

## ΠΛΗΡΟΦΟΡΙΕΣ 3<sup>ΗΣ</sup> ΑΣΚΗΣΗΣ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: Μιχαήλ Κρατημένος

Αρχικά, διαβάστηκε το αρχείο με τους αριθμούς που μας δόθηκε, μέσω του ορίσματος της main και μετά άνοιξε μέσω της κλάσης RAFile. Ύστερα δημιουργήθηκε η κλάση LinearHashing ( που είναι ο γραμμικός πίνακας κατακερματισμού που ζητήθηκε να υλοποιηθεί ), καθώς και η κλάση HashBucket, η οποία είναι οι κουβάδες της LinearHashing, αλλά και οι μέθοδοι που ζητήθηκαν ( εισαγωγή, αναζήτηση, διαγραφή ). Τέλος, δημιουργήθηκε και η κλάση BSTDynamic, που είναι μια δυναμική υλοποίηση δυαδικού δέντρου έρευνας με τις μεθόδους που ζητήθηκαν.

Στην LinearMain δημιουργήθηκαν 2 στιγμιότυπα της LinearHashing ( ένα με  $u > 50\%$  και ένα με  $u > 80\%$  ) με τα δεδομένα αρχικοποίησης που μας δόθηκαν, αλλά και ένα στιγμιότυπο της BSTDynamic, αλλά και κάποιες λίστες που κρατάνε τους αριθμούς που διαθέτει η κάθε δομή δεδομένων. Ύστερα καλείται ένα μικρό μενού επιλογών ( τυπώνεται μέσω της printMenu1 ) που ρωτά το χρήστη ( μέσω ενός στιγμιότυπου της StandardInputRead (reader) ) αν θέλει να τρέξει τα πειράματα που μας ζητήθηκαν ή να τερματίσει το πρόγραμμα κλείνοντας και το αρχείο.

Αν επιλέξει τα πειράματα τότε τυπώνεται ο πίνακας με τις μέσες συγκρίσεις ανά εισαγωγή, αναζήτηση και διαγραφή αντίστοιχα σε κάθε δομή δεδομένων. Αυτό επιτυγχάνεται με τις μεθόδους readNumbersHT, readNumbersBST, searchNumbersHT, searchNumbersBST, deleteNumbersHT, deleteNumbersBST, αλλά και μέσω του for loop που τα τυπώνει.

### ΜΕΘΟΔΟΙ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ ΑΛΛΑ ΔΕΝ ΑΝΑΦΕΡΘΗΚΑΝ

Στην κλάση LinearHashing:

- hashFunction: Επιστρέφει τη θέση του κουβά του κλειδιού που δίνεται σαν όρισμα.
- loadFactor: Υπολογίζει το συντελεστή πληρότητας ( στον παρονομαστή δεν υπολογίζονται τα overflow buckets ).
- bucketSplit: Σπάει τον κουβά που δείχνει ο δείκτης (nextSplit) (χρησιμοποιώντας και την αντίστοιχη μέθοδο (splitBucket) από την κλάση HashBucket).
- bucketMerge: Ενώνει τον κουβά που δείχνει ο δείκτης (nextSplit) με τον τελευταίο κουβά (χρησιμοποιώντας και την αντίστοιχη μέθοδο (mergeBucket) από την κλάση HashBucket).

- insertKey, deleteKey, searchKey: Εισάγει, διαγράφει και αναζητά ένα κλειδί αντίστοιχα ( χρησιμοποιώντας και τις αντίστοιχες μεθόδους από την κλάση HashBucket ).

Στην κλάση HashBucket:

- removeLastKey: Διαγράφει το τελευταίο στοιχείο του.

Στην κλάση BSTDynamic:

- insert, insertRec, deleteKey, deleteRec, search, searchRec: Εισάγει, διαγράφει και αναζητά ένα κλειδί αντίστοιχα.
- minValue: Επιστρέφει το μικρότερο κλειδί από τη δοσμένη ρίζα.
- finRoot: Επιστρέφει τον κόμβο του δοσμένου κλειδιού.

### **ΠΑΡΑΠΑΝΩ ΜΕΘΟΔΟΙ - ΛΑΘΗ :**

- Έχουν υλοποιηθεί μέθοδοι εκτύπωσης των δομών ( printHash, printBucket, print, printRec).

### **ΠΗΓΕΣ**

- LinearHashing, HashBucket: πήρα έναν έτοιμο κώδικα από τις περσινές σημειώσεις φροντιστηρίου του μαθήματος.
- BSTDynamic: πήρα τον κώδικα από την προηγούμενη άσκηση.
- deleteKey, deleteRec, minValue:  
<https://www.geeksforgeeks.org/binary-search-tree-set-2-delete/?ref=lbp>
- StandardInputRead: αυτή η κλάση μας δόθηκε στο 2ο εξάμηνο από τους εργαστηριακούς βοηθούς στο μάθημα Δομημένος Προγραμματισμός.