

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

PROJECT(2017)

!Αναφορά!

Κωνσταντίνος Αδαμόπουλος

6270

adamopoulos@ceid.upatras.gr

Python

ΕΙΣΑΓΩΓΗ

Το πρόγραμμα τρέχει μόνο σε **python3**. Αν το εκτελέσετε σε linux τότε μπορείτε να χρησιμοποιήσετε το bash script που υπάρχει στο φάκελο Launcher. Το script αυτό εκτελεί την εντολή εκτέλεσης του προγράμματος `python3 Main.py <PathToFile>` και επιπλέον με τα κατάλληλα argument values μπορείτε να ορίσετε αν θα πάρει το default .csv file από τον φάκελο CSVFiles όπου εκεί στεγάζονται όλα τα csv files, αν θα πάρει από τον ίδιο φάκελο άλλο .csv file ή αν θα πάρει ένα άλλο αρχείο από ένα άλλο path. Για περισσότερες πληροφορίες απλώς πληκτρολογήστε την εντολή `./launcher.sh -h` στο τερματικό σας.

Αν τρέχετε σε windows και θέλετε να εκτελεστεί το πρόγραμμα με το default αρχείο που βρίσκεται στον φάκελο CSVFiles τότε απλώς πηγαίνετε στον φάκελο Source και κάντε double-click το αρχείο Main.py. Αν θέλετε να τρέξετε άλλο αρχείο από τον φάκελο CSVFiles ή από άλλο path καλό είναι να το κάνετε μέσω cmd με την εντολή **python3 Main.py <PathToFile>**.

Συναρτήσεις

1.Load():

Στην συνάρτηση Load() διαβάζω το csv file και εκχωρώ τα ids των ξενοδοχείων σαν κλειδιά σε ένα dictionary και την υπολοιπή πληροφορία σαν το περιεχόμενο του συγκεκριμένου κλειδιού του dictionary.

2.Add():

Στην συνάρτηση Add() προσθέτω καινούργια ξενοδοχεία μαζί με τις πληροφορίες τους καθώς και τις κρατήσεις τους και τα αποθηκεύω σε μια λίστα.

3.Save():

Στην συνάρτηση Save() αποθηκεύω τα δεδομένα της λίστας από την Add().

4.Search by id:

Αναζητω με γραμμικη αναζητηση,binary search,interpolation search τα ξενοδοχεια με βαση το id.

5.Search by surname:

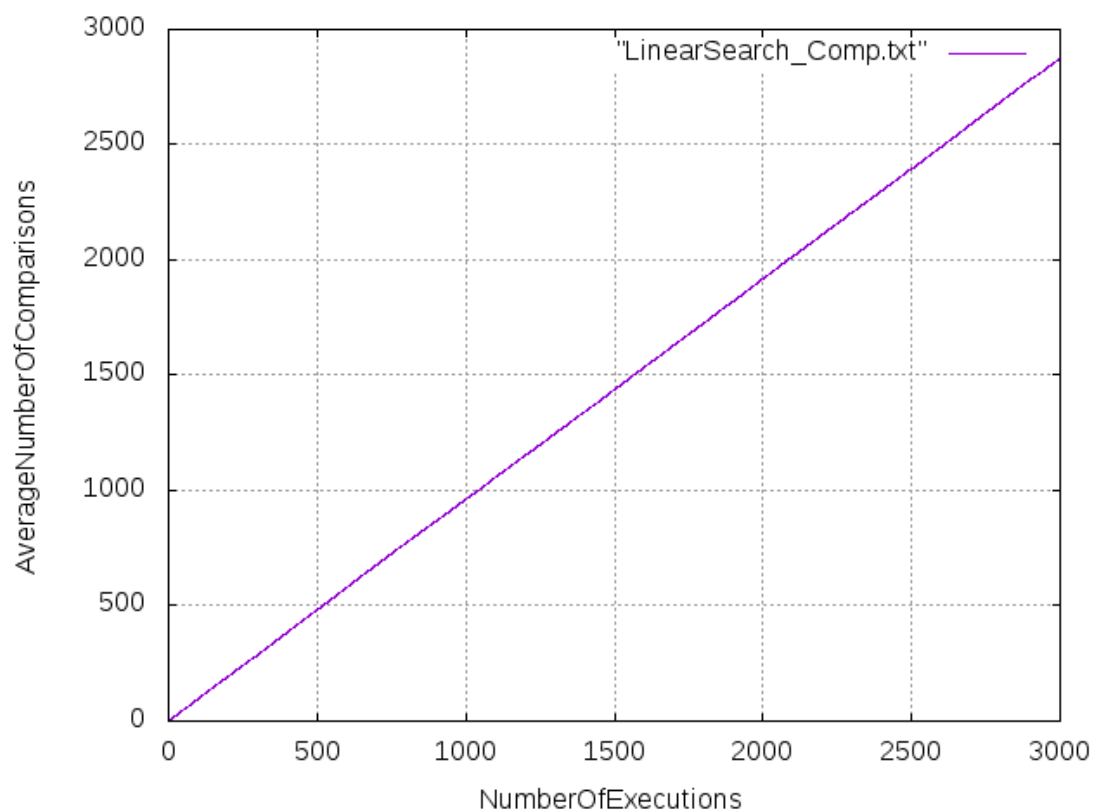
Αναζητω με γραμμικη αναζητηση τις κρατησεις με βαση το ονομα του πελατη.

