

Ονοματεπώνυμο : Γεώργιος Γαλάνης

AM: 1115201800024

Λειτουργικότητα: Έχω υλοποιήσει όλα τα ερωτήματα της άσκησης όπως θα παρατηρήσετε. Επιπροσθέτως έχω προσθέσει μερικές printf ώστε να τυπώνω το μέσο χρόνο αναζήτησης για όσα αεροδρόμια βρέθηκαν.

Οδηγίες χρήσης: Όσον αφορά την μεταγλώττιση, περα από τα AirportManagement.c, Evretirio.h, Evretirio.c, TSEvr.h, TSEvr.c , TSDDA.h, TSDDA.c στα οποία έχω προσθέσει επιπλέον πράγματα (ότι είναι απαραίτητο σε αντιστοιχία με τα πράγματα που ζητάει η εκφώνηση), έχω γράψει το AirportManagementFuncts.c το οποίο περιέχει τις συναρτήσεις που αντιστοιχούν στα ερωτήματα της άσκησης τα οποία χειρίζεται η main (AirportManagement) και το AirportManagement.h το οποίο περιέχει τα πρότυπα των παραπάνω συναρτήσεων καθώς και τα #includes για τα απαραίτητα headerfiles. Επίσης στον φάκελο που θα παραδώσω πρέπει να μπουν τα αρχεία τα οποία θα ανοίξει η main για την είσοδο, την έξοδο και την αναζήτηση (πράγμα το οποίο λογικά το γνωρίζετε αλλά το αναφέρω για να μαι σίγουρος).

Περιβάλλον ανάπτυξης και δοκιμών: Αναπτύχθηκε και δοκιμάστηκε σε gcc στο Unix

Εργασία/Χρόνος	Μέγεθος	Απλό ΔΔΑ random input	Απλό ΔΔΑ sorted input	AVL ΔΔΑ random input	AVL ΔΔΑ sorted input
Εισαγωγές (Ερ. 2)	511	0.001645 s	0.005926 s	0.001702 s	0.001493 s
	1023	0.003283 s	0.011448 s	0.003408 s	0.002917 s
	2047	0.005958 s	0.031898 s	0.00616 s	0.005341 s
	4095	0.008034 s	0.096788 s	0.008331 s	0.007216 s
	Όλα	0.01129 s	0.273339 s	0.011672 s	0.00998 s
		1.57419e-06 s	3.74702e-05 s	1.59841e-06 s	1.33964e-06 s
Αναζητήσεις(Ερ.3)	Όλα	5.66857e-06 s	0.000531962 s	5.12764e-06 s	4.42901e-06 s
(Ερ.3)	Βρέθηκαν	3.06372e-07 s	2.87512e-05 s	2.77137e-07 s	2.39377e-07 s
Εκτύπωση (Ερ. 5)	Όλα	0.001702 s	0.001114 s	0.001772 s	0.001106 s

Σχολιασμός: Όσον αφορά τις εισαγωγές και την αναζήτηση για τα BST, ισχύουν οι ίδιες πολυπλοκότητες $O(n)$ αφού στην χειρότερη περίπτωση θα πρέπει να διατρέξουμε όλους τους κόμβους αν βρίσκονται όλοι πχ μόνο στο δεξί παιδί κάθε κόμβου-πατέρα. Στην εκτύπωση σίγουρα θα διατρέξουμε όλους τους κόμβους οπότε παλι $O(n)$. Οι διαφορές χρόνου που παρατηρούνται στις εισαγωγές και στις αναζητήσεις οφείλονται στο γεγονός ότι εφόσον τα δεδομένα είναι ήδη ταξινομημένα κάθε κόμβος βρίσκεται μόνο στο ένα παιδί του πατέρα του οπότε παντα ισχυει η χειρότερη περίπτωση. Για τα AVL, ισχύει η ίδια πολυπλότητα για την εκτύπωση οπότε έτσι εξηγείτε το γεγονός ότι οι χρόνοι εκτύπωσης είναι σχετικά κοντά μεταξύ τους ενώ εφόσον το δέντρο είναι σχετικά ισοζυγισμένο οι πολυπλοκότητες για τις άλλες πράξεις είναι $O(\log n)$ οπότε παρατηρούμε μικρότερους χρόνους σε σχέση με τα BST. Τέλος θεωρώ ότι οι χρόνοι για το sorted input για το AVL είναι μικρότεροι από το random γιατί παρατηρούμε περιστροφές από μόνο μια πλευρά οπότε δεν χρησιμοποιούνται δυο περιστροφές και από τις δύο πλευρές για να ρυθμίσουν το ύψος του δέντρου αφού σε περίπτωση αύξουσας διάταξης κάθε κόμβος τοποθετείται στο δεξιότερο φύλλο και έπειτα αν η διαφορά ύψους με το αριστερό είναι μεγαλύτερη του 1 γίνεται μια περιστροφή.