

HY240: Δομές Δεδομένων

Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2018-19

Παναγιώτα Φατούρου

2^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 29 Οκτωβρίου 2018

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, 29 Οκτωβρίου 2018, ώρα 15:00-16:00. Ασκήσεις που παραδίδονται μετά τις 16:00 της Δευτέρας, 29/10/2018 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin.

Άσκηση 1 [45 μονάδες]

α. Έστω ότι μια βιβλιοθήκη σας παρέχει πρόσβαση σε στοιβες χαρακτήρων. Η βιβλιοθήκη σας επιτρέπει να ορίσετε μια στοιβία και να καλέσετε τις 5 βασικές λειτουργίες σε αυτή. Για παράδειγμα, ο ορισμός μιας στοιβίας γίνεται γράφοντας: `Stack S;`. Για τη στοιβία 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)`.

Παρουσιάστε ψευδοκώδικα για έναν αλγόριθμο που θα διαβάζει n αριθμούς από το πληκτρολόγιο και θα τους ταξινομεί, εφαρμόζοντας την τεχνική της `InsertionSort()` που έχετε διδαχθεί στο μάθημα, χρησιμοποιώντας δύο στοιβες. Κατά την περάτωση του αλγορίθμου, οι αριθμοί που αναγνώσθηκαν θα πρέπει να έχουν εκτυπωθεί, σε αύξουσα διάταξη, στην οθόνη, ενώ οι δύο στοιβες που χρησιμοποίησε ο αλγόριθμος, θα πρέπει να είναι άδειες. [15M]

Σημειώσεις: Ο αλγόριθμος θα πρέπει να χρησιμοποιεί μόνο δύο στοιβες από τη βιβλιοθήκη που 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 αριθμών της εισόδου. Χρησιμοποιήστε την `scanf()` για την ανάγνωση των αριθμών αυτών.

β. Θεωρήστε μια ταξινομημένη κατ' αύξουσα διάταξη απλά-συνδεδεμένη λίστα L με n στοιχεία. [30M]

i. Ζητείται ψευδοκώδικας για έναν αλγόριθμο που θα τυπώνει τα περιεχόμενα των κόμβων σε φθίνουσα διάταξη (δηλαδή θα τυπώνει τα στοιχεία της L από το τελευταίο προς το πρώτο) **και θα έχει χρονική πολυπλοκότητα $\Theta(n^2)$** . [15M]

ii. Ζητείται ψευδοκώδικας για έναν αλγόριθμο που θα τυπώνει τα περιεχόμενα των κόμβων σε φθίνουσα διάταξη (δηλαδή θα τυπώνει τα στοιχεία της L από το τελευταίο προς το πρώτο) **και θα έχει χρονική πολυπλοκότητα $\Theta(n)$** . [15M]

Επιχειρηματολογήστε για την χρονική πολυπλοκότητα των αλγορίθμων που σχεδιάσατε.

Άσκηση 2 [30 μονάδες]

Θεωρήστε δύο απλά-συνδεδεμένες λίστες $L1$ και $L2$. Έστω ότι κάθε στοιχείο των λιστών $L1$ και $L2$, αποθηκεύει έναν ακέραιο num , μια boolean μεταβλητή f και τον δείκτη $next$ στο επόμενο στοιχείο της εκάστοτε λίστας. Οι λίστες $L1$ και $L2$ είναι ταξινομημένες ως προς το πεδίο num . Ζητείται ψευδοκώδικας για ένα n αλγόριθμο που θα δημιουργεί μια νέα απλά-συνδεδεμένη λίστα L , κάθε στοιχείο της οποίας θα περιέχει δύο ακεραίους $num1$ και $num2$, καθώς και τον δείκτη $next$ προς το επόμενο στοιχείο της λίστας. Θεωρήστε ότι σε κάθε μια από τις λίστες $L1$ και $L2$, το num πεδίο κάθε κόμβου είναι μοναδικό και ότι η $L1$ έχει ίσο αριθμό στοιχείων με την $L2$, έστω n . Επίσης, θεωρήστε ότι ο αριθμός (έστω $n1$) των στοιχείων που έχουν την τιμή 0 στο πεδίο f είναι ίδιος και στις δύο λίστες και ότι το ίδιο ισχύει και για τον αριθμό των στοιχείων που έχουν την τιμή 1 στο πεδίο f (έστω ότι ο αριθμός αυτός είναι $n2$).

