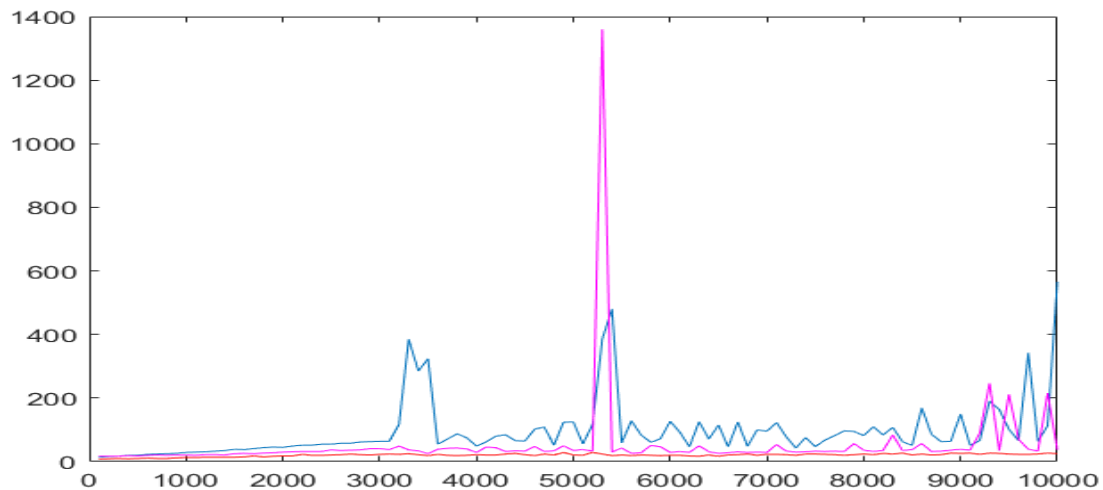


ΜΕΤΡΗΣΕΙΣ 3ης ΑΣΚΗΣΗΣ

Τιμές για N=10000:

Μέθοδος	Average comparisons for 50 insertions.	Average comparisons for 50 searches.	Average comparisons for 50 deletions.
Linear Hashing $u > 80\%$	566	25	37
Linear Hashing $u > 50\%$	790	12	3474
Binary Search Tree(Dynamic Implementation)	53	48	62

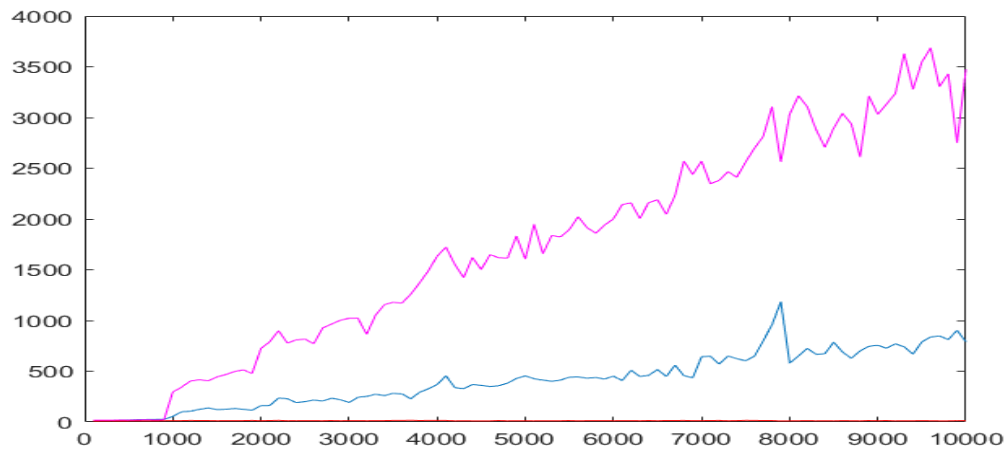
Διάγραμμα X-Y για τις εισαγωγές του linear hashing με $u > 80\%$:



