



Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Μάθημα: Δομές Δεδομένων

Εργασία Ακ. Έτος 2015-2016

Μέρος Α (25%):

- Στην παρούσα εργασία ζητείται να γραφτεί το λογισμικό για την οργάνωση στοιχείων ξενοδοχείων.
- Η αποθήκευση όλης της πληροφορίας θα γίνεται σε ειδικά διαμορφωμένο αρχείο δεδομένων.
- Με την έναρξη του λογισμικού διαχείρισης, όλες οι εγγραφές από το αρχείο μεταφορτώνονται στην κύρια μνήμη όπου ο χρήστης θα μπορεί να επεξεργάζεται την συλλογή (πχ. να προσθέτει ξενοδοχείο, να επεξεργαστεί μία εγγραφή ξενοδοχείου ή κράτησης, να κάνει ερωτήματα αναζήτησης κλπ.).
- Με την λήξη του προγράμματος ή όταν το επιλέγει ο χρήστης, τα δεδομένα από την μνήμη θα αποθηκεύονται στο αρχείο.
- Παρακάτω περιγράφονται οι βασικές δομή των δεδομένων και λειτουργίες με παραδείγματα σε C.
- Η υλοποίηση των λειτουργιών πρέπει να γίνει με άλλη γλώσσα προγραμματισμού (πχ. C++, C#, Java, Python).

Για κάθε εγγραφή Ξενοδοχείου χρησιμοποιείται η ακόλουθη δομή:

```
typedef struct Hotel {  
    int id;  
    char name[256];  
    int stars;  
    int numberOfRooms;  
    Reservation *reservations;  
};
```

Για κάθε Κράτηση χρησιμοποιείται η ακόλουθη δομή:

```
typedef struct Reservation{  
    char name[56];  
    date checkinDate;  
    int stayDurationDays;  
};
```

- Υλοποιήστε τις κατάλληλες συναρτήσεις ώστε να μπορούν να πραγματοποιηθούν οι ακόλουθες πράξεις μέσα από ένα μενού επιλογών:
 1. Load Hotels and Reservations from file
 2. Save Hotels and Reservations to file
 3. Add a Hotel (μαζί και τις κρατήσεις του)
 4. Search and Display a Hotel by id
 5. Display all Hotels of specific stars category and number of reservations
 6. Display Reservations by surname search
 7. Exit
- Το αρχείο στο οποίο θα σώζονται ή θα διαβάζονται τα δεδομένα θα περνιέται σαν argument στη main. Σε περίπτωση που δεν περαστεί argument, θα χρησιμοποιείται μια default τιμή για ένα τοπικό αρχείο.

- Προτείνεται το αρχείο να έχει την παρακάτω μορφή (.csv) για να υπάρχει εύκολη επέκταση όταν θα χρησιμοποιηθεί το αρχείο εισόδου του 5^{ου} μέρους της εργασίας.

NumberOfHotels

id;name;stars;numberOfRooms;name;checkinDate;stayDurationDays;name; checkinDate;stayDurationDays...

.....

Στην πρώτη γραμμή του αρχείου είναι το πλήθος των γραμμών (ξενοδοχεία). Κάθε ξενοδοχείο έχει μία εγγραφή (γραμμή) μαζί με τις κρατήσεις. Το ξενοδοχείο περιγράφεται από τις πρώτες 4 στήλες και στην συνέχεια ανά 3 στήλες είναι τα στοιχεία κράτησης.

- Στην επιλογή 1, ο τρέχων πίνακας ξενοδοχείων αποδεσμεύεται και θα δημιουργείται εκ νέου με malloc. Για το λόγο αυτό, η πρώτη τιμή που θα γράφεται στο αρχείο θα είναι το πλήθος των εγγραφών (Ξενοδοχείων) που ακολουθούν. Στην επιλογή 3, ο πίνακας θα αυξάνει σε μέγεθος κατά 1 (με realloc) και το νέο στοιχείο θα μπαίνει στο τέλος. Στις επιλογές 4 και 5, θα γίνεται γραμμική αναζήτηση χρησιμοποιώντας id, ενώ στην επιλογή 6, θα γίνεται αναζήτηση με βάση το επώνυμο αντίστοιχα.
- Σε αυτό το πρώτο μέρος οι πράξεις της αναζήτησης (επιλογές 4,5,6) θα γίνονται με γραμμική αναζήτηση.

