

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

3^η άσκηση

Ημερομηνία παράδοσης: 22 Μαΐου 2020

Η άσκηση είναι ατομική

Γραμμικός Κατακερματισμός (Linear Hashing) (4 μονάδες)

Κατασκευάστε την κλάση «Γραμμικός Κατακερματισμός» (Linear Hashing) στην **κεντρική μνήμη**. Η κλάση υποστηρίζει τις πράξεις:

- Εισαγωγή τυχαίου κλειδιού
- Αναζήτηση τυχαίου κλειδιού
- Διαγραφή τυχαίου κλειδιού

Ο πίνακας κατακερματισμού αρχικά έχει μέγεθος $M=100$ θέσεων και η κάθε θέση έχει χωρητικότητα **10** κλειδιά. Η χωρητικότητα των σελίδων υπερχείλισης είναι 10 κλειδιά. Το **κριτήριο διάσπασης σελίδας** (page split) πράξεις εισαγωγής στοιχείων ορίζεται στην αρχή της λειτουργίας της μεθόδου και εξαρτάται από την τιμή του «συντελεστή πληρότητας» u και μπορεί να είναι $u > 50\%$ ή $u > 80\%$. Το **κριτήριο συγχώνευσης σελίδων** (page merging) σε πράξεις διαγραφής ορίζεται $u < 50\%$.

Απόδοση Γραμμικού Κατακερματισμού (4 μονάδες)

Θα εξεταστεί η απόδοση της μεθόδου για διαφορετικές τιμές του αριθμού κλειδιών N . Σας δίνεται ένα αρχείο με 10^4 κλειδιά τα οποία θα χρησιμοποιήσετε για να κατασκευάσετε τον πίνακα κατακερματισμού. Ο πίνακας κατακερματισμού μεγαλώνει ανά 100 κλειδιά (σταδιακά): αρχικά κάντε $N=100$ εισαγωγές (με τα πρώτα 100 κλειδιά) και πάρτε τις παρακάτω μετρήσεις. Συνεχίστε με τις επόμενες 100 εισαγωγές κάθε φορά και ξαναπάρτε μετρήσεις. Συνεχίστε το ίδιο μέχρι $N=10000$. Για κάθε $N = 100, 200, 300, \dots, 10000$ μετρήστε:

1. Τον μέσο αριθμό συγκρίσεων ανά εισαγωγή κλειδιού.
2. Τον μέσο αριθμό συγκρίσεων ανά αναζήτηση πάνω σε 50 κλειδιά που επιλέγονται τυχαία από τα παραπάνω κλειδιά.
3. Τον μέσο αριθμό συγκρίσεων ανά διαγραφή πάνω σε 50 κλειδιά που επιλέγονται τυχαία από τα παραπάνω κλειδιά (στην πορεία των μετρήσεων και αφού έχουν διαγραφεί κάποια κλειδιά, ο πίνακας έχει μικρότερο μέγεθος απ' ότι προβλέπεται στις εισαγωγές).

Κατασκευάστε το διάγραμμα $X-Y$ όπου Y είναι ο μέσος αριθμός συγκρίσεων και X είναι το N . Πάνω στο ίδιο διάγραμμα σχεδιάστε 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. Επαναλάβετε το ίδιο πείραμα (τρεις επιπλέον καμπύλες) για $u > 80\%$ για περιπτώσεις εισαγωγής και αναζήτησης και $u < 50\%$ για πράξεις διαγραφής. **Προσέξτε ότι όταν το κριτήριο διάσπασης είναι $u > 80\%$ ενδέχεται να μην προκληθούν συγχωνεύσεις σελίδων κατά την διαγραφή.**

Στο ίδιο διάγραμμα και για κάθε N μετρήστε το μέσο αριθμό συγκρίσεων ανά αναζήτηση (πάνω στα ίδια 50 κλειδιά που χρησιμοποιήσατε παραπάνω) σε ένα Δυαδικό Δένδρο Έρευνας (πάρτε μια έτοιμη υλοποίηση από το Διαδίκτυο).

Κατασκευάστε το ίδιο διάγραμμα για τις δύο περιπτώσεις του κριτηρίου διάσπασης δηλαδή για $u > 50\%$ και $u > 80\%$ για πράξεις εισαγωγής και $u < 50\%$ για πράξεις διαγραφής.

Τεκμηρίωση των Αποτελεσμάτων (2 μονάδες)

Σχολιάστε (ξεχωριστό αρχείο κειμένου) την απόδοση όλων των μεθόδων και προσπαθήστε να δικαιολογήσετε την απόδοση κάθε μεθόδου. Προτείνετε πιθανές βελτιώσεις (δεν ζητείται να τις υλοποιήσετε). **Δώστε ιδιαίτερη βαρύτητα στην ερμηνεία των αποτελεσμάτων και μην δείξετε μόνο τις τιμές.**

Συστάσεις:

