σελίδα και στην πιθανή σελίδα υπερχειλίσσης έως ότου το βρω. Τότε εμφανίζω κατάλληλο μήνυμα με τη θέση που βρίσκεται, διαφορετικά αν δεν υπάρχει εμφανίζω ανάλογο μήνυμα.

Χρησιμοποιώντας τη `Random` εξάγω έναν αριθμό από το διάστημα 0 έως 9999, ώστε να το χρησιμοποιήσω για να επιλέξω μία τυχαία θέση στον πίνακα `myArray`, όπου έχει όλα τα κλειδιά του αρχείου μου, και αναζητώ το συγκεκριμένο κλειδί. Αυτό το κάνω 50 φορές πατώντας την επιλογή «s» από το μενού ώστε να αναζητηθούν τελικά 50 τυχαία κλειδιά στον πίνακα κατακερματισμού. Ακριβώς τα ίδια κλειδιά αναζητούνται και στο δυαδικό δέντρο έρευνας με την αντίστοιχη μέθοδο. Μετά τα μηνύματα που εμφανίζει η μέθοδος `search`, εμφανίζεται ο μέσος αριθμός συγκρίσεων ανά αναζήτηση στον πίνακα κατακερματισμού και ο μέσος αριθμός συγκρίσεων ανά αναζήτηση στο δυαδικό δέντρο έρευνας.

Η πράξη της διαγραφής

Για την πράξη της διαγραφής γίνεται η χρήση της μεθόδου `delete`. Η μέθοδος αυτή έχει τη λειτουργία της `search` όπως την περιέγραψα παραπάνω. Αν υπάρχει το κλειδί προς διαγραφή μέσα στον πίνακα κατακερματισμού, το αφαιρώ και μετακινώ όλα τα επόμενα κλειδιά αριστερά. Στην περίπτωση που υπάρχει σελίδα υπερχειλίσης κάνω την ίδια διαδικασία και αν αδειάσει, την αφαιρώ. Στη συνέχεια ακολουθώ την αντίστροφη διαδικασία της εισαγωγής για το συντελεστή πληρότητας. Δηλαδή ελέγχω αν ικανοποιεί το κριτήριο συγχώνευσης, το οποίο ορίζει ο συντελεστής να είναι μικρότερος του 50%. Εφόσον ικανοποιείται και έχει γίνει τουλάχιστον μία διάσπαση, μεταφέρω τα κλειδιά της τελευταίας σελίδας στη σελίδα που διασπάστηκε τελευταία και τότε τη διαγράφω. Θέτω επίσης νέα τιμή στο δείκτη της σελίδας που είναι για διάσπαση και αν χρειαστεί προσαρμόζω κατάλληλα και τους συντελεστές των συναρτήσεων κατακερματισμού. Στην περίπτωση διαγραφής εμφανίζεται σε ποια θέση έγινε, διαφορετικά εμφανίζεται ανάλογο μήνυμα.

Η πράξη αυτή γίνεται για 50 τυχαία κλειδιά πατώντας την επιλογή «d» από το μενού, τα οποία υπολογίζονται με τον ίδιο τρόπο που υπολογίζονται για την πράξη της αναζήτησης. Μετά τα μηνύματα της μεθόδου εμφανίζεται ο πίνακας κατακερματισμού και ο μέσος όρος συγκρίσεων ανά διαγραφή κλειδιού.

Για όλες τις παραπάνω πράξεις χρησιμοποιώ τη μέθοδο `get` η οποία επιστρέφει τη σελίδα που χρειάζεται κάθε φορά.

Απόδοση Γραμμικού Κατακερματισμού

Για τον υπολογισμό του μέσου όρου συγκρίσεων της κάθε πράξης δημιούργησα δύο μεθόδους, τις `increaseCounter`, που η αυξάνει τον αριθμό των συγκρίσεων κατά τον αριθμό της εισόδου και επιστρέφει `true` και η άλλη καλεί την πρώτη με είσοδο 1. Τις δύο αυτές μεθόδους τις χρησιμοποίησα κάθε φορά που είχα ανάθεση τιμή ή σύγκριση στις μεθόδους `insert`, `search` και `delete`. Αντίστοιχες μεθόδους έφτιαξα για το δυαδικό δέντρο έρευνας οι οποίες καλούνται στην αντίστοιχη `search`.

