**Project Data Structures 2017**

**Αναφορα**

**Κωνσταντίνος Αδαμόπουλος**

**6270**

**adamopoulos@ceid.upatras.gr**

**Python**

ΕΙΣΑΓΩΓΗ

Το προγραμμα τρεχει μονο σε python3.Αν το εκτελεσετε σε linux τοτε μπορειτε να χρισημοποιησετε το bash script που υπαρχει στο φακελο Launcher.Το script αυτο εκετελει την εντολη εκτελεσης του προγραμματος python3 και εππιπλεον με τα καταλληλα argument values μποριτε να ορισετε αν θα παρει το deafult .csv file απο τον φακελο CSVFiles οπου εκει στεγαζωνται 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 πληκτρολογιστε την εντολη ./launcher.sh -h.

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 ξενοδοχια μαζι με τις πληροφοριες τους καθως και τις κρατησεις τους και τα αποθυκευω σε μια λιστα.

1. Save()

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

1. Search by id:

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

1. Search by surname:

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

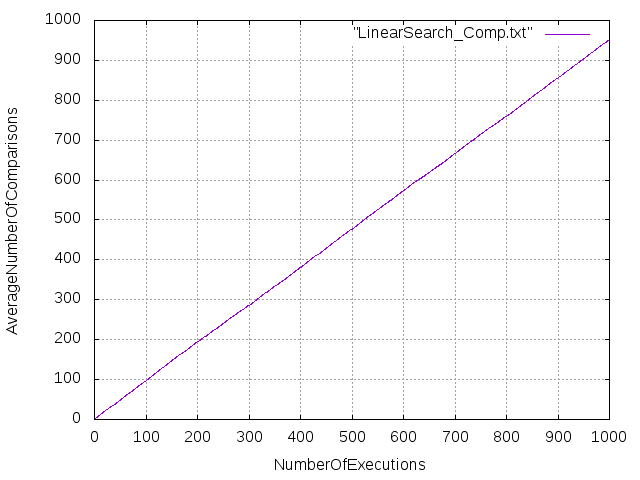
1. Exit()

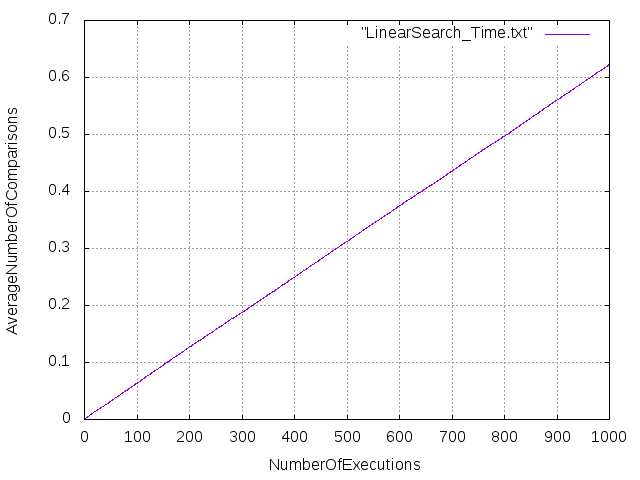
Στην συναρτηση Exit() καθαριζω τις λιστες με την εντολη del list[:] και μετα καλω μια αλλη συναρτηση η οποια ανανεωνει τον μετρητη των ξενοδοχειων και αποθυκευει τις τελευταιες αλλαγες.

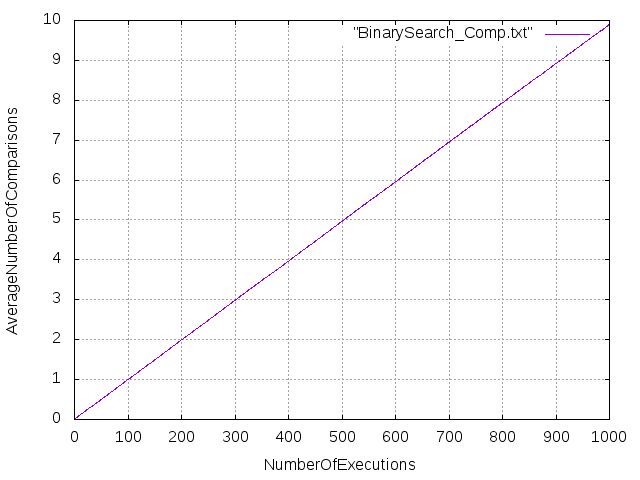
Benchmarks

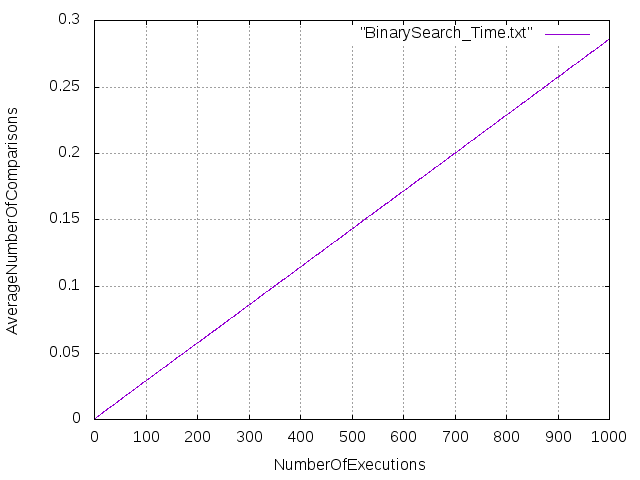
Πιο κατω ακολουθουν διαγραμματα συγκρισεων και χρονου συναρτηση των φορων εκτελεσης.

