Η/Α Επέκταση Εργασίας 2021-22
Διαχείριση Στοιχείων Αεροδρομίων με τον ΑΤΔ Ευρετήριο (με χρήση ΑΤΔ r/b tree)

Σας δίδονται τα στοιχεία 7184 αεροδρομίων σε όλο τον κόσμο (από το http://openflights.org/data.html) με την εξής δομή

Airport ID	Unique OpenFlights identifier for this airport.
Name	Name of airport. May or may not contain the **City** name.
City	Main city served by airport. May be spelled differently from **Name**.
Country	Country or territory where airport is located. See countries.dat to cross-reference to ISO 3166-1 codes.
IATA	3-letter IATA code. Null if not assigned/unknown. **IATA**-International Air Transport Association, represents most major scheduled airlines, deals with commercial aspect of airline operations, i.e. ticketing, interline baggage transfer, liability limits, etc.
ICAO	4-letter ICAO code. Null if not assigned. **ICAO**-International Civil Aviation Organization, is a branch of the United Nations, represents aviation authorities of UN member nations.

Τα στοιχεία των αεροδρομίων περιέχονται σε δυο αρχεία, ένα ταξινομημένο (airportsWinSorted) σε αύξουσα σειρά βάσει **Airport ID** και ένα με τυχαία διάταξη (airportsWinRandom). Σας δίδονται επίσης αντίστοιχα αρχεία σε Linux (η διαφορά στο EOL). Μια τυπική γραμμή (αεροδρόμιο) είναι η ακόλουθη:

11176;Umberto Modiano Airport;Buzios;Brazil;BZC;SBBZ

Σας δίδεται επίσης ένα αρχείο που περιέχει πληροφορίες για 67290 συνδέσεις (routes) μεταξύ των αεροδρομίων με την εξής δομή (όχι πτήσεις αεροπορικών εταιριών, μόνο συνδέσεις)

Airline	2-letter (IATA) or 3-letter (ICAO) code of the airline.
Airline ID	Unique OpenFlights identifier for airline (see Airline).
Source airport	3-letter (IATA) or 4-letter (ICAO) code of the source airport.
Source airport ID	Unique OpenFlights identifier for source airport (see Airport)
Destination airport	3-letter (IATA) or 4-letter (ICAO) code of the destination airport.
Destination airport ID	Unique OpenFlights identifier for destination airport (see Airport)
Codeshare	"Y" if this flight is a codeshare (that is, not operated by *Airline*, but another carrier), empty otherwise.
Stops	Number of stops on this flight ("0" for direct)
Equipment	3-letter codes for plane type(s) generally used on this flight, separated by spaces

Τυπικές γραμμές του αρχείου - δίδονται σε δυο μορφές για Windows (routesWin) και linux (routesLinux).

3U;4608;WUH;3376;JJN;6386;;0;319

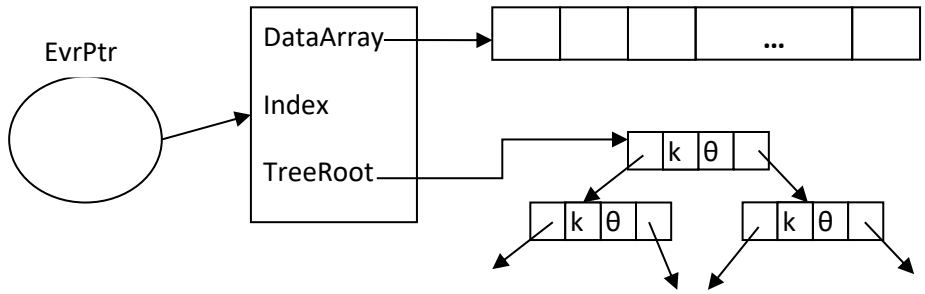
3U;4608;XIY;3379;CTU;3395;;0;320 319

Ζητείται να κατασκευάστε ευρετήριο Αεροδρομίων (με χρήση του ΑΤΔ Ευρετήριο) και να ενημερώσετε από το αρχείο των συνδέσεων (routes) τα στοιχεία των αεροδρομίων με τον αριθμό αναχωρήσεων και αφίξεων.