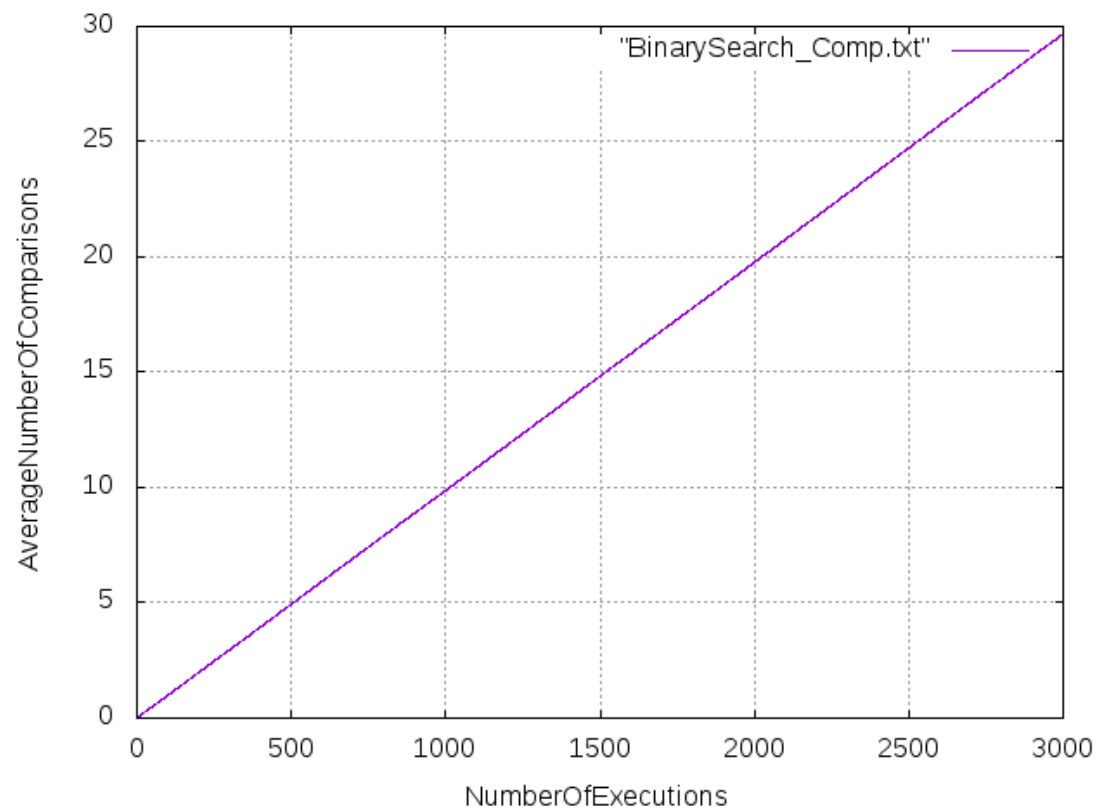
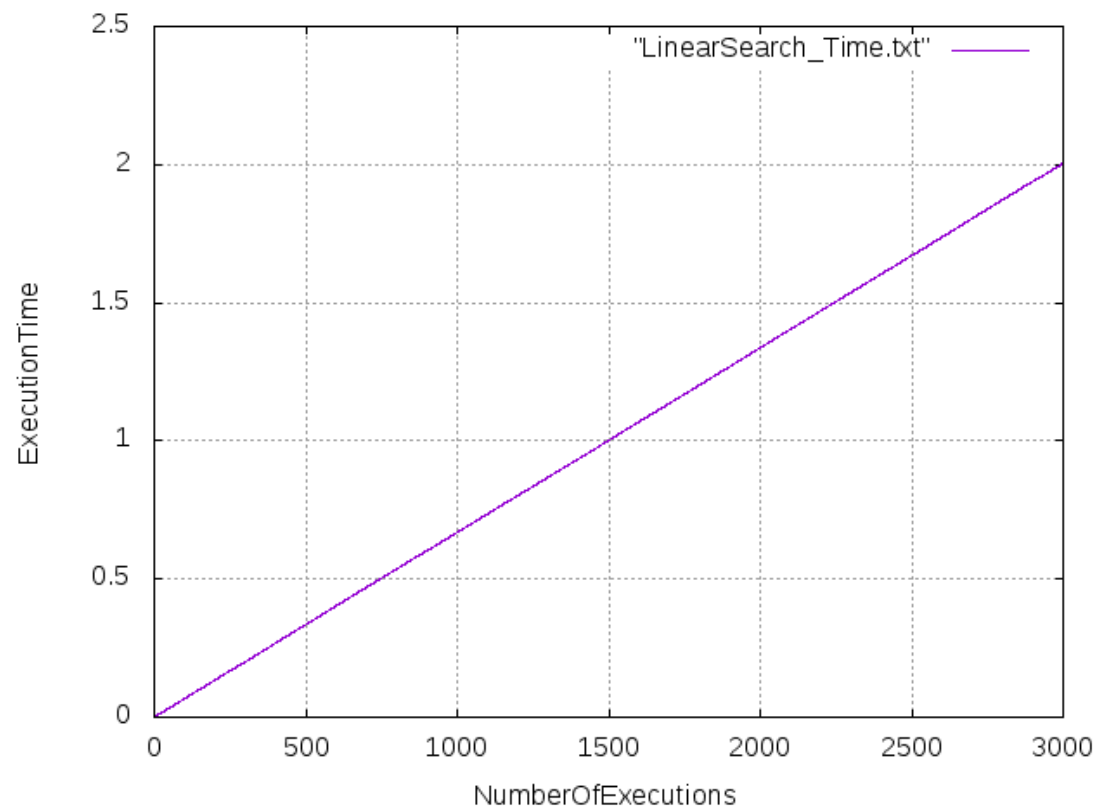
6.Exit():

