

ΠΛΗ202 – ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ & ΑΡΧΕΙΩΝ

3η ΑΣΚΗΣΗ

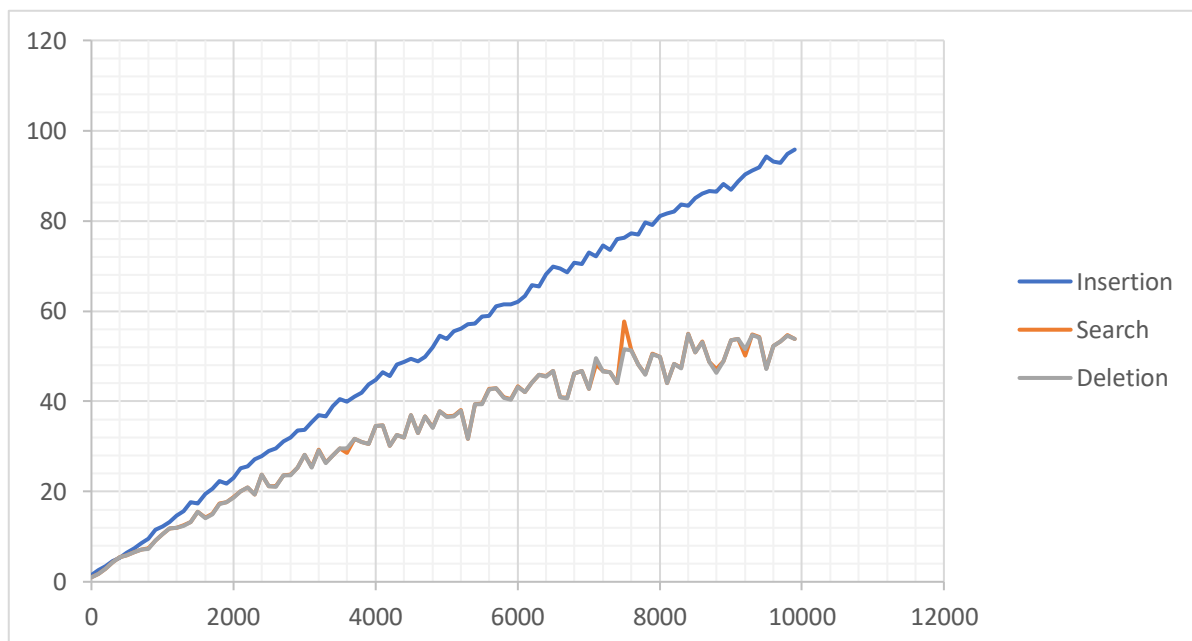
ΠΕΤΡΟΥ ΔΗΜΗΤΡΗΣ - 2018030070

ΑΝΑΦΟΡΑ ΚΑΤΑΓΡΑΦΗΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

- Οι δοκιμές πραγματοποιήθηκαν χρησιμοποιώντας το αρχείο “testnumbers_1000000_BE.bin” που δόθηκε μεταξύ άλλων, με την εκφώνηση της άσκησης.
- Χρησιμοποιήθηκε γεννήτρια τυχαίων αριθμών προκειμένου να επιλεγθούν στοιχεία μέσα από το αρχείο ως arguments πράξεων.