Α. Ο ΑΤΔ Ευρετήριο (Evr)

Αποτελείται από 1) έναν πίνακα, όπου αποθηκεύονται τα στοιχεία των αεροδρομίων (που δίδονται στο αρχείο airports και επιπλέον μετρητές αναχωρήσεων-αφίξεων συνδέσεων), 2) έναν ακέραιο που δείχνει το πρώτο διαθέσιμο κελί στον πίνακα για την επόμενη εισαγωγή αεροδρομίου στον πίνακα, και 3) την ρίζα ενός R/B Δυαδικού Δένδρου Αναζήτησης, όπου αποθηκεύονται α) ο κωδικός **Airport ID**, που είναι το κλειδί (key) τουR/B ΔΔΑ και β) η θέση (int) στον πίνακα με τα πλήρη δεδομένα του αεροδρομίου.

```
typedef struct EvrNode * EvrPtr;
struct EvrNode{
       typos_deikti TreeRoot;          /* Root of BST Tree */
       TStoixeiouOther *DataArray;     /* array of size MaxSize */
       int Index;                      /* index of first available element in array */
};
```



Ακολουθούν οι πράξεις του Evr

`EvrPtr EvrConstruct(int MaxSize);` Δημιουργεί έναν κόμβο EvrNode και αρχικοποιεί τα 3 μέλη του. Ειδικά για το DataArray δημιουργεί δυναμικά έναν πίνακα MaxSize στοιχείων. Πίνακας με 7200 στοιχεία είναι αρκετός.

`int EvrDestruct(EvrPtr *E);` Καταστρέφει την Δομή Ε (όλα τα στοιχεία του δένδρου, τον πίνακα και τον κόμβο EvrNode). Επιστρέφει ένδειξη αν εντοπίσει λάθος.

`int EvrInsert(EvrPtr E,TStoixeiouEvr Data);` Εισάγει στο πρώτο ελεύθερο στοιχείο του πίνακα τα δεδομένα Data, και εισάγει στο ΔΔΑ το κλειδί key μαζί με την θέση, όπου εισήχθησαν τα Data. Επιστρέφει ένδειξη αν εντοπίσει λάθος.

`int EvrSearch(EvrPtr E, keyType key, int InOut, int *found);` Αναζητά στην Ε (ΔΔΑ) το κλειδί key και ανάλογα θέτει τιμή στο found. Επιστρέφει ένδειξη αν εντοπίσει λάθος. Αν το βρει, ανάλογα με την τιμή της InOut (0 ή 1) αυξάνει κατά ένα τις αφίξεις ή τις αναχωρήσεις.

`int EvrPrintAll(EvrPtr E, FILE *out, int Counter);` Εκτυπώνει στο αρχείο out όλα τα στοιχεία με αλφαβητική σειρά των κλειδιών (ενδοδιάταξη), τις αναχωρήσεις και αφίξεις (όχι τα άλλα στοιχεία). Επιστρέφει ένδειξη αν εντοπίσει λάθος.

Οι πράξεις να υλοποιηθούν στο Evr.c χρησιμοποιώντας πράξεις του ΑΤΔ R/B που έχετε αναπτύξει.

Απαιτείται επίσης να αναπτύξετε τύπους στοιχείων TSEvr (για τον πίνακα) και TSDDA (για το ΔΔΑ) με οδηγό τους τύπους που σας έχουν δοθεί.

Συνιστάται να αναπτύξτε και να ελέγξετε τις πράξεις του Evr σε ξεχωριστό πρόγραμμα test.c πριν απαντήσετε στα επόμενα ερωτήματα για το πρόγραμμα-πελάτη.

Β. Διαχείριση Στοιχείων Αεροδρομίων - Πρόγραμμα Πελάτη

1) Δημιουργήστε ένα Ευρετήριο.

2) Εισάγετε στοιχεία στο Ευρετήριο (ΔΔΑ+πίνακα) από το αρχείο AirportsRandom που περιέχει τα στοιχεία των αεροδρομίων με τους κωδικούς σε τυχαία διάταξη α) να εισάγετε τα στοιχεία στον πίνακα και β) να εισάγετε στο ΔΔΑ τον κωδικό (κλειδί) μαζί με την θέση του πίνακα, όπου αποθηκεύτηκαν τα στοιχεία. Επίσης να μετράτε και να εμφανίζετε τον χρόνο εισαγωγής μετά από (2^N -1) στοιχεία, όπου N=9,10,11,12, καθώς τον ολικό αριθμό στοιχείων και και τον ολικό χρόνο που απαιτήθηκε.

3) Ενημέρωση στοιχείων αφίξεων και αναχωρήσεων. Για κάθε σύνδεση από το αρχείο routes

α) αναζητάτε το AirportIn στο ΔΔΑ και αν βρεθεί αυξάνετε τον μετρητή των αφίξεων στον πίνακα

β) αναζητάτε το AirportOut στο ΔΔΑ και αν βρεθεί αυξάνετε τον μετρητή των αναχωρήσεων στον πίνακα.

Κατά την εισαγωγή μετράτε πόσες συνδέσεις διαβάσατε, πόσα αεροδρόμια βρέθηκαν και πόσα δεν βρέθηκαν στο ΔΔΑ και τον συνολικό χρόνο που χρειάστηκε η ενημέρωση. Υπολογίστε τον μέσο όρο αναζήτησης για κάθε αεροδρόμιο. Εκτυπώσετε τις τιμές των μετρητών και χρόνων στο αρχείο OUTPUTRandomBST.txt.