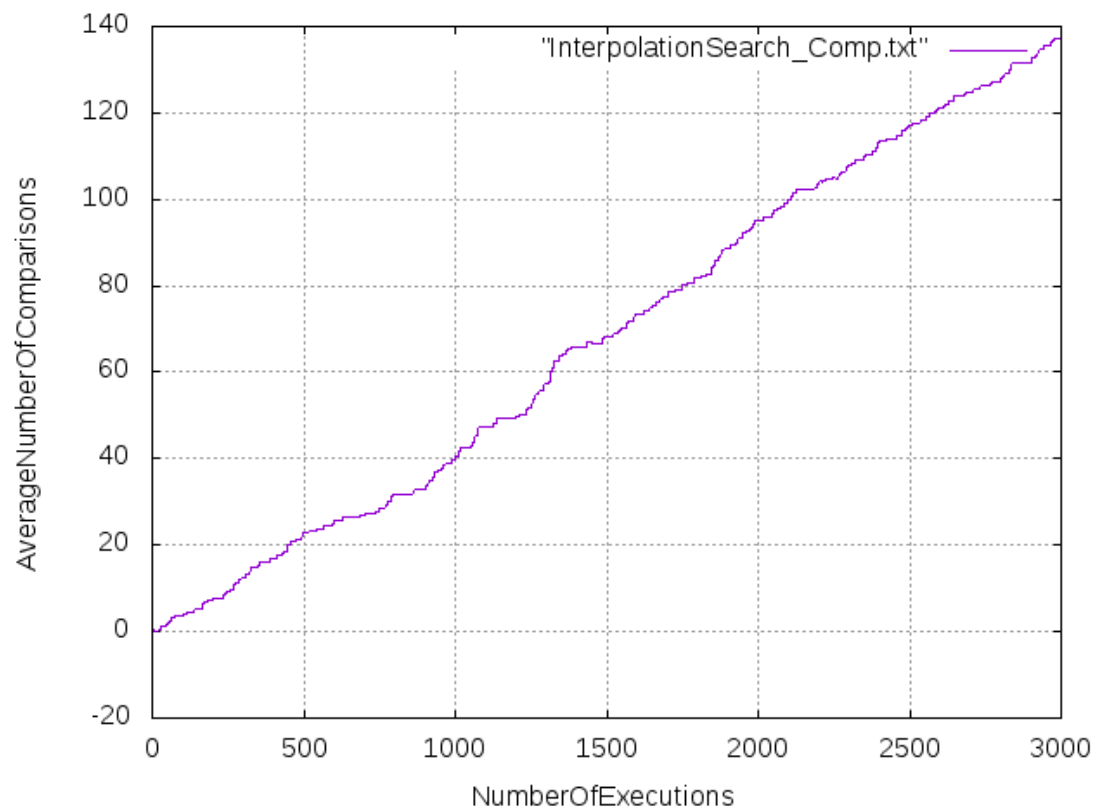
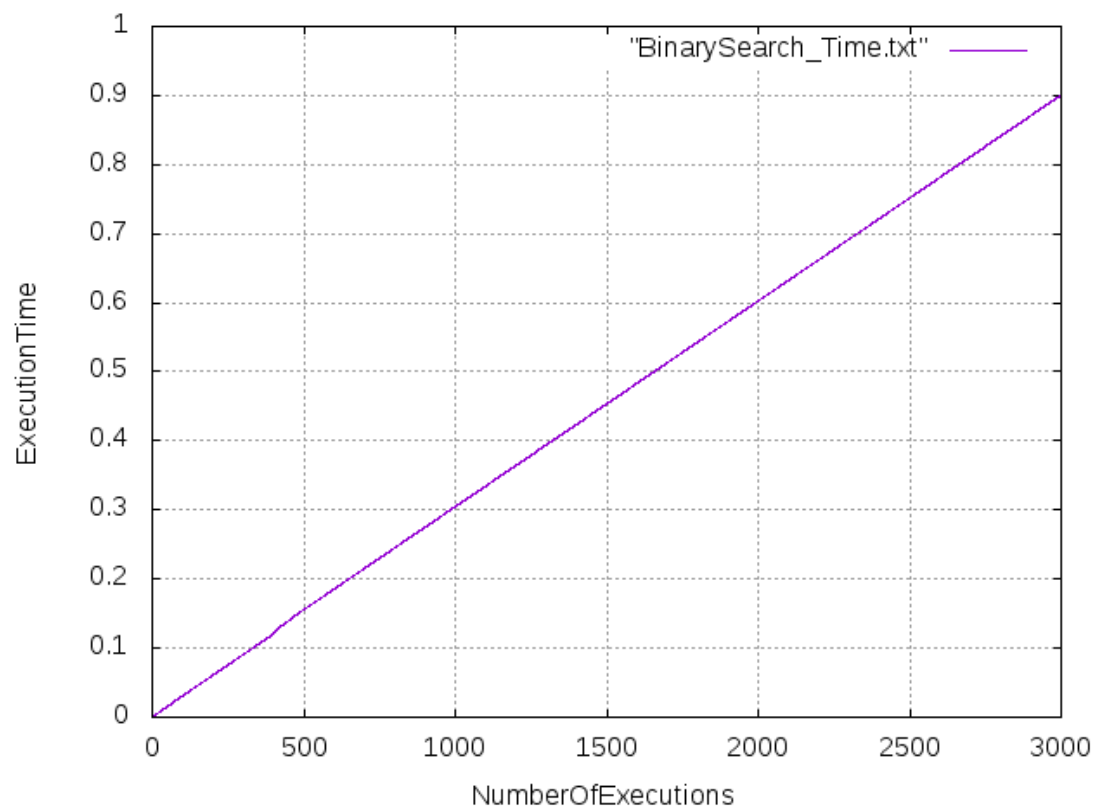
Στην συναρτηση Exit() καθαριζω τις λιστες με την εντολη `del list[:]` και μετα καλω μια αλλη συναρτηση η οποια ανανεωνει τον μετρητη των ξενοδοχειων και αποθηκευει τις τελευταιες αλλαγες.

