

Τεκμηρίωση Άσκησης 3

Στοιχεία

Όνομα: Γεώργιος
Επώνυμο: Καμάρας
ΑΜ: 1115201400058

Λειτουργικότητα

Το πρόγραμμα αυτό υλοποιεί όλες τις συναρτήσεις, τους τύπους στοιχείων και τα αρχεία που ζητούνται στα πλαίσια της άσκησης. Επιπλέον των όσων ζητήθηκαν, μαζί με τον συνολικό χρόνο εισαγωγής, εκτυπώνεται και ο μέσος χρόνος εισαγωγής, και, μαζί με τους μέσους χρόνους αναζήτησης, εκτυπώνεται πάντα και ο συνολικός χρόνος αναζήτησης. Στο αρχείο Enretirio.c έχει αναπτυχθεί επιπλέον συνάρτηση endodiataksi(), για να βοηθήσει στην αλφαβητική εκτύπωση των στοιχείων.

Οδηγίες Χρήσης

Το πρόγραμμα τρέχει ως εξής:

\$./index

και παράγει ως έξοδο όσα περιγράφονται στην Λειτουργικότητα.

Οι υλοποιήσεις των ΔΔΑ BST και AVL αναμένονται να βρίσκονται στις, σχετικές ως προς τον φάκελο του project, διευθύνσεις: **BST** και **AVL** αντίστοιχα. Δηλαδή, αν **sol3** είναι ο φάκελος του project, θα πρέπει στη μνήμη του υπολογιστή να βρίσκονται ως: **/Documents/sol3/AVL** και **/Documents/sol3/BST** αντίστοιχα (σε περιβάλλον Linux).

Τα αρχεία δεδομένων αναμένονται να βρίσκονται στη, σχετική ως προς τον φάκελο του project, διεύθυνση: **Linux Input**. Για παράδειγμα, όπως προηγουμένως: **/Documents/sol3/Linux Input** (Linux Input είναι στην ουσία ο φάκελος που δόθηκε λίγες μέρες μετά την ανακοίνωση της άσκησης, με αρχεία εισόδου για τους χρήστες Linux, στον οποίο, τρέχοντας το πρόγραμμα, προστίθενται τα αρχεία εξόδου).

Περιβάλλον Υλοποίησης και Δοκιμών

Το πρόγραμμα αναπτύχθηκε και δοκιμάστηκε σε περιβάλλον Linux Ubuntu (14.04 πλήρως ενημερωμένο) με τη χρήση του gcc. Το συγκεκριμένο έγγραφο γράφτηκε στον text editor LibreOffice Writer και έγινε "Export as PDF...".

Πίνακας Μετρήσεων

Εργασία/Χρόνος	Μέγεθος	Απλό ΔΔΑ random input	Απλό ΔΔΑ sorted input	AVL ΔΔΑ random input	AVL ΔΔΑ sorted input
Εισαγωγές (Ερ. 2)	511	2.371 msec	8.667 msec	0.774 msec	0.502 msec
	1023	4.848 msec	28.305 msec	1.474 msec	1.07 msec
	2047	10.13 msec	93.447 msec	2.904 msec	2.187 msec
	4095	-	-	-	-
	4037	20.376	293.407 msec	5.749 msec	4.568 msec
	4037	9.88631e-07 msec	3.08947e-05 msec	2.47158e-07 msec	2.47158e-07 msec
Αναζητήσεις	8788	0 msec	2.04825e-05	0 msec	0 msec

Εργασία/Χρόνος	Μέγεθος	Απλό ΔΔΑ random input	Απλό ΔΔΑ sorted input	AVL ΔΔΑ random input	AVL ΔΔΑ sorted input
(Ερ. 3)			msec		
Αναζητήσεις (Ερ. 4)	3841	2.60349e-07 msec	5.05077e-05 msec	0 msec	2.60349e-07 msec
Εκτύπωση (Επ. 5)	4037	4.591 msec	1.705 msec	2.231 msec	1.798 msec

Σημείωση: οι παραπάνω χρόνοι προκύπτουν από την εκτέλεση του προγράμματος σε περιβάλλον Linux Ubuntu (14.04 πλήρως ενημερωμένο), σε υπολογιστή με επεξεργαστή intel CORE i7 και μνήμη RAM 6 GB.

Σύντομος σχολιασμός ανωτέρω χρόνων σε σχέση με θεωρητικούς χρόνους

Η χειρότερη περίπτωση (*worst case*) για εισαγωγή, αναζήτηση και πρόσβαση (εκτύπωση) στοιχείων ενός Απλού ΔΔΑ είναι $O(n)$, ενώ ενός AVL ΔΔΑ είναι $O(\log(n))$. Δηλαδή, αναμένεται η εισαγωγή, αναζήτηση και πρόσβαση (εκτύπωση) στοιχείων ενός Απλού ΔΔΑ να χρειαστεί (αρκετά) περισσότερο χρόνο απ' ό,τι, αντίστοιχα, σε ένα AVL. Αυτό επιβεβαιώνεται και από τα από τα αποτελέσματα των μετρήσεων που περιγράφονται στον παραπάνω πίνακα. Σε αυτόν, βλέπουμε ότι η εισαγωγή σε AVL είναι πολύ ταχύτερη απ' ό,τι σε Απλό ΔΔΑ, ενώ στην αναζήτηση και στην εκτύπωση οι χρόνοι διαφέρουν ελάχιστα.

Για ένα Απλό ΔΔΑ, η χειρότερη περίπτωση (*worst case*) συναντάται στην περίπτωση μίας ταξινομημένης εισόδου (*sorted input*), καθώς για την είσοδο ενός στοιχείου στο δέντρο χρειάζεται πάντα να διασχιστεί όλο του το, έως τότε, μήκος. Το AVL ΔΔΑ, έχοντας την ικανότητα να “αλλάζει” ρίζα ώστε να “ισορροπεί”, χειρίζεται αποτελεσματικότερα μιά τέτοια περίπτωση, μειώνοντας το μήκος της διαδρομής και πετυχαίνοντας είσοδο σε πολύ καλύτερο χρόνο.

Για ένα AVL ΔΔΑ για εισαγωγή, αναζήτηση και πρόσβαση (εκτύπωση) στοιχείων, η περίπτωση μίας μη ταξινομημένης (τυχαίας) εισόδου (*random input*) και μίας ταξινομημένης εισόδου (*sorted input*), παρουσιάζουν πολύ μικρές χρονικές διαφορές μεταξύ τους. Αυτό οφείλεται στις περιστροφές του AVL, οι οποίες έχουν ως αποτέλεσμα τον περιορισμό των μεγάλων διαδρομών για την εύρεση θέσης εισαγωγής, ύπαρξης, πρόσβασης (εκτύπωσης) ενός στοιχείου (σε αντίθεση με ένα πιθανότατα μη ισοζυγισμένο Απλό ΔΔΑ). Πάντως, θα μπορούσαμε να χαρακτηρίσουμε ως χειρότερη περίπτωση (*worst case*) την μη ταξινομημένη (τυχαία) είσοδο (*random input*), καθώς απαιτεί περισσότερες περιστροφές για τον ισοζυγισμό του δέντρου, ο οποίος κατά τη διάρκεια της διαδικασίας πιθανότατα αλλάζει συνεχώς. Σε αυτό οφείλεται η ελάχιστη χρονική διαφορά που παρατηρείται στον παραπάνω πίνακα, η οποία καθιστά καλύτερη χρονικά την χρήση AVL ΔΔΑ με ταξινομημένη είσοδο (*sorted input*).