4) Να συνεχίσετε την εκτύπωση στο αρχείο OUTPUTRandomBST.txt με τους κωδικούς των αεροδρομίων σε αύξουσα σειρά (ενδοδιάταξη) μαζί με τις αναχωρήσεις και αφήξεις. Τέλος εκτυπώστε τον αριθμό των στοιχείων που εκτύπωσε και τον συνολικό χρόνο που απαιτήθηκε για την διαδρομή της εκτύπωσης.

5) Να καταστρέψετε το Ευρετήριο.

6) Να επαναλάβετε τα ζητούμενα 1-5, διαβάζοντας όμως τα αεροδρόμια από το AirportSorted, που περιέχει αερδρόμια ταξινομημένα σε αύξουσα σειρά και παράγοντας το αρχείο OUTPUTSortedBST.txt.

Οδηγίες Σχεδίασης και Ανάπτυξης Προγράμματος

Σας δίδετε σκελετός προγράμματος, που απαιτείται να αναπτύξετε. Το πρόγραμμά σας να είναι οργανωμένο σε ενότητες (modules) και σε πρόγραμμα-πελάτη. Να αναπτύξετε το AirportManagement.c, την υλοποίηση του Evr.h στο Evr, TSEvr.h, TSEvr.c, TSDDA.h και TSDDA.c.

Το κυρίως πρόγραμμά σας να είναι οργανωμένο σε συναρτήσεις που να αντιστοιχούν στα ερωτήματα (ιδίως τα 2, 3, 4) της άσκησης, ώστε το σώμα της main να είναι σύντομο. Να ελέγχετε την εγκυρότητα των δεδομένων, ενδείξεις λάθους από συναρτήσεις βιβλιοθηκών, του Evr.h, κλπ. Να αναπτύξτε και να ελέγξετε τις πράξεις του Evr σε ξεχωριστό πρόγραμμα test.c πριν απαντήσετε στα ερωτήματα 2 έως 5.

Για την χρονομέτρηση σας δίνονται στον κατάλογο TimingDemos τρία προγράμματα επίδειξης: το SecResolutionTiming.c χρησιμοποιεί την βιβλιοθήκη της C με ακρίβεια δευτερολέπτου και τα άλλα δύο WinMilliSecResTiming.c και LinuxMillSecResTiming.c βιβλιοθήκες των Windows και Linux αντίστοιχα για ακρίβεια msec, που μάλλον θα σας χρειαστεί για μεγαλύτερη ακρίβεια μετρήσεων.

Παραδοτέα και Οδηγίες Παράδοσης

1. Πηγαίος κώδικας, δηλαδή AirportManagement.c, Evr.h, Evr.c, TSEvr.h, TSEvr.c , TSDDA.h, TSDDA.c και υλοποιήσεις του R/B. Όσοι αναπτύσσετε σε dev να συμπεριλάβετε και το .dev και όσοι στο Linux το Makefile.
2. Αρχείο Τεκμηρίωσης (pdf) με την εξής δομή:
   - Τα στοιχεία σας: (Όνομα-Επώνυμο-ΑΜ)
   - Λειτουργικότητα: Ποια ερωτήματα της άσκησης έχετε υλοποιήσει.
   - Οδηγίες Χρήσης του προγράμματος σας: π.χ. Ορισμός ονομάτων αρχείων
   - Περιβάλλον Υλοποίησης και Δοκιμών: πχ. Αναπτύχθηκε σε Dev C++ σε περιβάλλον Windows XP, δοκιμάστηκε επίσης σε gcc και Linux.
   - Τον ακόλουθο πίνακα συμπληρωμένο σύμφωνα με τις μετρήσεις σας (στα κενά βάζετε χρόνους, όπου «μέσος όρος» τον αντίστοιχο μέσο όρο, και στα σκιασμένα κελιά τον αντίστοιχο αριθμό στοιχείων.

Εργασία/Χρόνος	**μέγεθος**	**R/B ΔΔΑ random input**	**R/B ΔΔΑ sorted input**
Εισαγωγές (Ερ. 2)	511		
	1023		
	2047		
	4095		
	Όλα		
		Μέσος όρος	Μέσος όρος
Αναζητήσεις (Ερ.3)	Όλα	Μέσος όρος	Μέσος όρος
(Ερ. 3)	Βρέθηκαν	Μέσος όρος	Μέσος όρος
Εκτύπωση (Ερ. 5)	Όλα		

   - Σύντομο σχολιασμό των χρόνων σε σχέση με τους θεωρητική πολυπλοκότητα που αναμένατε (π.χ. O(n), O(logn), $O(n^2)$)

Το αρχείο τεκμηρίωσης μαζί με τα πηγαία αρχεία του προγράμματος να τα βάλετε σε έναν φάκελο (directory), τον οποίο θα συμπιέσετε (zip, rar) και θα ανεβάσετε στο eclass.