Για να καταγράψω την απόδοση του γραμμικού κατακερματισμού έκανα εισαγωγή των πρώτων 100 κλειδιών, όπου υπολογίστηκε ο μέσος αριθμός συγκρίσεων ανά εισαγωγή. Μετά έκανα αναζήτηση σε αυτά πάνω σε 50 κλειδιά του αρχείου που επιλέγονται τυχαία ώστε να υπολογιστεί ο μέσος αριθμός συγκρίσεων ανά αναζήτηση. Και τέλος έκανα διαγραφή από τον πίνακα πάνω σε 50 κλειδιά του αρχείου που επιλέγονται τυχαία ώστε να υπολογιστεί ο μέσος αριθμός συγκρίσεων ανά διαγραφή. Αφού έχουν διαγραφεί κάποια κλειδιά, ο πίνακας δεν έχει όλα τα κλειδιά που θα έπρεπε και στην πορεία της διαδικασίας θα έχει μικρότερο μέγεθος απ' ό,τι προβλέπεται στις εισαγωγές. Οπότε ξαναέτρεξα το πρόγραμμα κάνοντας αυτή τη φορά εισαγωγή των πρώτων 100 κλειδιών και των επόμενων 100 και ξαναπήρα τις ίδιες μετρήσεις. Ακολούθησα την ίδια διαδικασία και κατέγραψα τις ίδιες μετρήσεις για εισαγωγή, αναζήτηση και διαγραφή μέχρις ότου να φτάσω τα 10000 κλειδιά. Το πείραμα το επανέλαβα δύο φορές, την πρώτη με το συντελεστή πληρότητα $u > 50\%$ και τη δεύτερη να είναι $u > 80\%$. Το κριτήριο συγχώνευσης σελίδων (`page merging`) σε πράξεις διαγραφής είναι $u < 50\%$ και στις δύο περιπτώσεις.

Με τις παραπάνω μετρήσεις κατασκεύασα το διάγραμμα X-Y όπου Y είναι ο μέσος αριθμός συγκρίσεων και X είναι το N, δηλαδή ο αριθμός των στοιχείων στον πίνακα κατακερματισμού. Πάνω στο ίδιο διάγραμμα σχεδίασα 3 διαφορετικές καμπύλες που δείχνουν την εξάρτηση του μέσου αριθμού συγκρίσεων ανά εισαγωγή/αναζήτηση/διαγραφή αντίστοιχα για $u > 50\%$ πράξεις εισαγωγής και αναζήτησης και $u < 50\%$ για πράξεις διαγραφής. Στο ίδιο διάγραμμα πρόσθεσα τρεις αντίστοιχες καμπύλες για $u > 80\%$ για περιπτώσεις εισαγωγής και αναζήτησης και $u < 50\%$ για πράξεις διαγραφής. Στο ίδιο διάγραμμα και για κάθε N προστέθηκε η καμπύλη από τις μετρήσεις στο Δυαδικό Δένδρο Έρευνας, που ανέφερα παραπάνω. Πήρα μια έτοιμη υλοποίηση από το Διαδίκτυο για τη δημιουργία του δέντρου, την εισαγωγή και την αναζήτηση στοιχείων σε αυτό.

Πηγές

Για τη διεκπεραίωση της εργασίας βασίστηκα στις διαλέξεις και τα φροντιστήρια του μαθήματος. Επίσης για την υλοποίηση του γραμμικού κατακερματισμού βασίστηκα σε κώδικα από τη σελίδα [github.com](https://www.geeksforgeeks.org/binary-search-tree-data-structure/). Τέλος, για την υλοποίηση του δυαδικού δέντρου έρευνας χρησιμοποίησα τον κώδικα από τη σελίδα <https://www.geeksforgeeks.org/binary-search-tree-data-structure/>