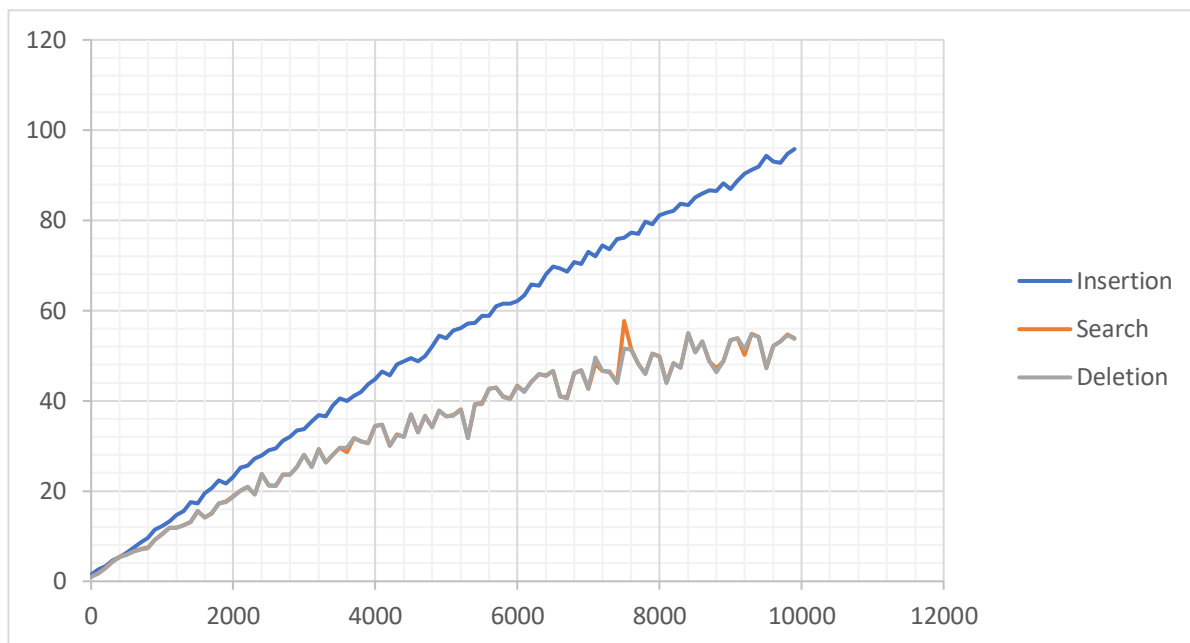
Κατά τις δοκιμές μετρήθηκε ο αριθμός συγκρίσεων για τις εισαγωγές, αναζητήσεις και διαγραφές στοιχείων σε Hash Table (με διαφορετικές παραμέτρους) και σε δυαδικό δέντρο που υλοποιήθηκαν στον κώδικα.

Στο παρακάτω διάγραμμα απεικονίζεται ο αριθμός των συγκρίσεων σε πράξεις εισαγωγής, αναζήτησης και διαγραφής σε πίνακα γραμμικού κατακερματισμού με κριτήριο διάσπασης $u > 50\%$.



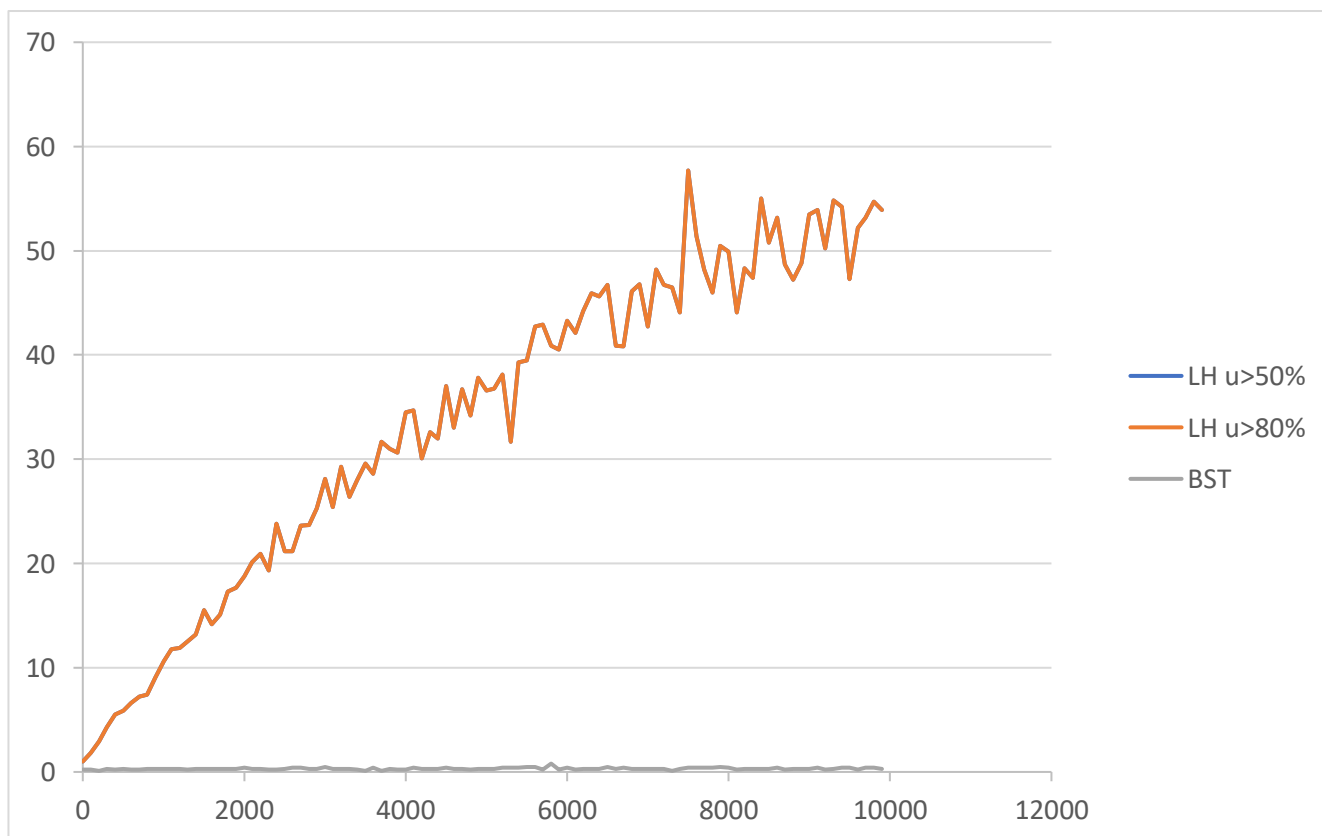
Όπως παρατηρείται, το πλήθος των συγκρίσεων μεταξύ στοιχείων του πίνακα και παραμέτρων των μεθόδων αυξάνεται ανάλογα με το πλήθος των στοιχείων που υπάρχουν στον πίνακα. Η πολυπλοκότητα του Hash Table επιβεβαιώνεται. Η πράξη της διαγραφής και της αναζήτησης κινούνται περίπου στα ίδια επίπεδα εφόσον πραγματοποιήθηκαν με ίδια arguments και η πολυπλοκότητα τους δεν διαφέρει κατά μεγάλο βαθμό.

