

**3<sup>η</sup> Άσκηση ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ 2018-19**  
**Διαχείριση Στοιχείων Αεροδρομίων με τον ΑΤΔ Ευρετήριο (με χρήση ΔΔΑ)**  
**Προθεσμία 26/05/2019 στις 20:00**

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

<b>Airport ID</b>	Unique OpenFlights identifier for this airport.
<b>Name</b>	Name of airport. May or may not contain the <b>City</b> name.
<b>City</b>	Main city served by airport. May be spelled differently from <b>Name</b> .
<b>Country</b>	Country or territory where airport is located. See <a href="#">countries.dat</a> to cross-reference to ISO 3166-1 codes.
<b>IATA</b>	3-letter IATA code. Null if not assigned/unknown. <b>IATA</b> -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
<b>ICAO</b>	4-letter ICAO code. Null if not assigned. <b>ICAO</b> -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 ταξινομημένα (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

<b>Airline</b>	2-letter (IATA) or 3-letter (ICAO) code of the airline.
<b>Airline ID</b>	Unique OpenFlights identifier for airline (see <a href="#">Airline</a> ).
<b>Source airport</b>	3-letter (IATA) or 4-letter (ICAO) code of the source airport.
<b>Source airport ID</b>	Unique OpenFlights identifier for source airport (see <a href="#">Airport</a> )
<b>Destination airport</b>	3-letter (IATA) or 4-letter (ICAO) code of the destination airport.
<b>Destination airport ID</b>	Unique OpenFlights identifier for destination airport (see <a href="#">Airport</a> )
<b>Codeshare</b>	"Y" if this flight is a codeshare (that is, not operated by <i>Airline</i> , but another carrier), empty otherwise.
<b>Stops</b>	Number of stops on this flight ("0" for direct)
<b>Equipment</b>	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;JIN;6386;;0;319

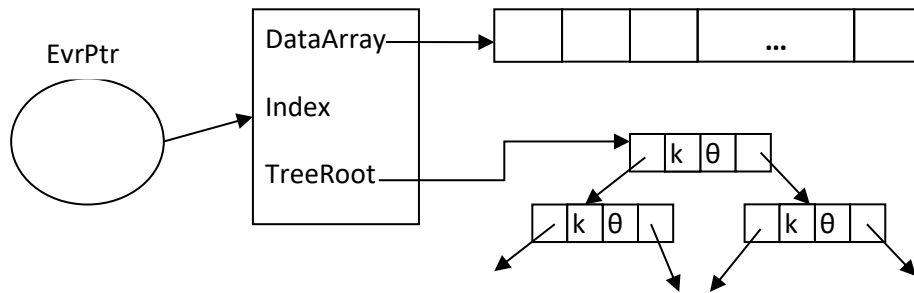
3U;4608;XIV;3379;CTU;3395;;0;320 319

Ζητείται να κατασκευάσετε 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.

#### **A. 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 Δυαδικού Δένδρου Αναζήτησης, όπου αποθηκεύονται α) ο κωδικός **Airport ID**, που είναι το κλειδί (key) του 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 Evt

`EvtPtr EvtConstruct(int MaxSize);` Δημιουργεί έναν κόμβο `EvtNode` και αρχικοποιεί τα 3 μέλη του. Ειδικά για το `dataArray` δημιουργεί δυναμικά έναν πίνακα `MaxSize` στοιχείων. Πίνακας με 7200 στοιχεία είναι αρκετός.

`int EvtDestruct(EvtPtr *E);` Καταστρέφει την Δομή `E` (όλα τα στοιχεία του δένδρου, τον πίνακα και τον κόμβο `EvtNode`). Επιστρέφει ένδειξη αν εντοπίσει λάθος.

`int EvtInsert(EvtPtr E, TStoixeiouEvt 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 EvtSearch(EvtPtr E, keyType key, int InOut, int *found);` Αναζητά στην `E` (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 EvtPrintAll(EvtPtr 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 `Evt.c` χρησιμοποιώντας πράξεις των ΑΤΔ ΔΔΑ και ΑΤΔ AVL (μία διεπαφή `Evt.h` και δυο υλοποιήσεις στο ίδιο αρχείο `Evt.c` με χρήση `#if ...#else...#endif` – δείτε τον κώδικα που σας δίδεται). Οι υλοποιήσεις των ΑΤΔ ΔΔΑ και AVL σας δίνονται έτοιμες στα αντίστοιχα ζεύγη αρχείων (`BST.c`, `BST.h`) και (`AVL.c`, `AVL.h`) σε υποφακέλους `BST` και `AVL`, αντίστοιχα. **Δεν πρέπει και δεν χρειάζεται να επεμβείτε στον κώδικά τους.**

Οι δυο υλοποιήσεις `BST` και `AVL` έχουν παρόμοια διεπαφή. Οι διαφορές είναι α) ότι οι πράξεις του απλού `BST` έχουν το πρόθεμα `Tree` ενώ του `AVL` το `AVLTree` και β) οι πράξεις εισαγωγής και διαγραφής σε `AVL` χρειάζονται μια `*int` παράμετρο επιπλέον (που δεν χρειάζεται αρχική τιμή κατά την κλήση των συναρτήσεων). Δεν πρέπει να αλλάξετε την υλοποίηση των `BST`, `AVL` ή τα ονόματα των αρχείων δεδομένων. Το πρόγραμμά σας θα ελεγχθεί με αυτά τα αρχεία. Ως οδηγό της υλοποίησης μπορείτε να δείτε την τεχνική της υλοποίησης ουράς με χρήση του ΑΤΔ Λίστα στις διαφάνειες με τις εφαρμογές τις λίστες (7.8.3 των πρότυπων υλοποιήσεων).

