	Πανεπιστήμιο Θεσσαλίας Τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική
	Δομές Δεδομένων Ακαδημαϊκό Έτος 2015-2016 <a href="http://emarkou.users.uth.gr/greek/teach/data_structures">http://emarkou.users.uth.gr/greek/teach/data_structures</a> Ε. Μάρκου

## Σετ Ασκήσεων

**Ημερομηνία Παράδοσης:** Παρασκευή 20 Μαΐου 2016.

**Τρόπος Παράδοσης:** Χειρόγραφες ή εκτυπωμένες σελίδες που τις αφήνετε στη γραμματεία του τμήματος (κατά τις ώρες λειτουργίας της).

### Άσκηση 1 [40 μονάδες]

Έστω ότι μια βιβλιοθήκη σας παρέχει πρόσβαση σε ουρές χαρακτήρων. Η βιβλιοθήκη σας επιτρέπει να ορίσετε μια ουρά και να καλέσετε τις 5 βασικές λειτουργίες σε αυτή. Για παράδειγμα, ο ορισμός μιας ουράς γίνεται γράφοντας: Queue Q; Για την ουρά υποστηρίζονται οι εξής λειτουργίες:

- (1) void MakeEmptyQueue(Queue Q) /\* αρχικοποιεί την ουρά Q \*/
- (2) boolean IsEmptyQueue(Queue Q) /\* επιστρέφει 1 αν η ουρά Q είναι κενή και 0 διαφορετικά \*/
- (3) char Front(Queue Q) /\* επιστρέφει το χαρακτήρα που βρίσκεται στο εμπρός μέρος της ουράς Q \*/
- (4) char Dequeue(Queue Q) /\* επιστρέφει το χαρακτήρα που βρίσκεται στο εμπρός μέρος της ουράς Q και τον διαγράφει από την Q \*/
- (5) void Enqueue(Queue Q, char x) /\* εισάγει τον χαρακτήρα x στην ουρά Q \*/

#### i. Παλίνδρομη Συμβολοακολουθία

Μια συμβολοακολουθία  $a$  ονομάζεται παλίνδρομη αν έχει τη μορφή  $a = wcw^T$ , όπου  $w^T$  είναι η αντίστροφη συμβολοακολουθία της  $w$  και  $c$  είναι ένας χαρακτήρας. Για παράδειγμα, η αντίστροφη συμβολοακολουθία της “yam” είναι η “may”. Έτσι, π.χ., η συμβολοακολουθία “yamamay” είναι παλίνδρομη. Να παρουσιαστεί αλγόριθμος που θα διαβάζει μια συμβολοακολουθία από το πληκτρολόγιο και θα ελέγχει αν αυτή είναι παλίνδρομη χρησιμοποιώντας μία ουρά.

#### ii. Ταξινόμηση ακολουθίας χαρακτήρων

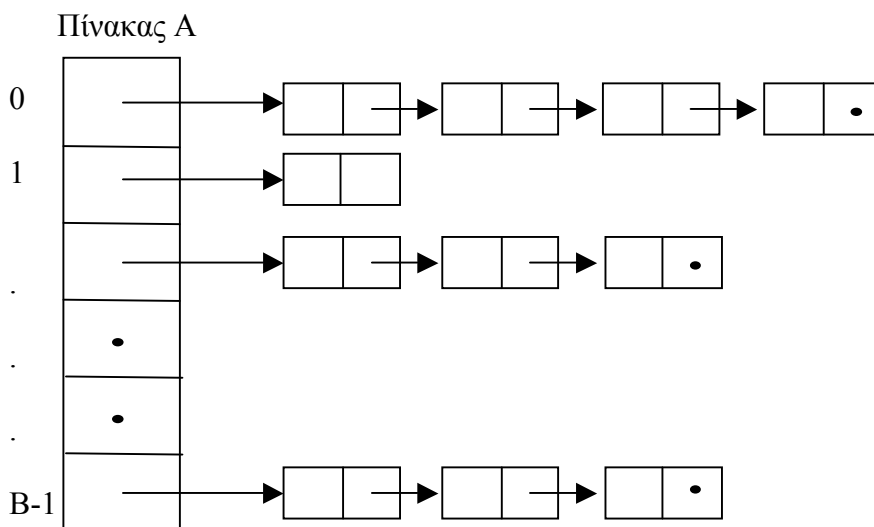
Σχεδιάστε αλγόριθμο ο οποίος θα διαβάζει μια ακολουθία από χαρακτήρες από το

πληκτρολόγιο (η ακολουθία δεν είναι απαραίτητα σε λεξικογραφική διάταξη) και θα χρησιμοποιεί μία ουρά για να τυπώσει τους χαρακτήρες αυτούς σε λεξικογραφική διάταξη.