Μέρος Β (10%):

- **Υλοποιήστε** την αναζήτηση με βάση το id (επιλογές 4,5) με χρήση δυναμικής αναζήτησης και interpolation search. Αυτό έχει ως αποτέλεσμα να πρέπει να διατηρείται η συλλογή ταξινομημένη με βάση το id.

Μέρος Γ (30%):

- Με χρήση των RED-BLACK δέντρων, οργανώστε την αναζήτηση με βάση το id (επιλογή 4). Η δομή θα δημιουργείται όταν φορτώνονται οι εγγραφές στη μνήμη, και θα αποδεσμεύεται όταν τερματίζει η εφαρμογή. Πρέπει να **υλοποιήσετε** τη δομή με τις πράξεις εισαγωγής και εύρεσης.

Μέρος Δ (20%):

- Με χρήση των ψηφιακών δέντρων (digital tries), οργανώστε την αναζήτηση με βάση το όνομα που έχει γίνει μία κράτηση. Η δομή θα κατασκευάζεται μόλις φορτώνονται οι εγγραφές στη μνήμη και θα αποδεσμεύεται όταν τερματίζει η εφαρμογή. Θα δημιουργείτε ένα συνολικό TRIE για όλες τις κρατήσεις από όλα τα ξενοδοχεία. Σε περίπτωση που υπάρχει ξανά το επίθετο ενός πελάτη από άλλο ξενοδοχείο πρέπει στο αποτέλεσμα να επιστρέφονται όλα τα πιθανά ξενοδοχεία. Στα φύλλα του TRIE θα βρίσκονται τα ονόματα των ξενοδοχείων που περιέχουν την κράτηση και τα στοιχεία της κράτησης. Πρέπει να **υλοποιήσετε** τη δομή και τις πράξεις εισαγωγής και εύρεσης στο TRIE.

Μέρος Ε (15%):

- Χρησιμοποιήστε το αρχείο data.csv (; delimited) που σας δίνεται και εκτελέστε ένα μεγάλο αριθμό από αναζητήσεις (πχ 1000) με βάση το id (τυχαία id στο range min-max των κλειδιών των δεδομένων) και με βάση κάποια επίθετα. Μετρήστε τον μέσο χρόνο αναζήτησης **σε συγκρίσεις** και **εκτελέσιμο χρόνο** σε κάθε ένα από τα προηγούμενα μέρη. (**Πειραματική αξιολόγηση:** Μέρος Α,Β,Γ επιλογή 4 | Μέρος Α,Β επιλογή 5 | Μέρος Α,Δ επιλογή 6) Τι παρατηρείτε όσον αφορά την απόδοση; Αναλύστε τα συμπεράσματά σας με βάση την απόδοση της κάθε μεθόδου που μας δίνει

η θεωρία. Δώστε γραφήματα που να δείχνουν τα αποτελέσματα σας. Τα πειράματα πρέπει να εκτελεστούν με τυχαία σύνολα ερωτημάτων ίδιου μεγέθους τουλάχιστον 10 φορές και παίρνετε ως αποτέλεσμα τον μέσο όρο.

Προσοχή: Το Red-Black και το TRIE πρέπει να τροποποιηθούν ώστε τα δεδομένα που επιστρέφονται στον χρήστη να μην εξάγονται από χρήση κάποιας άλλης ανεξάρτητης δομής.

Σημείωση 1: Το λογισμικό σας πρέπει να τρέχει και ανεξαρτήτως του αρχείου εισόδου του 5^{ου} μέρους. Θα παίρνει ως είσοδο ένα αρχείο data.csv που αν είναι άδειο δεν θα φορτώνει κάτι στη μνήμη και θα αποθηκεύονται σε αυτό όλες οι εισαγωγές του χρήστη. Αν περιέχει εγγραφές αφού φορτωθούν και γίνουν αλλαγές θα αποθηκεύονται πίσω στο ίδιο αρχείο. Οι εγγραφές στο αρχείο δεν είναι ανάγκη να είναι σε αύξουσα ταξινόμηση με βάση το id αλλά όταν η μέθοδος που εκτελείται επιβάλει τα ids να είναι σε αύξουσα σειρά πρέπει να γίνονται οι κατάλληλες ενέργειες.

Σημείωση 2: Στο μενού ανάλογα με τη βασική επιλογή μπορείτε να βάλετε υποεπιλογές ώστε να επιλέγεται η μέθοδος της αναζήτησης.

Παραδοχή: Τα ids είναι μοναδικά και δεν επιλαμβάνονται. Δεν χρειάζεται αν δεν θέλετε να γίνεται έλεγχος αλλά να είστε σίγουροι στις δοκιμές σας ότι δεν επαναλαμβάνονται.