- Μην προχωρήσετε στην υλοποίηση πριν καταλάβετε καλά την μέθοδο Γραμμικού Κατακερματισμού.
- Υπάρχουν πολλές υλοποιήσεις στο Διαδίκτυο που **έχουν τον παραπλανητικό τίτλο «Γραμμικός Κατακερματισμός» (Linear Hashing) αλλά δεν είναι η μέθοδος που διδάχτηκε στο μάθημα. Οποιαδήποτε τέτοια μέθοδος δεν θα βαθμολογηθεί.**
- Η μέθοδος «Γραμμικός Κατακερματισμός» (Linear Hashing) υπάρχει
 - ο Στις σημειώσεις του μαθήματος (σελ. 34)
 - ο https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_742
 - ο <http://delab.csd.auth.gr/papers/LinearHashing2017.pdf>
 - ο Η μέθοδος έχει προταθεί αρχικά από τον Witold Litwin το έτος 1980 στο <http://infolab.cs.unipi.gr/pre-eclass/courses/db/db-post/readings/Litwin-VLDB80.pdf>
- Υπάρχουν κώδικες στο Διαδίκτυο και στον ιστοχώρο μαθήματος. Η άσκηση είναι διατυπωμένη με τέτοιο τρόπο ώστε θα χρειαστεί να καταλάβετε και να αλλάξετε τους κώδικες προκειμένου να πάρετε τα αποτελέσματα που ζητούνται. **Διαφορετικά υπάρχει κίνδυνος η άσκηση να μην υλοποιεί αυτό που ζητείται και να είναι απλή αντιγραφή κώδικα. Σε αυτή την περίπτωση η άσκηση δεν βαθμολογείται.**
- Το πρόγραμμά σας πρέπει να δουλέψει για την είσοδο κλειδιών σε αρχείο που θα δώσουμε εμείς κατά την διόρθωση της άσκησης ή για τυχαία τιμή κλειδιού.
- Το πρόγραμμά σας πρέπει να τυπώνει στην οθόνη τις τιμές ανά πράξη (αναζήτησης, εισαγωγής ή διαγραφής) για οποιοδήποτε αρχείο κλειδιών που θα δίνεται στην είσοδο ή για οποιαδήποτε τυχαία τιμή κλειδιού που δίνεται στην είσοδο.
- Ενδεικτική έξοδος εκτέλεσης της εφαρμογής σας (δείτε το υλικό από το 1^ο φροντιστήριο για το πως παράγετε στοιχισμένο αποτέλεσμα):

```
java mypackage.MyClass path_to_input_file.bin
```

Input size (N)	LH u>50% avg # comparisons per insertion	LH u>50% avg # comparisons per search	LH u>50% avg # comparisons per deletion	LH u>80% avg # comparisons per insertion	LH u>80% avg # comparisons per search	LH u>80% avg # comparisons per deletion	BST avg # comparisons per insertion	BST avg # comparisons per search	BST avg # comparisons per deletion
100	5.8	1.1
200	3.1	1.2
300	1.2	1.3
.....									
10000	4.5	1.4

Παραδοτέα:

Ένα συμπίεσμένο zip αρχείο που περιέχει:

- Μία έκθεση 2-3 σελίδες με τα αποτελέσματα που σας ζητούνται δηλαδή, έναν πίνακα για $N=10^4$ και το διάγραμμα X-Y που δείχνει πως εξαρτάται ο χρόνος εκτέλεσης από τον N. **Δώστε ιδιαίτερη βαρύτητα στην ερμηνεία των αποτελεσμάτων και μην δείξετε μόνο τις τιμές. Ο βαθμός του 3^{ου} μέρους (2 μονάδες) εξαρτάται από την ερμηνεία.**
- Μία έκθεση που περιγράφει σε 1-2 σελίδες πώς φτιάχτηκε ο κώδικας (δηλ. για κάθε ερώτημα ποια είναι η γενική ιδέα της λύσης σε 3-4 προτάσεις), υπάρχουν σαφείς οδηγίες μετάφρασης από compiler και εκτέλεσης, τι λάθη έχει (αν έχει, περιπτώσεις που δεν δουλεύει το πρόγραμμα, ή περιπτώσεις που κάνει περισσότερα από όσα σας ζητεί η άσκηση, τι χρησιμοποιήσατε από έτοιμα προγράμματα ή πηγές πληροφόρησης. Υποδείξετε ακόμα και πηγές στο WWW όπως Wikipedia ή Stackoverflow (πλήρης διεύθυνση σχετικών σελίδων).
- Ο κώδικας περιέχει συνοπτικά σχόλια που εξηγούν την υλοποίηση. Προσθέστε σχόλια σε μορφή javadoc στην αρχή της κάθε κλάσης και κάθε μεθόδου. Επίσης javadoc σχόλια πριν από κάθε member variable των κλάσεων. Και όπου απαιτείται μέσα στον κώδικά σας.
- Εκτός των παραπάνω, οι ασκήσεις βαθμολογούνται με άριστα εφόσον:
 - ο Το zip είναι πλήρες.
 - ο Οι κώδικες περνούν από compiler και εκτελούνται κανονικά και σωστά σε windows ή Linux περιβάλλον (Προσοχή: Θα πρέπει να κάνετε χρήση σχετικών directory paths και όχι απόλυτων, τα οποία ισχύουν μόνο για τους υπολογιστές σας).
 - ο Ο κώδικάς σας δουλεύει για οποιαδήποτε τιμή παραμέτρων.
- Οι ασκήσεις υποβάλλονται ηλεκτρονικά στον ιστοχώρο του μαθήματος και όχι με e-mail.
- Οι αντιγραφές (ακόμη και μέρους της υλοποίησης) μηδενίζονται.