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

Συνιστάται να αναπτύξετε και να ελέγξετε τις πράξεις του `Evt` 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 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. Εκτυπώστε 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 αεροδρόμια ταξινομημένα σε αύξουσα σειρά και παράγοντας το αρχείο OUTPUTSortedBST.txt.

Να μεταγλωττίσετε και εκτελέσετε το κυρίως πρόγραμμα με τις δυο υλοποιήσεις του Evt (BST και AVL). Για το AVL το επίθεμα BST στο αρχείο εξόδου αντικαθίσταται με το επίθεμα AVL.

### 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 Evt.h στο Evt.c (ολική απόκρυψη με δύο εναλλακτικές υλοποιήσεις για απλό BST και AVL), TSEvt.h, TSEvt.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 Evt.h, κλπ. Να αναπτύξετε και να ελέγξετε τις πράξεις του Evt 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 LinuxMilliSecResTiming.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, Evt.h, Evt.c, TSEvt.h, TSEvt.c, TSDDA.h, TSDDA.c (όχι τις υλοποιήσεις των ΔΔΑ, ούτε τα αρχεία δεδομένων, ούτε τα .o). 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-AM)
  - 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	Απλό ΔΔΑ random input	Απλό ΔΔΑ sorted input	AVL ΔΔΑ random input	AVL ΔΔΑ sorted input
Εισαγωγές (Ερ. 2)	511				
	1023				
	2047				
	4095				
	Όλα				
		Μέσος όρος	Μέσος όρος	Μέσος όρος	Μέσος όρος
Αναζητήσεις (Ερ.3)	Όλα	Μέσος όρος	Μέσος όρος	Μέσος όρος	Μέσος όρος
(Ερ. 3)	Βρέθηκαν	Μέσος όρος	Μέσος όρος	Μέσος όρος	Μέσος όρος
Εκτύπωση (Ερ. 5)	Όλα				

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

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. Προσοχή ανεβάστε το στην κατάλληλη κατηγορία υλοποίησης (Dev-C++ ή gcc).

**Ημερομηνία Παράδοσης Κυριακή 26/5/2019 στις 20:00 (πριν βγουν τα αποτελέσματα των ευροεκλογών).**

### Τρόπος Αξιολόγησης

Οι ασκήσεις είναι **ατομικές** και θα ελεγχθούν για ομοιότητες χρησιμοποιώντας ειδικό σύστημα εντοπισμού ομοιοτήτων/αντιγραφών. Σε περίπτωση αντιγραφής όλοι οι εμπλεκόμενοι θα κόβονται στο μάθημα για το τρέχον ακαδημαϊκό έτος. Θα αξιολογηθούν η λειτουργικότητα, η δομή και η τεκμηρίωση του προγράμματος. Αναλυτικά:

#### Λειτουργικότητα (70/100)

- |                                                    |         |
|----------------------------------------------------|---------|
| 1. Ενρ Δημιουργία                                  | (05/70) |
| 2. Ενρ Εισαγωγή Στοιχείων από αρχείο               | (10/70) |
| 3. Ενρ Αναζήτηση Δεδομένων από αρχείο              | (05/70) |
| 4. Ενρ Εκτύπωση σε αρχείο                          | (15/70) |
| 5. Ενρ Καταστροφή                                  | (05/70) |
| 6. Κυρίως Πρόγραμμα (είσοδος με Random και Sorted) | (20/70) |
| 7. TSEnr                                           | (05/70) |
| 8. TSDDA                                           | (05/70) |

#### Δομή (15/100)

- |                                                                  |         |
|------------------------------------------------------------------|---------|
| 9. Διπλή Υλοποίηση σε ένα πηγαίο με απλό ΔΔΑ και AVL ΔΔΑ         | (05/15) |
| 10. Δομημένο Πρόγραμμα-πελάτη (μορφοποίηση, σχόλια, έλεγχοι κλπ) | (10/15) |

#### Τεκμηρίωση και Παρουσίαση (15/100)

- |                                             |         |
|---------------------------------------------|---------|
| 11. Πίνακας Χρόνων και σχολιασμός           | (10/15) |
| 12. Υπόλοιπο Τεκμηρίωσης, γενική παρουσίαση | (05/15) |

**ΠΡΟΣΟΧΗ:** Για να αξιολογηθεί το πρόγραμμα σας (έστω για την δομή του) πρέπει τουλάχιστον να μεταγλωττίζεται. Αν δεν μεταγλωττίζεται δεν παίρνει βαθμό. Πριν παραδώσετε το πρόγραμμά σας δοκιμάστε το μια τελευταία φορά και βεβαιωθείτε ότι παραδίδετε τα σωστά αρχεία.