

Δομές Δεδομένων και Τεχνικές Προγραμματισμού- Εργασία 3

- **Στοιχεία:** Σάμιος – Γρηγόρης -1115201500141
- **Λειτουργικότητα :** Το πρόγραμμα αποτελεί μια λειτουργία εισαγωγής και αναζήτησης λέξεων σε μια δομή ΔΔΑ. Παίρνει λέξεις από τον χρήστη ή από αρχείο .txt, τις εισάγει στο δέντρο και τις αναζητά είτε ο χρήστης είτε αναζητούνται αυτόματα από αρχείο .txt.
- **Οδηγίες Χρήσης του προγράμματος :** Τα δεδομένα εισάγονται είτε από τον χρήστη είτε από ένα αρχείο .txt και τα αποτελέσματα προβάλλονται σε .txt. Για την επιλογή της λειτουργίας AVL ή BST απαιτείται το ανάλογο define στο main.c και στο Words.c.
- **Περιβάλλον Υλοποίησης και Δοκιμών:** Αναπτύχθηκε σε Visual Studio 2015 σε περιβάλλον Windows 10, δοκιμάστηκε σε gcc σε Linux.
- Αποτελέσματα μετρήσεων στο linux09:

Αρχείο	AVL	BST
wordsByABC.txt	InsertTime:0 seconds μετά από 1024 λέξεις InsertTime:0 seconds μετά από 2048 λέξεις InsertTime:0 seconds μετά από 4096 λέξεις InsertTime:0 seconds μετά από 8192 λέξεις InsertTime:0.02 seconds μετά από 16384 λέξεις InsertTime:0.02 seconds μετά από 32768 λέξεις InsertTime:0.05 seconds μετά από 65536 λέξεις InsertTime (all words):0.08 seconds CheckTime:0.04 seconds DiadromhTime:0.01 seconds	InsertTime:0.02 seconds μετά από 1024 λέξεις InsertTime:0.1 seconds μετά από 2048 λέξεις InsertTime:0.4 seconds μετά από 4096 λέξεις InsertTime:1.57 seconds μετά από 8192 λέξεις InsertTime:6.23 seconds μετά από 16384 λέξεις InsertTime:6.23 seconds μετά από 32768 λέξεις InsertTime:26.44 seconds μετά από 65536 λέξεις InsertTime (all words):59.39 seconds CheckTime:45.7 seconds DiadromhTime:0 seconds
wordsByRandom.txt	InsertTime:0 seconds μετά από 1024 λέξεις InsertTime:0 seconds μετά από 2048 λέξεις InsertTime:0 seconds μετά από 4096 λέξεις InsertTime:0.01 seconds μετά από 8192 λέξεις InsertTime:0.02 seconds μετά από 16384 λέξεις InsertTime:0.06 seconds μετά από 32768 λέξεις InsertTime:0 seconds μετά από 65536 λέξεις InsertTime (all words):0.11 seconds CheckTime:0.04 seconds DiadromhTime:0.01 seconds	InsertTime:0 seconds μετά από 1024 λέξεις InsertTime:0 seconds μετά από 2048 λέξεις InsertTime:0 seconds μετά από 4096 λέξεις InsertTime:0.01 seconds μετά από 8192 λέξεις InsertTime:0.02 seconds μετά από 16384 λέξεις InsertTime:0.06 seconds μετά από 32768 λέξεις InsertTime:0 seconds μετά από 65536 λέξεις InsertTime (all words):0.1 seconds CheckTime:0.06 seconds DiadromhTime:0.01 seconds
WordsByFrequency.txt	InsertTime:0 seconds μετά από 1024 λέξεις InsertTime:0 seconds μετά από 2048 λέξεις InsertTime:0 seconds μετά από 4096 λέξεις	InsertTime:0 seconds μετά από 1024 λέξεις InsertTime:0 seconds μετά από 2048 λέξεις InsertTime:0 seconds μετά από 4096 λέξεις

	InsertTime:0.01 seconds μετά από 8192 λέξεις InsertTime:0.03 seconds μετά από 16384 λέξεις InsertTime:0.07 seconds μετά από 32768 λέξεις InsertTime:0 seconds μετά από 65536 λέξεις InsertTime (all words):0.1 seconds CheckTime:0.05 seconds DiadromhTime:0.01 seconds	InsertTime:0.01 seconds μετά από 8192 λέξεις InsertTime:0.03 seconds μετά από 16384 λέξεις InsertTime:0.06 seconds μετά από 32768 λέξεις InsertTime:0 seconds μετά από 65536 λέξεις InsertTime (all words):0.11 seconds CheckTime:0.05 seconds DiadromhTime:0 seconds
--	--	--

Συμπέρασμα :Γενικά δεν παρατηρείται μεγάλη διαφορά στην εκτέλεση με τους δύο τρόπους,εκτός από την περίπτωση που οι λέξεις εισάγονται αλφαβητικά. Ήταν αναμενόμενο εφόσον στην περίπτωση του AVL το δέντρο δεν εκφυλίζεται σε λίστα ,όπως στο ΔΔΑ, γεγονός που κάνει την εισαγωγή λέξης να καθυστερήσει λόγω αναδρομής.