Η λίστα L θα πρέπει να φτιαχτεί έτσι ώστε αντίγραφα των $n1$ στοιχείων της $L1$ που έχουν τιμή ίση με 0 στο πεδίο f , να εμφανίζονται πρώτα στη λίστα L , με τη σειρά που εμφανίζονται και στην $L1$ (δηλαδή ταξινομημένα ως προς το πεδίο num). Στη συνέχεια, θα πρέπει να εμφανίζονται αντίγραφα των $n2$ στοιχείων της $L1$ που έχουν τιμή ίση με 1 στο πεδίο f , με τη σειρά που εμφανίζονται στην $L1$. Επομένως, η L αποτελείται από δύο μέρη, $P1$ και $P2$, μεγέθους $n1$ και $n2$, αντίστοιχα, κάθε ένα εκ των οποίων είναι ταξινομημένα ως προς το πεδίο num . Επιπρόσθετα, το i -οστό στοιχείο στο μέρος $P1$ της L θα πρέπει να έχει στο πεδίο $num2$ τιμή ίση με την τιμή του πεδίου num του i -οστού στοιχείου της $L2$ που περιέχει 0 στο πεδίο f . Παρομοίως, το j -οστό στοιχείο στο μέρος $P2$ της L θα πρέπει να έχει στο πεδίο $num2$ τιμή ίση με την τιμή του πεδίου num του j -οστού στοιχείου της $L2$ που περιέχει την τιμή 1 στο πεδίο f .

Παρουσιάστε ψευδοκώδικα για τον αλγόριθμο που θα δημιουργεί την L . **Ο αλγόριθμος θα πρέπει να έχει χρονική πολυπλοκότητα $O(n)$.** Τα n , $n1$ και $n2$ δεν είναι γνωστά στον αλγόριθμο που θα σχεδιάσετε. Ο αλγόριθμος θα πρέπει να παίρνει σαν ορίσματα δύο δείκτες, έναν στο πρώτο στοιχείο της $L1$ και έναν στο πρώτο στοιχείο της $L2$ και να επιστρέφει ένα δείκτη στο πρώτο στοιχείο της L . Ο αλγόριθμος δεν θα πρέπει να προκαλεί καμία αλλαγή στη σύνθεση των λιστών $L1$ και $L2$.

Παράδειγμα: Θεωρήστε τις λίστες:

$L1 = \langle 5,1 \rangle \rightarrow \langle 7,0 \rangle \rightarrow \langle 9,0 \rangle \rightarrow \langle 20,1 \rangle \rightarrow \langle 21,1 \rangle \rightarrow \langle 25,0 \rangle \rightarrow \langle 11,1 \rangle \rightarrow \langle 13,1 \rangle \rightarrow \langle 15,0 \rangle \rightarrow \langle 17,1 \rangle$

$L2 = \langle 2,0 \rangle \rightarrow \langle 4,1 \rangle \rightarrow \langle 9,0 \rangle \rightarrow \langle 10,0 \rangle \rightarrow \langle 20,1 \rangle \rightarrow \langle 12,1 \rangle \rightarrow \langle 14,0 \rangle \rightarrow \langle 14,1 \rangle \rightarrow \langle 16,1 \rangle \rightarrow \langle 18,1 \rangle$

Μετά την εκτέλεση του αλγορίθμου σας, η λίστα L θα πρέπει να έχει την εξής μορφή:

$L = \langle 7,2 \rangle \rightarrow \langle 9,9 \rangle \rightarrow \langle 25,10 \rangle \rightarrow \langle 15,14 \rangle \rightarrow \langle 5,4 \rangle \rightarrow \langle 20,20 \rangle \rightarrow \langle 21,12 \rangle \rightarrow \langle 11,14 \rangle \rightarrow \langle 13,16 \rangle \rightarrow \langle 17,18 \rangle$

