

HY240: Δομές Δεδομένων Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2017-18 Παναγιώτα Φατούρου

2^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 30 Οκτωβρίου 2017

Τρόπος Παράδοσης: Οι ασκήσεις μπορούν να παραδοθούν είτε σε ηλεκτρονική μορφή χρησιμοποιώντας το πρόγραμμα turnin (δείτε οδηγίες στην ιστοσελίδα του μαθήματος), είτε στους βοηθούς του μαθήματος στο εργαστήριο των μεταπτυχιακών, την Δευτέρα, 30 Οκτωβρίου 2017, ώρα 15:00-16:00. Ασκήσεις που παραδίδονται μετά τις 16:00 της Δευτέρας, 30/10/2017 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin.

Άσκηση 1 [40 μονάδες]

Έστω ότι μια βιβλιοθήκη σας παρέχει πρόσβαση σε στοίβες και ουρές χαρακτήρων. Η βιβλιοθήκη σας επιτρέπει να ορίσετε μια στοίβα (ή μια ουρά) και να καλέσετε τις 5 βασικές λειτουργίες σε αυτή. Για παράδειγμα, ο ορισμός μιας στοίβας (ή μιας ουράς) γίνεται γράφοντας: Stack S; (αντίστοιχα, Queue Q;). Για τη στοίβα υποστηρίζονται οι εξής λειτουργίες: (1) void MakeEmptyStack(Stack S), (2) boolean IsEmptyStack(Stack S), (3) Type Top(Stack S), (4) Type Pop(Stack S), (5) void Push(Stack S, char x). Αντίστοιχα, για την ουρά υποστηρίζονται οι λειτουργίες: (1) void MakeEmptyQueue(Queue Q), (2) boolean IsEmptyQueue(Queue Q), (3) char Front(Queue Q), (4) char Dequeue(Queue Q), (5) void Enqueue(Queue Q, char x).

Επιλύστε τα προβλήματα που παρουσιάζονται στη συνέχεια:

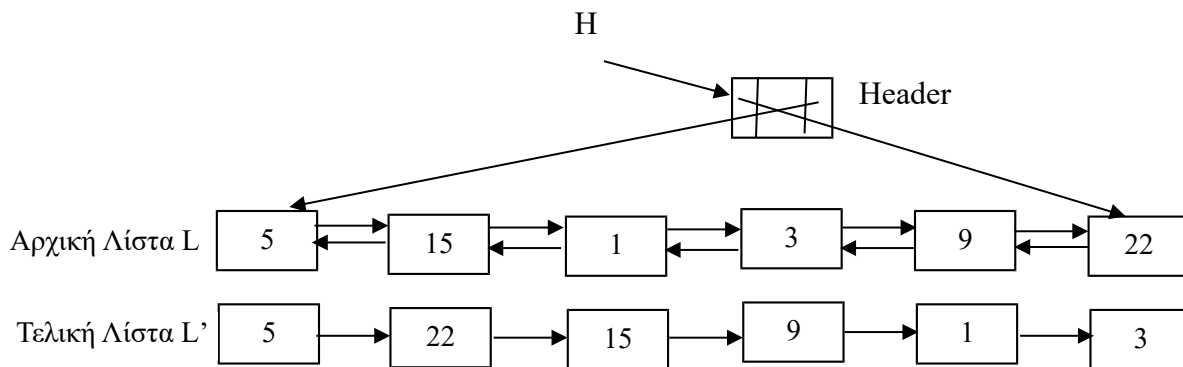
- α. Στην μεταδιατεταγμένη παράσταση, κάθε πράξη εφαρμόζεται στους δύο τελεστέους που την ακολουθούν. Για παράδειγμα, '1 2 +' αναπαριστά το άθροισμα του αριθμού 1 με τον αριθμό 2. Ο ορισμός είναι αναδρομικός και επομένως κάθε πράξη μπορεί να αναφέρεται σε τελεστέους που είναι το αποτέλεσμα άλλων πράξεων. Π.χ., '1 2 + 3 4 - * 5 -' είναι η μεταδιατεταγμένη αναπαράσταση της '(1+2)*(3-4)-5'. Παρουσιάστε αλγόριθμο, ο οποίος θα διαβάζει μια έκφραση σε μεταδιατεταγμένη μορφή και θα επιστρέφει το αποτέλεσμα του υπολογισμού της. [15M]
- β. Παρουσιάστε αλγόριθμο ο οποίος θα διαβάζει n στοιχεία από το πληκτρολόγιο και θα τα τοποθετεί σε μια στοίβα S_1 . Στη συνέχεια, ο αλγόριθμός σας θα πρέπει να ταξινομεί τα στοιχεία της S_1 χρησιμοποιώντας μία μόνο επιπρόσθετη βοηθητική στοίβα (και βοηθητικές μεταβλητές, π.χ. ακέραιους). Η ταξινόμηση θα πρέπει να πραγματοποιείται ακολουθώντας την τεχνική του αλγορίθμου InsertionSort(). Μετά το τέλος της εκτέλεσης του αλγορίθμου, το βαθύτερο στοιχείο της S_1 θα πρέπει να είναι το μικρότερο στοιχείο από αυτά που διαβάστηκαν, ενώ το κορυφαίο στοιχείο της S_1 θα πρέπει να είναι το μεγαλύτερο. [25M]

Σημειώσεις: I. Οι αλγόριθμοι και για τα δύο ερωτήματα θα πρέπει να χρησιμοποιούν ουρές και στοίβες από τη βιβλιοθήκη που σας παρέχεται, καθώς και ενδεχομένως κάποιες βοηθητικές μεταβλητές. Δεν επιτρέπεται η χρήση πινάκων ή άλλων δομών δεδομένων που δεν ανήκουν στη βιβλιοθήκη για την αποθήκευση ή την επεξεργασία των αλφαριθμητικών ή των ακολουθιών χαρακτήρων.