Στο παρακάτω διάγραμμα απεικονίζεται ο αριθμός των συγκρίσεων σε πράξεις εισαγωγής, αναζήτησης και διαγραφής σε πίνακα γραμμικού κατακερματισμού με κριτήριο διάσπασης $u > 80\%$.



Η απόκριση του πίνακα κατακερματισμού με κριτήριο διάσπασης $u > 80\%$ δείχνει να μην έχει απολύτως καμία διαφορά από αυτή με $u > 50\%$. Αυτό συμβαίνει διότι η πράξη σύγκρισης και εν τέλει η πολυπλοκότητα μιας μεθόδου δεν αφορά κατά κύριο λόγο τον τρόπο με τον οποίο κατανέμονται στοιχεία σε μια δομή δεδομένων, παρά το ακριβές πλήθος τους. Προς βελτίωση των μεθόδων θα μπορούσε να αποφευχθεί η αναδρομή και να επιλεγεί μια απλή iterative μορφή.

Οι πράξεις σύγκρισης για αναζήτηση 50 τυχαίων επιλεγμένων κλειδιών στον Hash Table και σε ένα δυναμικό δυαδικό δέντρο αναζήτησης φαίνονται διαγραμματικά παρακάτω:



Όπως αναμενόταν η διαφορά είναι έντονη. Ο γραμμικός κατακερματισμός σε σχέση με το δέντρο παρουσιάζει αύξηση πολυπλοκότητας που θυμίζει μονοδιάστατο πεδίο & δέντρο. Το BST, αν και σε σχετικά μικρά N , δείχνει να διατηρεί την δυαδικότητα του σταθερή. Την εξομάλυνση αυτή θα περίμενε κανείς να τη παρατηρήσει σε πολύ μεγάλα πλήθη στοιχείων. Το BST αποδεικνύεται ότι για άλλη μια φορά κυριαρχεί ως προς την πολυπλοκότητα, χωρίς ωστόσο αυτό να το κάνει την κατάλληλη δομή δεδομένων για όλες τις περιπτώσεις εφαρμογής.

Η πράξη της αναζήτησης στο LH έχει αρκετά περιθώρια βελτίωσης. Με πρακτικότερο κώδικα ή άλλες μεθόδους διαχείρισης overflow (Open Addressing) υπάρχει πιθανότητα οι συγκρίσεις να μειώνονταν και το πρόγραμμα να γινόταν πιο ευέλικτο ως προς τη διαθεσιμότητα Buckets.