Προσοχή: Οι αλγόριθμοι και για τα δύο ερωτήματα θα πρέπει να χρησιμοποιούν μία ουρά, καθώς και ενδεχόμενα κάποιες βοηθητικές μεταβλητές. Δεν επιτρέπεται η χρήση πινάκων ή άλλων δομών δεδομένων για την αποθήκευση ή την επεξεργασία των δεδομένων.

## Άσκηση 2 [20 μονάδες]

Θεωρείστε τον εξής αφηρημένο τύπο δεδομένων: σύνολο  $S$  με πράξεις εισαγωγή στοιχείου, διαγραφή στοιχείου και έλεγχος εάν ένα δεδομένο στοιχείο ανήκει στο σύνολο (μέλος). Αυτός ο ΑΤΔ ονομάζεται λεξικό. Μια υλοποίηση του λεξικού είναι χρησιμοποιώντας ένα μονοδιάστατο πίνακα  $A$  του οποίου κάθε στοιχείο είναι δείκτης σε μια συνδεδεμένη λίστα. Κάθε κόμβος μιας λίστας  $L$  περιέχει ένα πεδίο στο οποίο θα αποθηκεύεται ένα στοιχείο του  $S$  και ένα πεδίο στο οποίο υπάρχει ένας δείκτης που δείχνει στον επόμενο κόμβο της συνδεδεμένης λίστας  $L$  (βλ. σχήμα).



Δεδομένης μιας συνάρτησης  $h: S \rightarrow [0..B-1]$ , η εισαγωγή ενός στοιχείου  $x$  θα γίνεται στη λίστα που δείχνει ο δείκτης  $A[h(x)]$ . Συνεπώς ισχύει η εξής ιδιότητα:  $x \in S \Leftrightarrow$  το στοιχείο  $x$  βρίσκεται στη λίστα που δείχνει ο  $A[h(x)]$ .

Θεωρείστε ότι το σύνολο  $S$  περιέχει θετικούς ακέραιους και  $h(x) = x \bmod B$  με  $B = 100$ . Να γραφούν σε  $C$  (ή ψευδογλώσσα) οι κατάλληλες δηλώσεις και οι συναρτήσεις των εξής βασικών πράξεων του ΑΤΔ λεξικό:

- Δημιουργία( $S$ ): δημιουργεί ένα κενό λεξικό  $S$ .
- Εισαγωγή( $x, S$ ): εισάγει το στοιχείο  $x$  στο λεξικό  $S$  (εφόσον φυσικά  $x \notin S$ ).
- Διαγραφή( $x, S$ ): διαγράφει το στοιχείο  $x$  από το λεξικό  $S$ .
- Μέλος( $x, S$ ): επιστρέφει 1 αν  $x \in S$  και 0 σε αντίθετη περίπτωση.

### Άσκηση 3 [20 μονάδες]

Έστω οι μονοδιάστατοι πίνακες PREORDER, INORDER και POSTORDER πλήθους  $n$  οι οποίοι απεικονίζουν κάθε κόμβο ενός δυαδικού δέντρου (με  $n$  κόμβους) στην θέση του στην προδιατεταγμένη, ενδοδιατεταγμένη και μεταδιατεταγμένη ακολουθία κόμβων αντίστοιχα. Σχεδιάστε έναν αλγόριθμο που χρησιμοποιεί τους παραπάνω πίνακες για να αποφασίσει αν ο κόμβος  $i$  είναι πρόγονος του κόμβου  $j$  για οποιοδήποτε ζεύγος  $i, j$ . Εξηγήστε γιατί ο αλγόριθμός σας δουλεύει.

### Άσκηση 4 [20 μονάδες]

α) Έστω 8 στοιχεία με κλειδιά τους αριθμούς 63, 30, 36, 31, 12, 50, 35, 14. Να τα εισάγετε διαδοχικά με την παραπάνω σειρά σε ένα αρχικά άδειο δυαδικό δένδρο αναζήτησης. Στη συνέχεια να διαγράψετε τα στοιχεία με κλειδιά 30 και 63.

β) Εισάγετε τα κλειδιά 46, 24, 13, 21 και 56 στο AVL δένδρο του σχήματος.