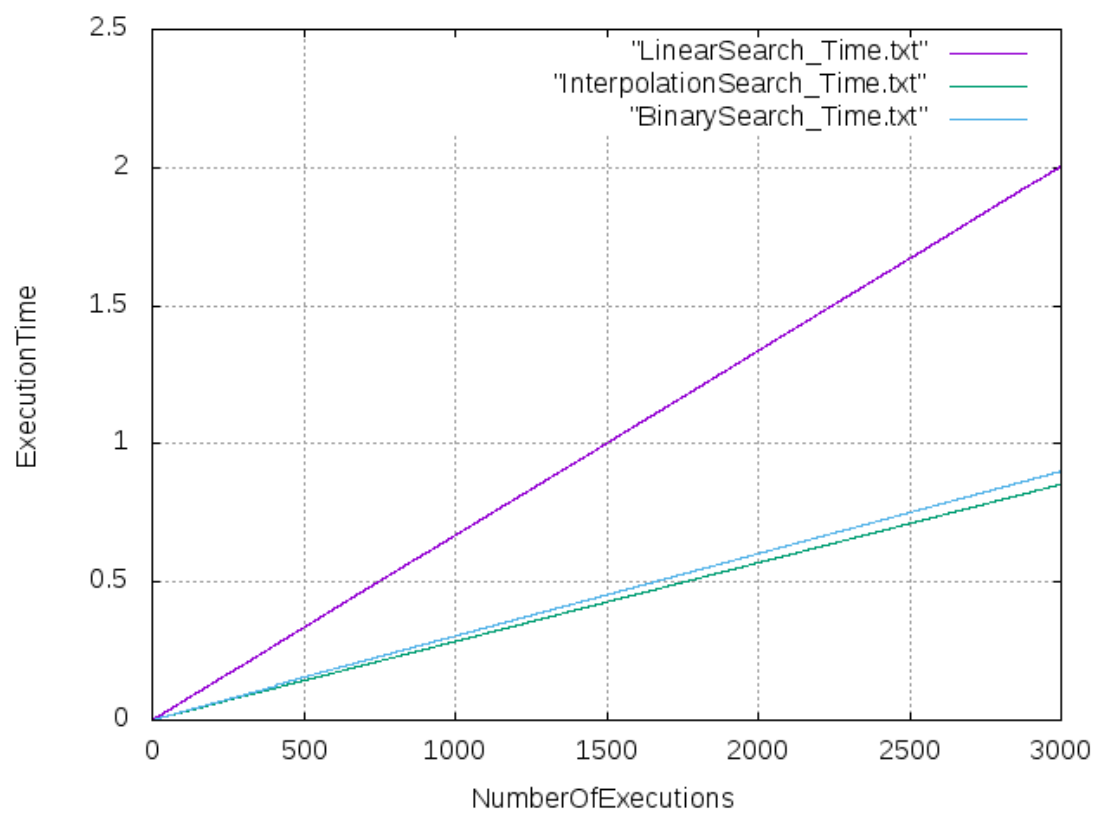