Υπόδειξη: Εκτελέστε ταυτόχρονα μια διάσχιση στην $L1$ και μια ή περισσότερες διασχίσεις στην $L2$. Χρησιμοποιήστε βοηθητικούς δείκτες για να θυμάστε το τελευταίο στοιχείο της $L2$ με τιμή 0 στο πεδίο $f2$ που έχετε μεταφέρει στην L , καθώς και το τελευταίο στοιχείο της $L2$ με τιμή 1 στο πεδίο $f2$ που έχετε μεταφέρει στην L . Η λίστα L θα πρέπει να δημιουργείται καθώς θα εκτελούνται οι διασχίσεις των $L1$, $L2$ και θα πρέπει να χρησιμοποιεί βοηθητικούς δείκτες που θα επιτρέπουν να γίνονται εισαγωγές, σε $O(1)$ χρόνο, στα κατάλληλα σημεία της L (ώστε τελικά να ισχύουν όσα περιγράφηκαν παραπάνω). Επιτρέπεται να χρησιμοποιήσετε βοηθητικές συναρτήσεις και ρουτίνες, αν θεωρείτε πως απλοποιούν την σχεδίαση του αλγορίθμου, αλλά θα πρέπει να παρουσιάσετε ψευδοκώδικα για όλες αυτές τις ρουτίνες/συναρτήσεις.

Άσκηση 3 [25 Μονάδες]

Θεωρείστε μια διπλά-συνδεδεμένη λίστα DL που είναι κυκλική (δηλαδή έχει κόμβο header, όπως παρουσιάστηκε στις διαφάνειες του μαθήματος) και έχει περιττό αριθμό στοιχείων. Έστω ότι σε κάθε στοιχείο της λίστας DL αποθηκεύονται, εκτός από τους δείκτες $prev$ και $next$, ένα πεδίο key που είναι μοναδικό και ένας ακέραιος num . Ζητείται ψευδοκώδικας για έναν αλγόριθμο που θα παίρνει ως ορίσματα, έναν δείκτη H στον κόμβο header της DL και έναν ακέραιο N . Ο αλγόριθμος θα πρέπει να δημιουργεί μια απλά-συνδεδεμένη λίστα L με τον ακόλουθο τρόπο. Ο αλγόριθμος θα πρέπει πρώτα να βρίσκει το μεσαίο στοιχείο της DL και να εισάγει ένα αντίγραφο αυτού στην L . Στη συνέχεια, για N φορές, ο αλγόριθμος θα πρέπει να εκτελεί μια μερική διάσχιση της λίστας ως εξής. Σε μια μερική

διάσχιση, ο αλγόριθμος θα πρέπει να διαβάσει το πεδίο `num` του τρέχοντος κόμβου και στη συνέχεια, να διασχίζει τόσες θέσεις της λίστας `DL` προς τα αριστερά (ή προς τα δεξιά) όσες και η τιμή αυτού του πεδίου. Μετά από κάθε μερική διάσχιση, η φορά αλλάζει και η επόμενη μερική διάσχιση θα πρέπει να είναι προς την αντίθετη κατεύθυνση από αυτήν της τρέχουσας μερικής διάσχισης. Μετά από κάθε μερική διάσχιση, ο αλγόριθμος εισάγει στην `L` τον τρέχοντα κόμβο (δηλαδή τον κόμβο στον οποίο έχει φτάσει μετά την εκτέλεση της τρέχουσας μερικής διάσχισης). Επομένως, με το τέλος του αλγορίθμου, η `L` θα πρέπει να περιέχει $N + 1$ κόμβους. Κατά την κίνηση προς τα αριστερά (δεξιά, αντίστοιχα), αν ο αλγόριθμος φτάσει στην αρχή (στο τέλος, αντίστοιχα) της λίστας, συνεχίζει από το τέλος (την αρχή, αντίστοιχα), μέσω του κόμβου `header` (δηλαδή χρησιμοποιώντας το γεγονός ότι η λίστα είναι κυκλική).

Θεωρήστε ότι το πεδίο `num` κάθε κόμβου δεν μπορεί να είναι μεγαλύτερο από M . Ποια είναι η πολυπλοκότητα του αλγορίθμου σας και γιατί;

Παράδειγμα: Θεωρήστε την εξής λίστα:

$DL = \langle 1,3 \rangle \leftrightarrow \langle 2,6 \rangle \leftrightarrow \langle 9,2 \rangle \leftrightarrow \langle 7,5 \rangle \leftrightarrow \langle 3,1 \rangle \leftrightarrow \langle 8,2 \rangle \leftrightarrow \langle 6,4 \rangle \leftrightarrow \langle 10,8 \rangle \leftrightarrow \langle 5,20 \rangle$

και έστω ότι $N = 10$ και αρχικά η κατεύθυνση είναι προς τα αριστερά. Τότε η λίστα `L` θα έχει την ακόλουθη μορφή:

$L = 3 \rightarrow 7 \rightarrow 5 \rightarrow 6 \rightarrow 2 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 5 \rightarrow 6$