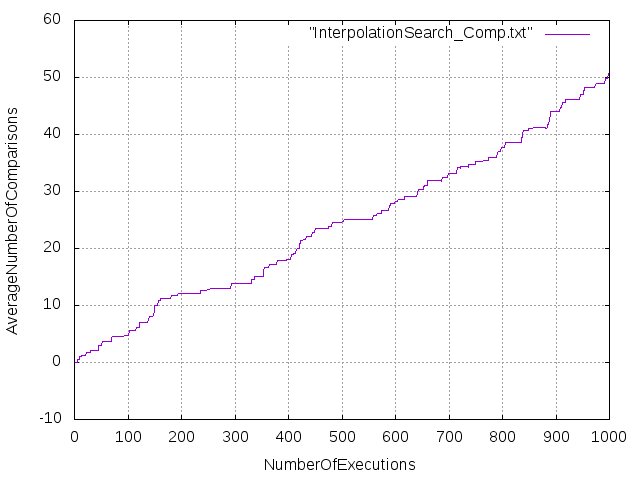
Το μετροπρογραμμα τρεχει 1000 αναζητησεις φορτονωντας ενα csv file 1000 εγγραφων.

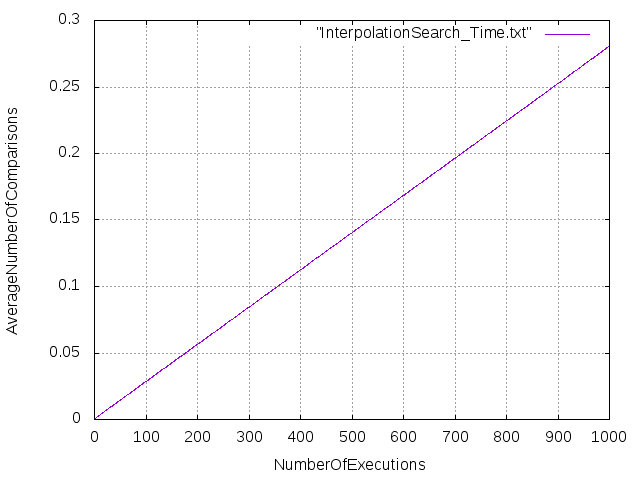


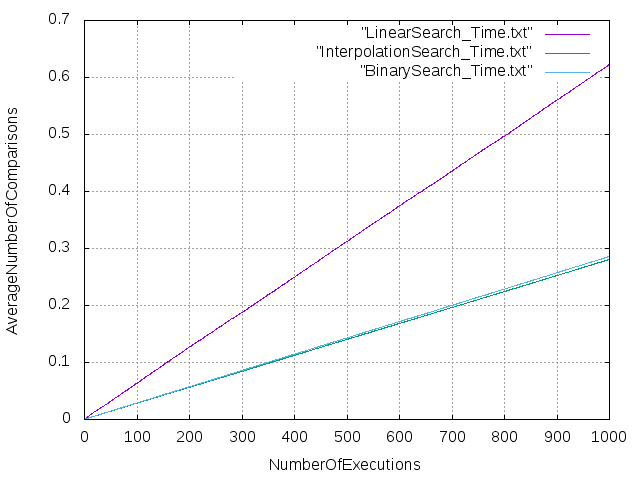


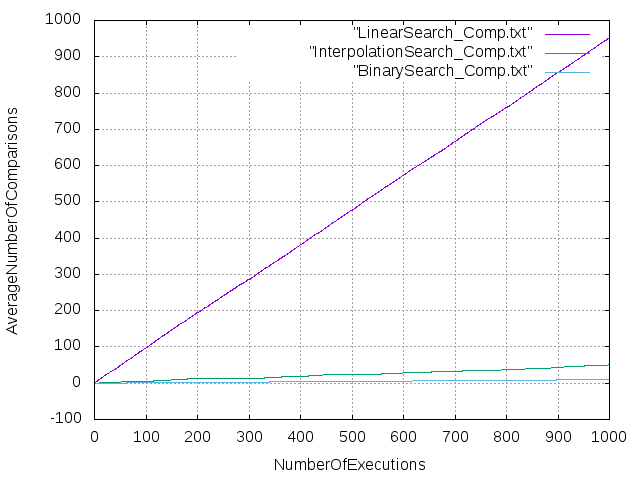












Στα Τελευταια συγκριτικα διαγραμματα βλεπουμε οτι η δυαδικη αναζητηση ειναι η πιο γρηγορη σε σχεση με την γραμμικη αναζητηση και την interpolation search που οντως ισχυει αφου θεωριτικα

Η γραμμικη αναζητηση εχει χρονικη πολυπλοκτοτητα Ο(n),Η δυαδικη αναζητηση εχει χρονικη πολυπλοκοτητα Ο(logn) και η interpolation search εχει Ο(log(logn)).