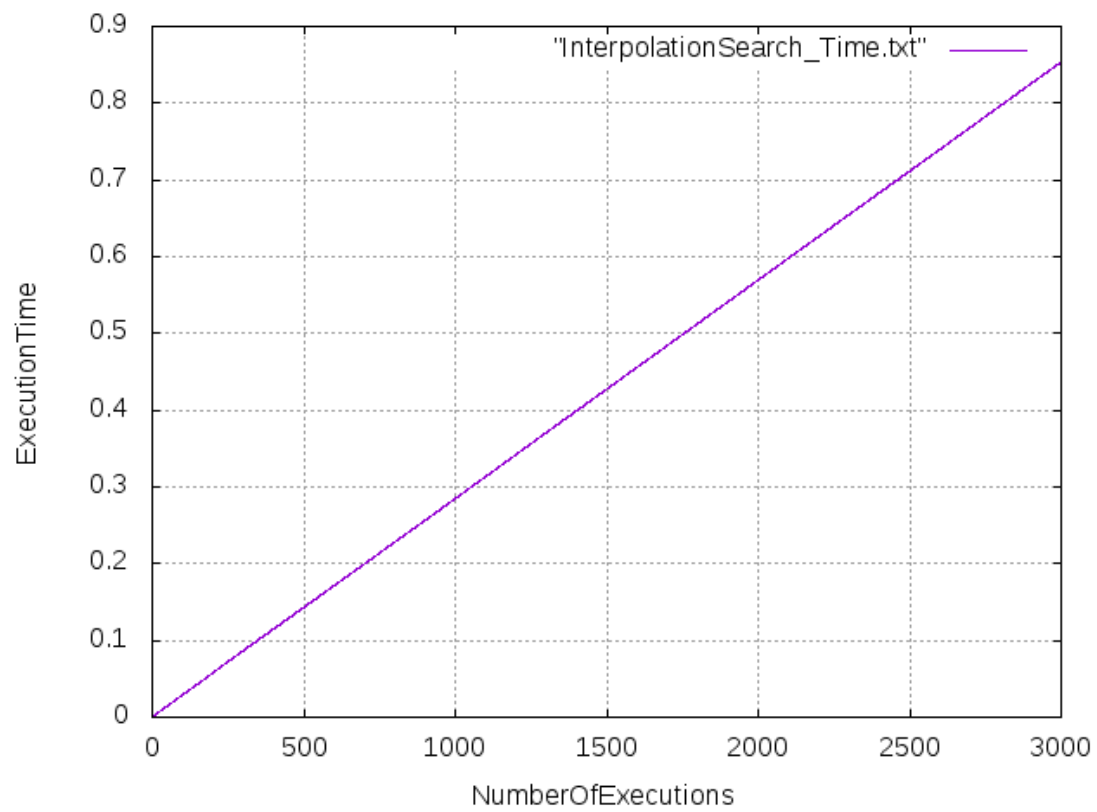
Benchmarks

