

HY240: Δομές Δεδομένων

Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2012-13

Παναγιώτα Φατούρου

2^ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 29 Οκτωβρίου 2012

Τρόπος Παράδοσης: Οι ασκήσεις μπορούν να παραδοθούν είτε σε ηλεκτρονική μορφή χρησιμοποιώντας το πρόγραμμα submit (δείτε οδηγίες στην ιστοσελίδα του μαθήματος), είτε στους βοηθούς του μαθήματος στο εργαστήριο των μεταπτυχιακών στο υπόγειο των λευκών κτιρίων, τη Δευτέρα, 29 Οκτωβρίου 2012, ώρα 10:00-11:00. Ασκήσεις που παραδίδονται μετά τις 11:00 της Δευτέρας, 29/10/2012 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα submit. Εναλλακτικά, εκπρόθεσμες ασκήσεις μπορούν να παραδοθούν στη διδάσκουσα του μαθήματος ή να σταλούν στο hy240a@csd.uoc.gr μέσω ηλεκτρονικού ταχυδρομείου.

ΠΡΟΣΟΧΗ: Παραδίδουμε τις ασκήσεις στο βοηθό του μαθήματος ή στη διδάσκουσα ή χρησιμοποιούμε το πρόγραμμα submit. Δεν τοποθετούμε τις ασκήσεις σε κουτιά που βρίσκονται στο υπόγειο των λευκών κτιρίων (ακόμη και αν σε αυτά αναγράφονται πληροφορίες που φαίνονται σχετικές με το μάθημα και την τρέχουσα σειρά ασκήσεων), αφού αυτό δεν εξασφαλίζει την ασφάλεια τους. Ασκήσεις που θα βρεθούν σε κουτιά δεν θα γίνουν δεκτές.

Άσκηση 1 [30 μονάδες]

1. Παρουσιάστε **μη-αναδρομικό** αλγόριθμο ο οποίος θα παίρνει ως παράμετρο έναν ακέραιο αριθμό και θα τυπώνει τη δυαδική αναπαράσταση του αριθμού αυτού. Ο αλγόριθμός σας επιτρέπεται να χρησιμοποιεί μια βοηθητική στοίβα ή ουρά. [15%]
2. Παρουσιάστε αλγόριθμο ο οποίος, χρησιμοποιώντας στοίβες ή/και ουρές θα ελέγχει αν μια έκφραση που περιέχει παρενθέσεις, αγκύλες και άγκιστρα είναι σωστή. [15%]

Παραδείγματα: Η έκφραση $\{ \{ (([]) []) \{ [] \} \} \}$ είναι σωστή, ενώ οι εκφράσεις $\{ (([]) []) \}$ και $\{ (([]) []) \}$ δεν είναι.

Παρατήρηση: Θεωρήστε ότι σας παρέχεται μια βιβλιοθήκη στοιβών και ουρών. 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, Type x). Αντίστοιχα, για την ουρά υποστηρίζονται οι λειτουργίες: (1) void MakeEmptyQueue(Queue Q), (2) boolean IsEmptyQueue(Queue Q), (3) Type Front(Queue Q), (4) Type Dequeue(Queue Q), (5) void Enqueue(Queue Q, Type x).

Άσκηση 2 [35 μονάδες]

1. Δείξτε πως μπορεί να υλοποιηθεί μια ουρά χρησιμοποιώντας δύο στοίβες. Αναλύστε τη χρονική πολυπλοκότητα των λειτουργιών της ουράς στην υλοποίηση που θα προτείνετε.
2. Παρουσιάστε δύο αλγόριθμους οι οποίοι δεδομένης μιας ταξινομημένης **κατ' αύξουσα διάταξη** λίστας L με n στοιχεία θα τυπώνουν τα περιεχόμενα των κόμβων **σε φθίνουσα διάταξη** (δηλαδή θα τυπώνουν τα στοιχεία από το τελευταίο προς το πρώτο). Θεωρήστε πως η λίστα είναι απλά συνδεδεμένη.

i. Ο πρώτος αλγόριθμος θα πρέπει να έχει χρονική πολυπλοκότητα $\Theta(n^2)$. [12%]

ii. Ο δεύτερος αλγόριθμος θα πρέπει να έχει χρονική πολυπλοκότητα $\Theta(n)$. [23%]

Επιχειρηματολογήστε για την χρονική πολυπλοκότητα των αλγορίθμων που σχεδιάσατε.

Άσκηση 3 [35 μονάδες]

Ένας συγχωνεύσιμος σωρός είναι ένας αφηρημένος τύπος δεδομένων που υποστηρίζει τις εξής λειτουργίες:

- Κατασκευή κενού σωρού, Set MakeEmptySet(): δημιουργεί έναν κενό συγχωνεύσιμο σωρό.
- Εισαγωγή, Set SetInsert(Set S, Type x): εισάγει το στοιχείο x στο σύνολο S και επιστρέφει το σύνολο S.
- Εύρεση ελάχιστου στοιχείου, Type SetFindMin(Set S): Επιστρέφει το ελάχιστο στοιχείο του συνόλου S.
- Διαγραφή ελάχιστου στοιχείου, Set SetDeleteMin(): Διαγράφει το ελάχιστο στοιχείο του συνόλου S.
- Συνένωση, Set SetMerge(Set S1, Set S2): Συνενώνει τα σύνολα S1 και S2 δημιουργώντας ένα σύνολο S3 που περιέχει τα στοιχεία του $S1 \cup S2$. Επιστρέφει το S3.

Παρουσιάστε υλοποιήσεις ενός συγχωνεύσιμου σωρού χρησιμοποιώντας λίστες των εξής τύπων:

i. Ταξινομημένες, απλά συνδεδεμένες. [9]

ii. Μη-ταξινομημένες, απλά συνδεδεμένες. [8]

iii. Μη-ταξινομημένες, απλά συνδεδεμένες και τα προς συγχώνευση δυναμικά σύνολα είναι ξένα. [9]

Οι λειτουργίες θα πρέπει να υλοποιηθούν έτσι ώστε να επιτυγχάνεται ο καλύτερος δυνατός χρόνος εκτέλεσης.

iv. Αναλύστε το χρόνο εκτέλεσης της κάθε λειτουργίας συναρτήσει του μεγέθους του συνόλου επί του οποίου εκτελείται (δηλαδή συναρτήσει του πλήθους n των στοιχείων που είναι αποθηκευμένα στο συγχωνεύσιμο σωρό) ή των μεγεθών των συνόλων αυτών εφόσον είναι περισσότερα από ένα. [9]