II. Στο ερώτημα (α), θεωρήστε πως σας παρέχεται μια βοηθητική ρουτίνα $ch = \text{getchar}()$ για την ανάγνωση αριθμών ή χαρακτήρων. Επίσης, σας παρέχονται οι βοηθητικές συναρτήσεις $\text{IsOperand}(\text{char } ch)$ και $\text{IsOperator}(\text{char } ch)$, οι οποίες επιστρέφουν TRUE αν το ch είναι τελεστής και τελεστής, αντίστοιχα. Θεωρήστε επίσης, πως όλοι οι αριθμοί κάθε μεταδιατεταγμένης παράστασης είναι μονοψήφιοι.

Άσκηση 2 [25 μονάδες]

Παρουσιάστε αλγόριθμο ο οποίος δοθείσης μιας **διπλά-συνδεδεμένης** λίστας L με n στοιχεία, θα δημιουργεί μια άλλη **απλά-συνδεδεμένη** λίστα L' με επίσης n στοιχεία για την οποία θα πρέπει να ισχύουν τα εξής. Τα στοιχεία της L' που βρίσκονται σε περιττές θέσεις ($1^\circ, 3^\circ, 5^\circ$, κλπ.) είναι αντίγραφα των πρώτων $n/2$ στοιχείων της L , ενώ τα στοιχεία της L' που βρίσκονται σε άρτιες θέσεις ($2^\circ, 4^\circ, 6^\circ$, κλπ.) είναι αντίγραφα των τελευταίων $n/2$ στοιχείων της L (από το τέλος προς την αρχή). Το n μπορεί να είναι οποιοσδήποτε άρτιος αριθμός (π.χ., 0, 2, 4, κλπ.). Ο αλγόριθμός σας θα πρέπει να εκτελείται σε χρόνο $O(n)$. Θεωρήστε ότι η διπλά-συνδεδεμένη λίστα είναι κυκλική (όπως στις διαφάνειες) και ότι δίνεται ένας δείκτης H στον κόμβο Header. Οι κόμβοι της λίστας αποτελούνται από τα πεδία num , prev και next .



Το παραπάνω σχήμα παρουσιάζει ένα παράδειγμα μιας λίστας L και της λίστας L' που πρέπει να δημιουργηθεί. Θεωρήστε πως το n δεν είναι γνωστό στον αλγόριθμο (δηλαδή αν το χρειάζεστε θα πρέπει ο αλγόριθμός που θα σχεδιάσετε να το υπολογίζει).

Άσκηση 3 [35 Μονάδες]

- α. Ένας πίνακας A με n γραμμές και n στήλες λέγεται **τριδιαγώνιος** αν για κάθε i, j , $0 \leq i, j \leq n-1$, ισχύει πως $A[i, j] = 0$, αν $i \neq j$ και $i \neq j-1$ και $i \neq j+1$. Ένα παράδειγμα τριδιαγώνιου πίνακα παρουσιάζεται στο Σχήμα 3. Περιγράψτε δομή δεδομένων για την αποθήκευση ενός τριδιαγώνιου πίνακα που θα χρησιμοποιεί χώρο $O(n)$. Παρουσιάστε ψευδοκώδικα για τη λειτουργία $\text{Access}(A, i, j)$, η οποία επιστρέφει την τιμή του στοιχείου $A[i, j]$. Ποια είναι η πολυπλοκότητα της $\text{Access}()$ που υλοποιήσατε; [20M]

12	11	0	0	0
34	43	44	0	0
0	76	23	31	0
0	0	7	62	61
0	0	0	1	32

Σχήμα 3

β. Θεωρήστε μια ειδική σκάλα η οποία συνδέει τη γη με κάποιους διαστημικούς σταθμούς και έστω ότι ένα από τα διαστημικά αθλήματα του μέλλοντος θα είναι η κατάκτηση των σκαλοπατιών της σκάλας (δηλαδή το να ανέβει ένας αθλητής όσο το δυνατόν περισσότερα σκαλοπάτια της σκάλας ξεκινώντας από τη Γη, φυσικά κουβαλώντας τον κατάλληλο διαστημικό εξοπλισμό που θα του επιτρέψει να επιβιώσει στις συνθήκες που θα συναντήσει σε κάθε φάση ανόδου της σκάλας). Με το πέρασμα του χρόνου, ο αριθμός των σκαλοπατιών που τα έχουν κατακτήσει οι αθλητές αυξάνεται (δηλαδή περιοδικά αθλητές κάνουν νέο διαστημικό ρεκόρ ανεβαίνοντας όλο και περισσότερα σκαλοπάτια). Ωστόσο, η σκάλα δεν έχει τελειωμό.

Η σκάλα έχει σε κάθε σκαλοπάτι της μια ειδική υποδοχή όπου κάθε αθλητής αφήνει μια μικρή σημαία η οποία υποδεικνύει ότι ο αθλητής έχει κατακτήσει αυτό το σκαλοπάτι της σκάλας. Σκαλοπάτια που δεν έχουν καμία σημαία δεν έχουν ακόμη κατακτηθεί, ενώ όσα έχουν κατακτηθεί φέρουν σημαία. Παρουσιάστε αλγόριθμο, ο οποίος θα βρίσκει το συνολικό αριθμό, n , των σκαλοπατιών που έχουν κατακτηθεί. Ο αλγόριθμός σας θα πρέπει να έχει χρονική πολυπλοκότητα $O(\log n)$. [15M]