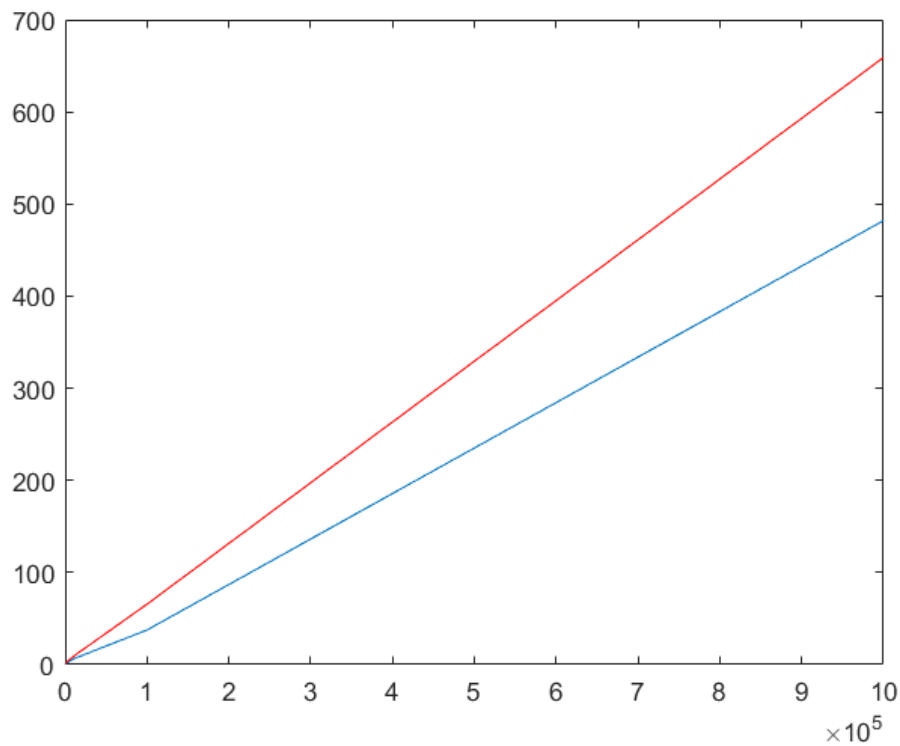


ΜΕΤΡΗΣΕΙΣ 2ης ΑΣΚΗΣΗΣ

Τιμές για N=1000000.

Μέθοδος	Μέσος αριθμός συγκρίσεων/ εισαγωγή	Συνολικός χρόνος για N εισαγωγές	Μέσος αριθμός συγκρίσεων/ τυχαία εισαγωγή	Συνολικός Χρόνος για 100 αναζητήσεις	Μέσος αριθμός συγκρίσεων/ αναζήτηση εύρους (K=100)	Μέσος αριθμός συγκρίσεων/ αναζήτηση εύρους (K=1000)
ΔΔΕ με δυναμική παραχώρηση μνήμης	15 συγκρίσεις	659ms	16 συγκρίσεις	0.645ms	126 συγκρίσεις	128 συγκρίσεις
ΔΔΕ με array	15 συγκρίσεις	482ms	13 συγκρίσεις	0.425ms	121 συγκρίσεις	123 συγκρίσεις
Ταξινομημέν ο πεδίο			29 συγκρίσεις	0.321ms	80 συγκρίσεις	81 συγκρίσεις

Διάγραμμα X-Y για τις εισαγωγές.



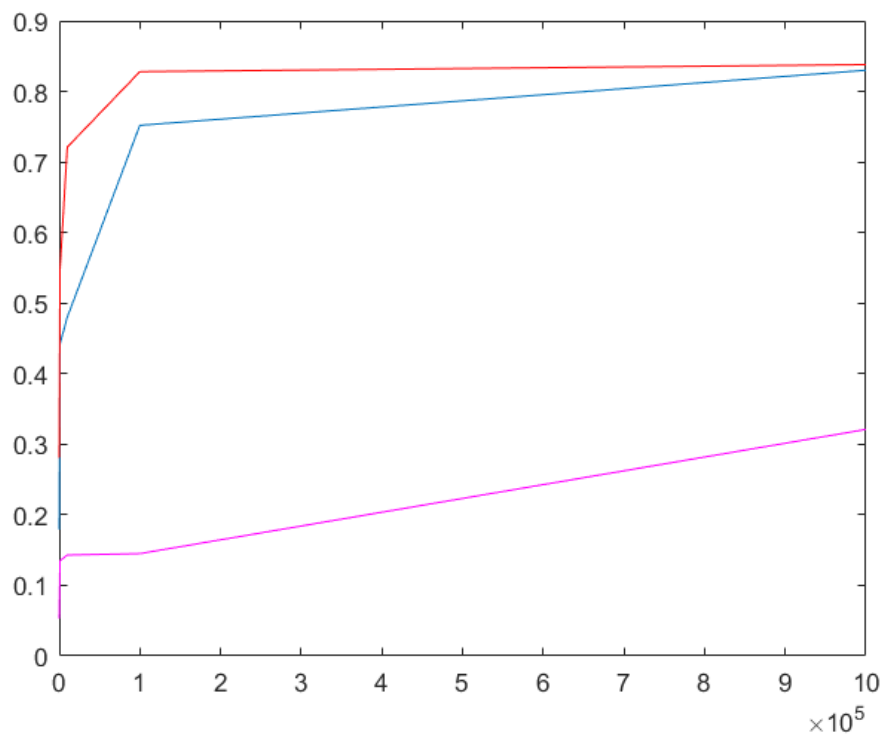
red=Dynamic Implementation, Blue= Array Implementation