Παρότρυνση: Μπορείτε να χρησιμοποιήσετε ότι προσφέρουν οι βιβλιοθήκες που παρουσιάστηκαν στο μάθημα αλλά **όχι** τυχόν έτοιμες υλοποιήσεις των αναζητήσεων, των Red-Black trees και των digital tries.

Διαδικαστικά:

Η άσκηση είναι ατομική!

Παράδοση: 28/06/2016

Εξέταση Εργαστηριακής Άσκησης: Θα ανακοινωθεί μετά την παράδοση των ασκήσεων

Παράδοση μέσω της πλατφόρμας του eclass (Eclass- Δομές Δεδομένων) eclass.upatras.gr

Προσοχή: στη μορφή και το όνομα του αρχείου που θα αποστείλετε (βλέπετε παρακάτω στα παραδοτέα). Ένα αρχείο rar/zip με όνομα αριθμόςμητρώου.zip (πχ 1821.zip)

Πρέπει να χρησιμοποιηθεί **ίδια γλώσσα προγραμματισμού** για όλα τα ζητούμενα.

Το αρχείο αυτό θα περιέχει:

- Αναφορά (doc, pdf) με τα ζητούμενα του κάθε ερωτήματος (τεκμηρίωση, γραφικές παραστάσεις κλπ). Στην αρχή της αναφοράς να αναγράφετε επίσης το όνομα, το ΑΜ, το e-mail και η γλώσσα προγραμματισμού που χρησιμοποιήσατε. ΔΕΝ πρέπει να περιέχονται αυτούσια κομμάτια κώδικα εκτός λίγων γραμμών όπου χρειάζεται για δικαιολόγηση.
- Τον πηγαίο κώδικα ΟΛΩΝ των προγραμμάτων που έχετε υλοποιήσει.
- Αρχεία που βοηθούν στην μεταγλώττιση, πχ makefiles, solutions files του VS, κλπ.
- Το αρχείο αυτό ΔΕΝ θα περιέχει τα εκτελέσιμα αρχεία.

Το μάθημα περιλαμβάνει τελική γραπτή εξέταση και εργασία.

1. Η παράδοση όλων των εργαστηριακών ασκήσεων είναι απαραίτητη για την συμμετοχή στην γραπτή εξέταση. Τα έτη ≤ 4 ου πρέπει να έχουν πρέπει να έχουν ολοκληρώσει τις εργαστηριακές ασκήσεις για να προσέλθουν σε γραπτή εξέταση. Οι βαθμοί των ασκήσεων θα προσμετρήσουν στην τελική βαθμολογία εφόσον ο βαθμός της τελικής γραπτής εξέτασης είναι ≥ 5 .

2. Βαθμολογία Άσκησης και Τελικός Βαθμός Μαθήματος:

Ανάλογα με το ποσοστό ολοκλήρωσης της εργασίας, και όταν αυτό είναι ικανοποιητικό, αυτή μπορεί να προσθέσει μέχρι και **1.5 βαθμό** στον **βαθμό της γραπτής εξέτασης** εφόσον παραδοθεί στην τρέχουσα περίοδο (Εξεταστική Ιουνίου) ενώ αν παραδοθεί τον Σεπτέμβριο μπορεί να προσθέσει μέχρι και 1 βαθμό. Θεωρείται ότι έχει παραδοθεί εργασία αν ο προσθετικός βαθμός που ανακοινωθεί είναι μεγαλύτερος του 0 (0-1,5) και έχει υλοποιήσει $\geq 50\%$ της εργασίας. Η βαθμολογία της εργασίας έχει ισχύ μόνο για το τρέχον ακ.έτος στο οποίο έχει παραδοθεί. Από τη στιγμή που έχει παραδοθεί μια εργασία θεωρείται ότι ο φοιτητής έχει καλύψει την υποχρέωση του ως προς αυτή και δεν χρειάζεται να παραδώσει ξανά σε επόμενα ακ. έτη όπου θα προσέλθει σε γραπτή εξέταση. Βεβαίως μπορεί να ξαναπαραδώσει σε περίπτωση που επιθυμεί να βελτιώσει τον βαθμό της γραπτής εξέτασης.

3. Απορίες σχετικά με τις εργασίες υποβάλλονται στο φόρουμ του μαθήματος (my.ceid) και στις ώρες γραφείου που θα ανακοινωθούν με βάση το ΑΜ σας.