**ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ**

**PROJECT(2017)**

**!Αναφορά!**

**Κωνσταντίνος Αδαμόπουλος**

**6270**

**adamopoulos@ceid.upatras.gr**

**Python**

**ΕΙΣΑΓΩΓΗ**

Το προγραμμα τρεχει μονο σε **python3**.Αν το εκτελεσετε σε linux τοτε μπορειτε να χρησιμοποιησετε το bash script που υπαρχει στο φακελο Launcher.Το script αυτο εκτελει την εντολη εκτελεσης του προγραμματος python3 Main.py <PathToFile> και επιπλεον με τα καταλληλα argument values μπορειτε να ορισετε αν θα παρει το deafult .csv file απο τον φακελο CSVFiles οπου εκει στεγαζονται ολα τα csv files,αν θα παρει απο τον ιδιο φακελο αλλο .csv file ή αν θα παρει ενα αλλο αρχειο απο ενα αλλο path.Για περισσοτερες πληροφοριες απλως πληκτρολογηστε την εντολη **./launcher.sh -h** στο τερματικο σας.

Αν τρεχετε σε windows και θελετε να εκτελεσεται το προγραμμα με το default αρχειο που βρισκεται στον φακελο CSVFIles τοτε απλως πηγαινετε στον φακελο Source και καντε double-click το αρχειο Main.py.Αν θελετε να τρεξετε αλλο αρχειο απο τον φακελο CSVFiles ή απο αλλο path καλο ειναι να το κανετε μεσω cmd με την εντολη **python3 Main.py <PathToFile>**.

**Συναρτησεις**

1. **Load():**

Στην συναρτηση Load() διαβαζω το csv file και εκχωρω τα ids των ξενοδοχειων σαν κλειδια σε ενα dictionary και την υπολοιπη πληροφορια σαν το περιεχομενο του συγκεκριμενου κλειδιου του dictionary.

**2.Add():**

Στην συναρτηση Add() προσθετω καινουργια ξενοδοχεια μαζι με τις πληροφοριες τους καθως και τις κρατησεις τους και τα αποθηκευω σε μια λιστα.

**3.Save():**

Στην συναρτηση Save() αποθηκευω τα δεδομενα τις λιστας απο την Add().

**4.Search by id:**

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

**5.Search by surname:**

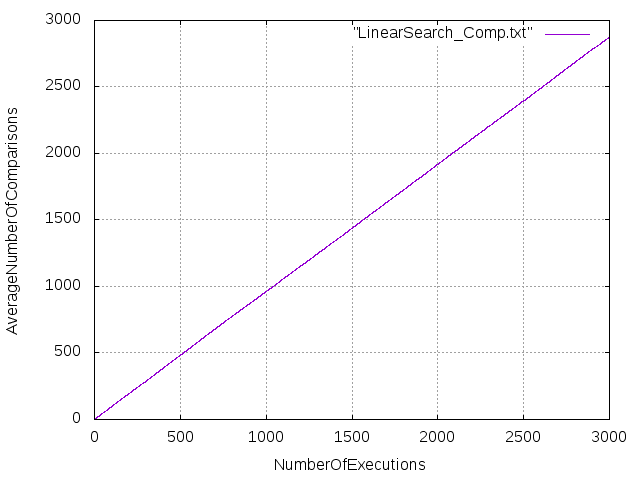
Αναζητω με γραμμικη αναζητηση τις κρατησεις με βαση το ονομα του πελατη.