Το παραπάνω διάγραμμα αποκλίνει από την προβλεπόμενη μέση κατάσταση που έχει πολυπλοκότητα $O(\log n)$. Αυτό μπορεί να συμβαίνει για τους παρακάτω λόγους:

- Το δέντρο μπορεί να μην είναι ισορροπημένο, με αποτέλεσμα να η συνάρτηση του χρόνου εισαγωγής να αποκλίνει από την $O(\log n)$.
- Κάθε εκτέλεση του κώδικα διαφέρει σε runtime από τις άλλες(αφού ο υπολογιστής μας κάνει πολλά πράγματα ταυτόχρονα), με αποτέλεσμα οι τιμές που επιλέχτηκαν να μην είναι σχετικές η μία με την άλλη(επειδή δεν έγινε ένας μέσος όρος τιμών για κάποιο αριθμό εκτελέσεων δεν θα έχουμε και πολύ ακριβές σχεδιάγραμμα).

Για την υλοποίηση και τον δύο μεθόδων χρησιμοποιήθηκε η αναδρομική μέθοδος της εισαγωγής και είναι αρκετά πιθανό, λόγω του συμβολομεταφραστή της java, να μην έχει την ίδια ταχύτητα με τον αντίστοιχο κώδικα με επανάληψη, αλλά παρόλα αυτά δεν εξηγείται η γραμμικότητα.

Διάγραμμα X-Y για τις αναζητήσεις.



Red= Dynamic Implementation, Blue=Array Implementation, Magenta= Sorted Array Implementation.

Η αναζήτηση στο Δυαδικό Δέντρο Αναζήτησης της δυναμικής υλοποίησης και της υλοποίησης με πίνακα έχουν την προβλεπόμενη συμπεριφορά, αφού η γραφική τους είναι η λογαριθμική συνάρτηση και ο ταξινομημένος πίνακας είναι αρκετά πιο γρήγορος λόγω της απλότητας του. Η υλοποίηση με πίνακα φαίνεται να είναι πιο γρήγορη από τη δυναμική, παρόλο που ο αλγόριθμος και στις δύο υλοποιήσεις είναι παρόμοιος.

Ο αριθμός των μέσων συγκρίσεων για κάθε μέθοδο είναι παρόμοιος για το ΔΔΕ με δυναμική υλοποίηση και με πίνακα, αφού και οι δύο υλοποιήσεις χρησιμοποιούν τις ίδιες μεθόδους. Διαφέρουν όμως χρονικά, αφού η πρόσβαση σε πίνακα είναι πιο γρήγορη από πρόσβαση σε instance ενός αντικειμένου. Το ταξινομημένο πεδίο είναι πιο γρήγορο, παρόλο που κάνει περισσότερες μέσες συγκρίσεις, επειδή έχει υλοποιηθεί με επαναλήψεις.

Οι πιθανές βελτιώσεις που μπορούν να γίνουν είναι γενικά η δημιουργία ενός μενού για την καλύτερη επαφή του χρήστη με το πρόγραμμα και βελτίωση της BinarySearchRange που έχει υλοποιηθεί βρίσκοντας τον μικρότερο και τον μεγαλύτερο αριθμό μέσα στον πίνακα και μετά τυπώνει τα ενδιάμεσα, ενώ μπορεί να βρει το μικρότερο και μετά γραμμικά να τυπώσει τους όλους τους αριθμούς που είναι μέσα στο όριο.(Ο πίνακας είναι ταξινομημένος).

Οι τιμές που χρησιμοποιήθηκαν για τη γραφική(Έγιναν σε matlab):

Arrayinsert:0.108 1.062 6.230 37.555

DynamicInsert:1 2.013 8.910 65.565

ArraySearch:0.179 0.442 0.480 0.550

Dynamicsearch:0.281 0.548 0.721 0.658

SortedArray: 0.053 0.135 0.143 0.145 0.321