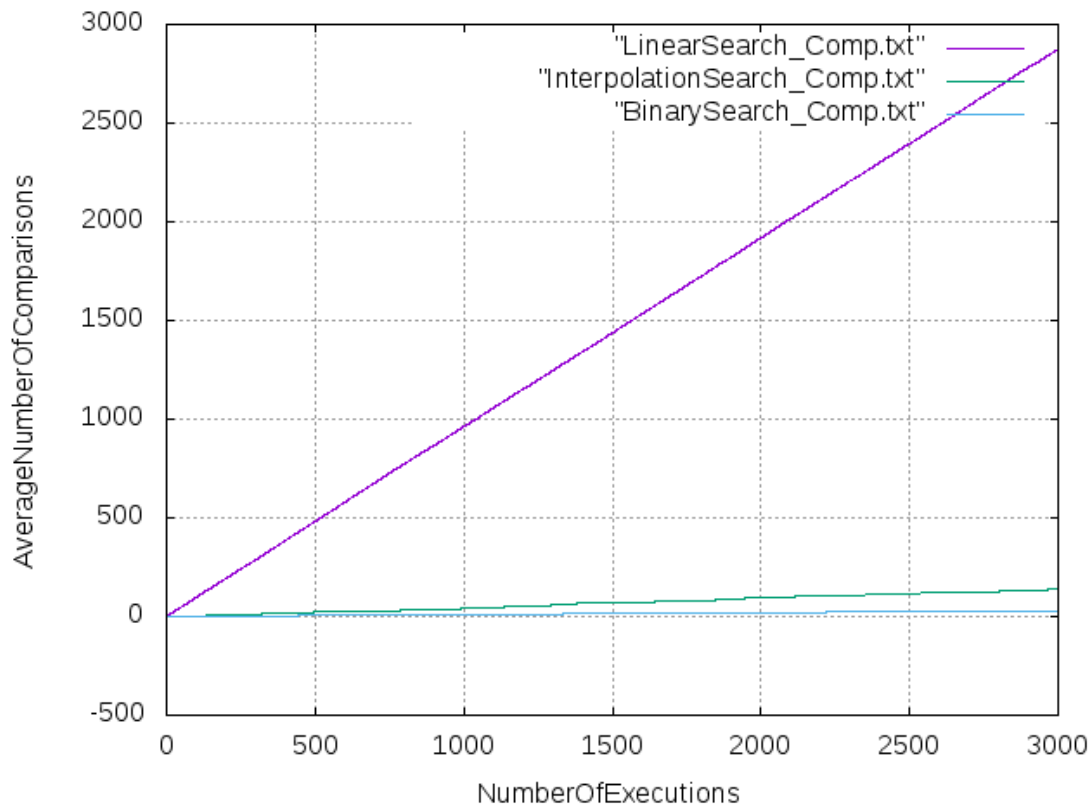
Πιο κατω ακολουθουν διαγραμματα συγκρισεων και χρονου συναρτήση των φορών εκτέλεσης.Το μετροπρόγραμμα εκτελεί 3000 αναζητήσεις φορτώνοντας ενα csv file 1000 εγγραφων.











Στα Τελευταία συγκριτικά διαγράμματα βλέπουμε ότι η δυαδική αναζήτηση είναι η πιο γρήγορη σε σχέση με την γραμμική αναζήτηση και την interpolation search που οντως ισχύει αφού θεωρητικά

Η γραμμική αναζήτηση έχει χρονική πολυπλοκότητα $O(n)$, Η δυαδική αναζήτηση έχει χρονική πολυπλοκότητα $O(\log n)$ και η interpolation search έχει $O(\log(\log n))$ σε συγκρίσεις. Όσο αναφορά το συγκριτικό διάγραμμα στον εκτελεστικό χρόνο βλέπουμε ότι οι χρόνοι της binary search σε σχέση με της interpolation search είναι σχετικά κοντά. Αυτό συμβαίνει διότι η υλοποίησή τους έχει γίνει iterative και όχι αναδρομικά (recursively) με αποτέλεσμα να είναι ακόμα γρηγορότερες σε σχέση με την γραμμική αλλά να έχουν σχεδόν την ίδια χρονική πολυπλοκότητα.