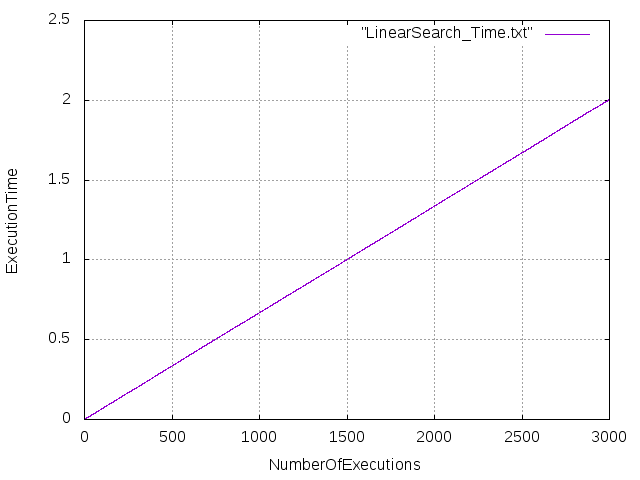
**6.Exit():**

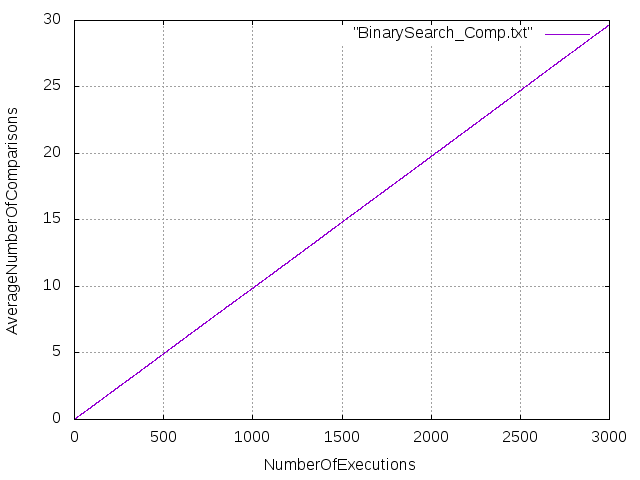
Στην συναρτηση Exit() καθαριζω τις λιστες με την εντολη del list[:] και μετα καλω μια αλλη συναρτηση η οποια ανανεωνει τον μετρητη των ξενοδοχειων και αποθυκευει τις τελευταιες αλλαγες.

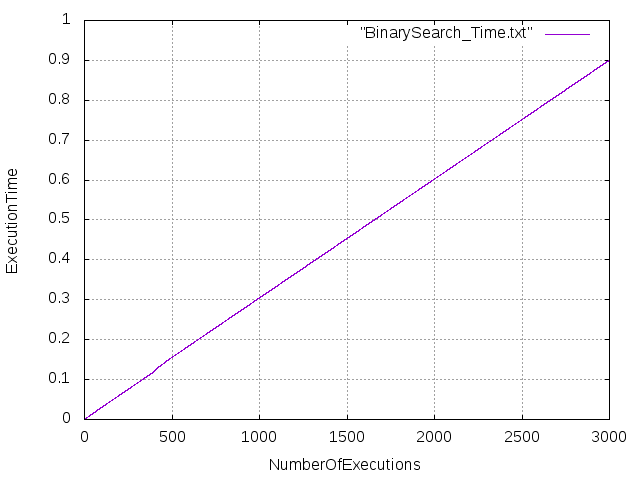
**Benchmarks**

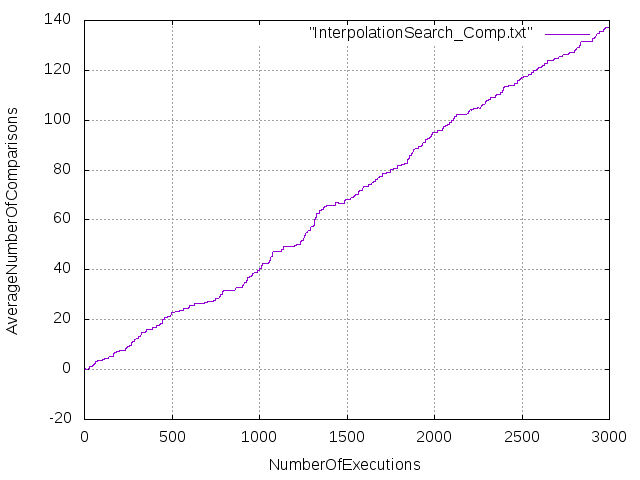
Πιο κατω ακολουθουν διαγραμματα συγκρισεων και χρονου συναρτήση των φορών εκτέλεσης.Το μετροπρόγραμμα εκτελεί 3000 αναζητήσεις φορτώνοντας ενα csv file 1000 εγγραφων.

