

HY240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2008-09
Διδάσκουσα: Παναγιώτα Φατούρου
3^η Σειρά Θεωρητικών Ασκήσεων

Ημερομηνία Παράδοσης: Δευτέρα, 23 Νοεμβρίου 2009, ώρα 19:00-20:00

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 τη Δευτέρα, 23 Νοεμβρίου 2009 (ώρα 19:00-20:00). Οι φοιτητές μπορούν να παραδίδουν τις ασκήσεις τους και νωρίτερα από αυτήν την ημερομηνία αλλά δικαιολογίες όπως «ήρθα την τάδε μέρα και δεν ήσασταν στο γραφείο σας» δεν θα γίνουν δεκτές. Όσοι από τους φοιτητές δεν έχουν καταφέρει να παραδώσουν τις ασκήσεις τους πριν τις 23/11 θα πρέπει να τις παραδώσουν στους βοηθούς του μαθήματος την προκαθορισμένη μέρα και ώρα. Εκπρόθεσμες ασκήσεις γίνονται δεκτές (με μείωση βαθμολογίας σύμφωνα με τον αλγόριθμο που περιγράφεται στην ιστοσελίδα του μαθήματος) σε ηλεκτρονική μορφή και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα submit. Εναλλακτικά, εκπρόθεσμες ασκήσεις μπορούν να παραδοθούν στη διδάσκουσα του μαθήματος.

ΠΡΟΣΟΧΗ: Παραδίδουμε τις ασκήσεις στο βοηθό του μαθήματος ή στη διδάσκουσα ή χρησιμοποιούμε το πρόγραμμα submit. Δεν τοποθετούμε σε καμία περίπτωση 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.

ΑΣΚΗΣΕΙΣ

- Δίνεται η ακόλουθη υλοποίηση για αραιούς δισδιάστατους πίνακες. Κάθε αραιός δισδιάστατος πίνακας $A[1..n, 1..m]$ υλοποιείται με έναν boolean δισδιάστατο πίνακα $B[1..n, 1..m]$ (κάθε στοιχείο του B έχει την τιμή 0 ή 1) και μια δυναμική συνδεδεμένη λίστα που περιέχει τα μη-μηδενικά στοιχεία του πίνακα A . Για τον πίνακα B ισχύουν τα ακόλουθα. Για κάθε i, j , με $1 \leq i \leq n, 1 \leq j \leq m$, $B[i][j] = 1$ αν και μόνο αν $A[i][j] \neq 0$, και $B[i][j] = 0$ αν και μόνο αν $A[i][j] = 0$. Τα μη-μηδενικά στοιχεία στη λίστα αποθηκεύονται ως εξής: «τα μη-μηδενικά στοιχεία οποιασδήποτε γραμμής i απαντώνται στη λίστα πριν από τα μη-μηδενικά στοιχεία οποιαδήποτε γραμμής $j > i$ και τα μη-μηδενικά στοιχεία οποιασδήποτε στήλης k απαντώνται στη λίστα πριν από τα μη-μηδενικά στοιχεία οποιασδήποτε στήλης $m > k$ ». Για παράδειγμα, δίνεται ο αραιός πίνακας A :

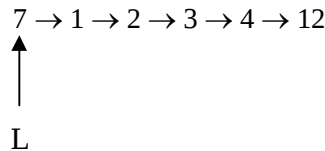
0	7	0	0	0
1	2	0	0	3
0	0	4	0	0
12	0	0	0	0

Ο πίνακας B που αντιστοιχεί στον A είναι ο ακόλουθος:

0	1	0	0	0
---	---	---	---	---

1	1	0	0	1
0	0	1	0	0
1	0	0	0	0

και η λίστα έχει την εξής μορφή:



- α. Παρουσιάστε αλγόριθμους που θα υλοποιούν τις ακόλουθες λειτουργίες στον αραιό πίνακα:
- Insert(int x, int i, int j):** εισάγει το στοιχείο x στη i-οστή γραμμή και τη j-οστή στήλη του αραιού πίνακα.
 - Delete(int x):** αναζητά στοιχείο με τιμή x στον αραιό πίνακα και αν υπάρχει το διαγράφει.
 - AddArrays(sparse_table A1, A2, A3):** προσθέτει τους αραιούς πίνακες A1 και A2, δηλαδή δημιουργεί έναν νέο αραιό πίνακα A3 για τον οποίο ισχύει ότι $A3[i][j] = A1[i][j] + A2[i][j]$, $\forall i, j$ με $1 \leq i \leq n$ και $1 \leq j \leq m$. Θεωρούμε πως όλοι οι αραιοί πίνακες, A1, A2 και A3, υλοποιούνται με τον τρόπο που περιγράφηκε πιο πάνω.
- β. Μελετήστε την χρονική πολυπλοκότητα των αλγορίθμων που παρουσιάσατε.
2. α. Γράψτε μια αναδρομική συνάρτηση που θα παίρνει ως παράμετρο ένα δείκτη στη ρίζα ενός δυαδικού δένδρου και θα επιστρέφει το βαθμό του διατεταγμένου δένδρου (δηλαδή το μέγιστο πλήθος παιδιών μεταξύ των κόμβων του δένδρου). Ποια είναι η χρονική πολυπλοκότητα του αλγορίθμου σας και γιατί;
- β. Να γραφεί μια συνάρτηση που θα παίρνει ως παράμετρο έναν δείκτη στη ρίζα ενός δυαδικού δένδρου και θα εξετάζει αν το δένδρο είναι πλήρες ή όχι.
- γ. Γράψτε μια αναδρομική συνάρτηση που θα παίρνει ως παράμετρο έναν δείκτη στη ρίζα ενός δυαδικού δένδρου **που αναπαριστά ένα (όχι απαραίτητα δυαδικό) διατεταγμένο δένδρο** και θα επιστρέφει το βαθμό του διατεταγμένου δένδρου (δηλαδή το μέγιστο πλήθος παιδιών μεταξύ των κόμβων του). Αν ο βαθμός του διατεταγμένου δένδρου είναι k, η συνάρτηση πρέπει να εκτελείται σε $O(nk)$ χρόνο, όπου n είναι ο συνολικός αριθμός κόμβων στο δένδρο. Εξηγήστε ποιες είναι οι διαφορές του αλγορίθμου που σχεδιάσατε σε αυτό το ερώτημα από τον αλγόριθμο που σχεδιάσατε στο ερώτημα α., καθώς και γιατί αυτές οι αλλαγές είναι απαραίτητο να γίνουν εδώ. Επιχειρηματολογήστε για τη χρονική πολυπλοκότητα του αλγορίθμου σας.
3. α. Έστω η ακολουθία Σ των γραμμάτων που απαρτίζουν το ονοματεπώνυμό σας. Για παράδειγμα, η ακολουθία αυτή για τη διδάσκουσα του μαθήματος θα ήταν $\Sigma = \{\Pi, A, N, A, \Gamma, I, \Omega, T, A, \Phi, A, T, O, Y, P, O, Y\}$. Παρουσιάστε ένα πλήρες δυαδικό δένδρο και

τοποθετήστε με τρεις τρόπους στους κόμβους του τα γράμματα της ακολουθίας του ονόματός σας, έτσι ώστε το ονοματεπώνυμό σας να τυπώνεται σωστά κατά την εκτέλεση των εξής διασχίσεων (όπου η διαδικασία `Visit()` είναι μια απλή `printf()` του χαρακτήρα που είναι αποθηκευμένος στον εκάστοτε κόμβο του δένδρου):

- (i) ενδο-διατεταγμένη διάσχιση
 - (ii) προ-διατεταγμένη διάσχιση
 - (iii) μετα-διατεταγμένη διάσχιση
- β. Περιγράψτε αλγόριθμο, ο οποίος θα παίρνει ως όρισμα έναν δείκτη σε έναν κόμβο ενός διπλά-συνδεδεμένου δυαδικού δένδρου και θα επιστρέφει τον προηγούμενο του κόμβου στην προδιατεταγμένη διάσχιση.
- γ. Παρουσιάστε μη αναδρομικό αλγόριθμο που θα εκτελεί μετα-διατεταγμένη διάσχιση σε ένα απλό δυαδικό δένδρο. Ο αλγόριθμός σας επιτρέπεται να χρησιμοποιεί μια επιπρόσθετη δομή στοίβας για να πραγματοποιήσει τη διάσχιση.