μπλε – avg insertions, κόκκινο – avg searches, ματζέντα – deletions

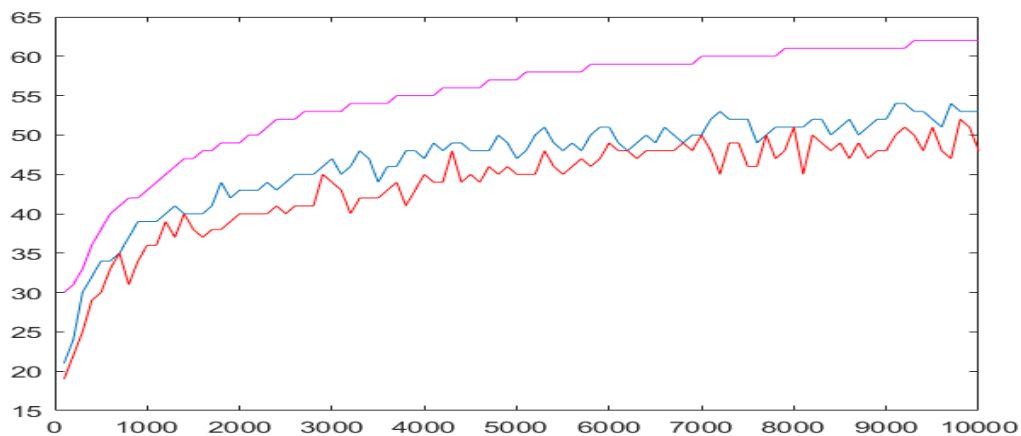
Από το διάγραμμα παρατηρείται πως το linear hashing με $u > 80\%$ συμπεριφέρεται όπως είναι αναμενόμενο από τη θεωρία, αφού οι εισαγωγές είναι σχετικά σταθερές, εκτός από κάποιες περιπτώσεις που γίνονται splits στον πίνακα, με αποτέλεσμα να αυξάνονται και οι πράξεις. Οι αναζητήσεις όπως είναι εμφανές, είναι σταθερές και συνάδει με την θεωρία, αφού ο αλγόριθμος έχει πολυπλοκότητα $O(1)$. Οι διαγραφές είναι, ως επί τον πλείστον, σταθερές, αν εξαιρέσουμε κάποιες ξαφνικές αυξήσεις που συμβαίνουν λόγω των merges. Από τα παραπάνω, μπορούμε να καταλήξουμε στο ότι με την επιλογή κλειδιού $u > 80\%$ έχουμε μια αρκετά γρήγορη linear hashing δομή.

Διάγραμμα για linear hashing με $u > 50\%$:

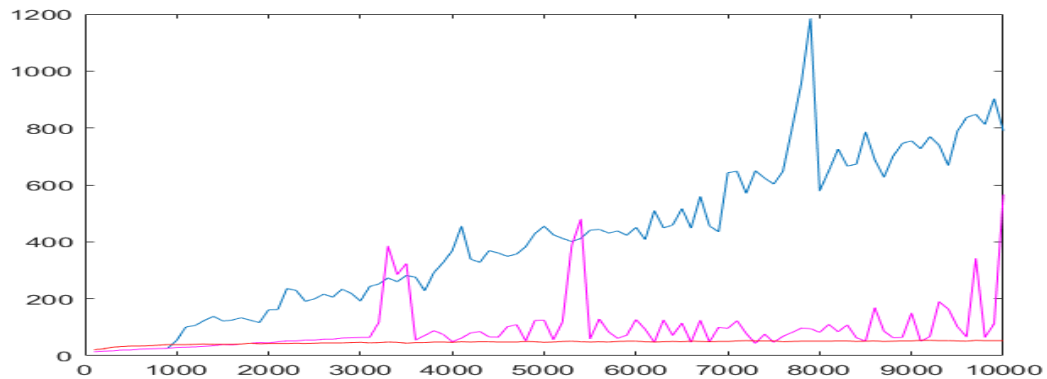


μπλε – avg insertions, κόκκινο – avg searches, ματζέντα – deletions

Στο διάγραμμα παρατηρείται ότι η linear hashing με $u > 50\%$, δεν είναι εξίσου αποδοτική με τη linear hashing με $u > 80\%$, επειδή λόγω των μικρών ποσοστών ($u > 50\%$ για sort και $u < 50\%$ για merge), γίνονται πολλές διασπάσεις σε κάθε εισαγωγή με αποτέλεσμα η δομή να μην είναι τόσο αποδοτική.



μπλε – avg insertions, κόκκινο – avg searches, ματζέντα – deletions



μπλε – avg insertions, κόκκινο – avg searches, ματζέντα – deletions

Η αναζήτηση στο Δυαδικό Δέντρο Αναζήτησης έχει την προβλεπόμενη συμπεριφορά, αφού η γραφική της είναι η λογαριθμική συνάρτηση. Παρόλο που το δυαδικό δένδρο έχει πιο γρήγορη εισαγωγή, η διαδικασία εύρεσης και διαγραφής είναι πιο αργή(στην κανονική λειτουργία της κάθε δομής).