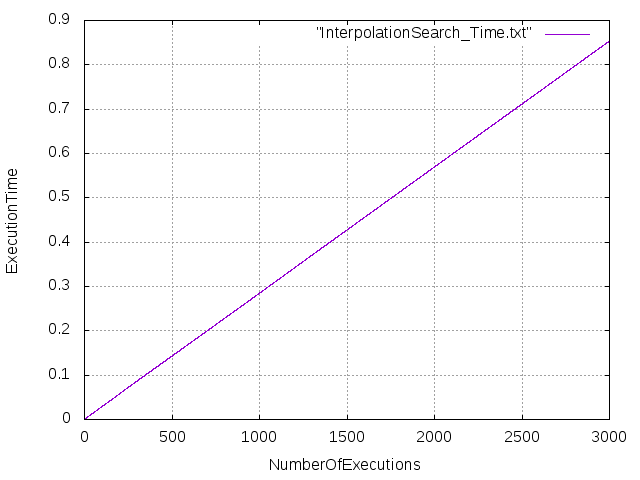


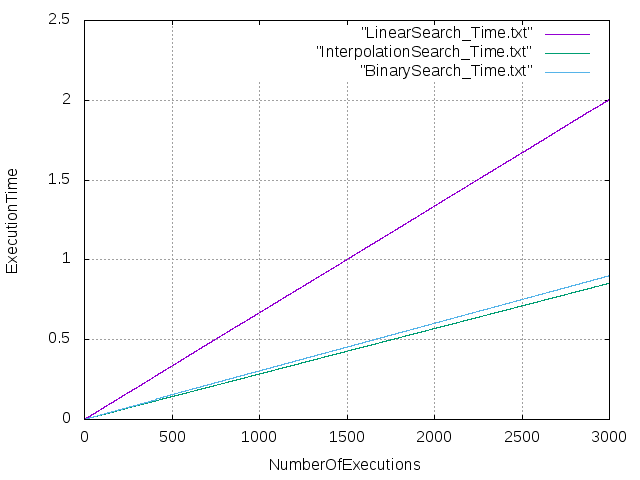


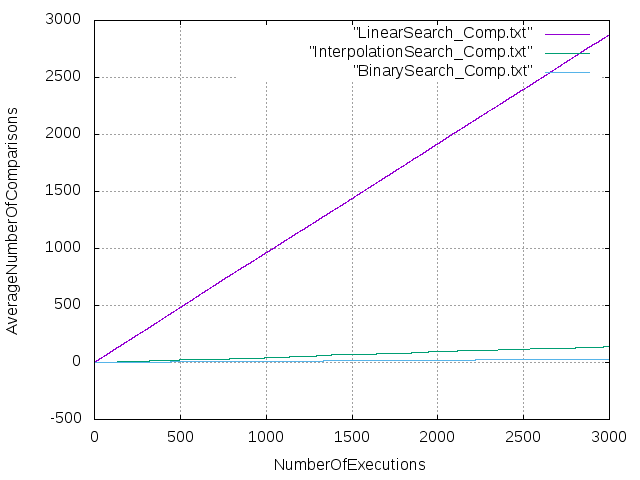












Στα Τελευταια συγκριτικα διαγραμματα βλεπουμε οτι η δυαδικη αναζητηση ειναι η πιο γρηγορη σε σχεση με την γραμμικη αναζητηση και την interpolation search που οντως ισχυει αφου θεωρητικα

Η γραμμικη αναζητηση εχει χρονικη πολυπλοκτοτητα **Ο(n)**,Η δυαδικη αναζητηση εχει χρονικη πολυπλοκοτητα **Ο(logn)** και η interpolation search εχει **Ο(log(logn))** σε συγκρισεις.Οσο αναφορα το συγκριτικο διαγραμμα στον εκτελεσιμο χρονο βλεπουμε οτι η χρονοι της binary search σε σχεση με της interpolation search ειναι σχετικα κοντα.Αυτο συμβαινει διοτι η υλοποιηση τους εχει γινει iterative και οχι αναδρομικα(recursively) με αποτελεσμα να ειναι ακομα γρηγορες σε σχεση με την γραμμικη αλλα να εχουν σχεδον την ιδια χρονικη πολυπλοκοτητα.