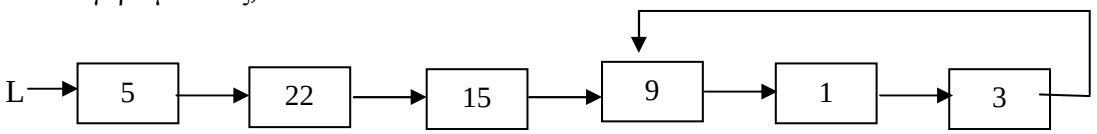
HY240: Δομές Δεδομένων
Εαρινό Εξάμηνο – Ακαδημαϊκό Έτος 2018-19
Παναγιώτα Φατούρου

2ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Τρίτη, 12 Μαρτίου 2019
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 Τρίτη, 12 Μαρτίου 2019, ώρα 13:00-14:00. Ασκήσεις που παραδίδονται μετά τις 14:00 της Τρίτης, 12/3/2019 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή (pdf) και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin. Εναλλακτικά, εκπρόθεσμες ασκήσεις μπορούν να παραδοθούν στη διδάσκουσα.

Άσκηση 1 [50 μονάδες]

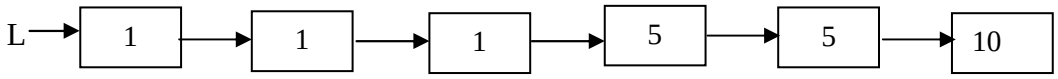
α. Παρουσιάστε ψευδοκώδικα για έναν αλγόριθμο, ο οποίος θα εξετάζει αν υπάρχει κύκλος σε μια απλά-συνδεδεμένη λίστα L. Σημειώνεται ότι στον κύκλο δεν είναι απαραίτητο να συμμετέχουν όλα τα στοιχεία της L. Για παράδειγμα, η λίστα του Σχήματος 1 περιέχει κύκλο. Ποια είναι η πολυπλοκότητα του αλγορίθμου σας; [20%]



Σχήμα 1

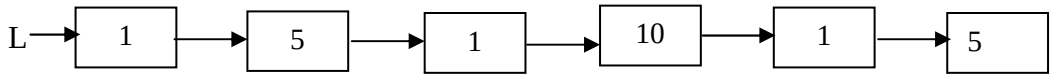
β. Παρουσιάστε ψευδοκώδικα για έναν αλγόριθμο, ο οποίος θα διαγράφει πολλαπλές εγγραφές του ίδιου στοιχείου από μια απλά-συνδεδεμένη ταξινομημένη λίστα. **Η πολυπλοκότητα του αλγορίθμου που θα σχεδιάσετε θα πρέπει να είναι O(n)**, όπου n είναι ο αριθμός των στοιχείων στη λίστα.

Για παράδειγμα, η λίστα του Σχήματος 2, περιέχει τρεις εγγραφές του στοιχείου με κλειδί 1, και δύο εγγραφές του στοιχείου με κλειδί 5. Ο αλγόριθμος που θα σχεδιάσετε θα πρέπει να διαγράφει τις δύο από τις τρείς εγγραφές του στοιχείου 1 και την μια από τις δύο εγγραφές του στοιχείου 2. Επομένως, η λίστα θα περιέχει τελικά τρία στοιχεία, 1, 5 και 10. [15%]



Σχήμα 2

γ. Παρουσιάστε ψευδοκώδικα για έναν αλγόριθμο, ο οποίος θα διαγράφει πολλαπλές εγγραφές του ίδιου στοιχείου από μια απλά-συνδεδεμένη μη-ταξινομημένη λίστα. Ποια είναι η πολυπλοκότητα του αλγορίθμου σας; [15%]



Σχήμα 3

Για παράδειγμα, όταν ο αλγόριθμος που θα σχεδιάσετε θα εφαρμόζεται στη λίστα του Σχήματος 3, θα πρέπει να διαγράφει δύο από τις τρεις εγγραφές του στοιχείου με κλειδί 1 και μία από τις δύο εγγραφές του στοιχείου με κλειδί 5.

Άσκηση 2 [25 μονάδες]

Έστω πως μια λίστα περιέχει στοιχεία του ακόλουθου τύπου:

```
struct node {
    int pid;
    int val;
    struct node *next;
}
```

Το σύνολο τιμών του πεδίου pid κάθε στοιχείου της λίστας είναι το {0, 1, 2}. Η λίστα είναι ταξινομημένη σε αύξουσα διάταξη, βάσει του πεδίου val κάθε στοιχείου της. Ζητείται αλγόριθμος που δοθείσης της λίστας L, θα κατασκευάζει τρεις άλλες λίστες L0, L1, L2, με στοιχεία ίδιου τύπου όπως η L, κάθε μια εκ των οποίων θα είναι επίσης ταξινομημένη σε αύξουσα διάταξη, βάσει του πεδίου val κάθε στοιχείου της. Για κάθε $j \in \{0, 1, 2\}$, η λίστα Lj θα πρέπει να περιέχει όλα τα στοιχεία της L των οποίων το πεδίο pid ισούται με j. Ο αλγόριθμος που θα σχεδιάσετε δεν θα πρέπει να προκαλεί κάποια αλλοίωση στην L. **Ο αλγόριθμός σας θα πρέπει να έχει πολυπλοκότητα O(n), όπου n είναι το πλήθος των στοιχείων της L** (πιο συγκεκριμένα, μια μόνο διάσχιση της L θα πρέπει να είναι αρκετή ώστε ο αλγόριθμος να δημιουργεί και τις τρεις νέες λίστες).

Άσκηση 3 [25 Μονάδες]

Θεωρήστε τη λειτουργία Subset(L_1, L_2) (ισότητα συνόλων), όπου S_1 και S_2 είναι δύο σύνολα. Παρουσιάστε αλγόριθμο που να υλοποιεί την Subset() δεδομένου ότι τα σύνολα υλοποιούνται με **ταξινομημένες απλά-συνδεδεμένες λίστες.** Η Subset() πρέπει να παίρνει ως όρισμα ένα δείκτη στο πρώτο στοιχείο της λίστας που υλοποιεί το σύνολο L_1 και ένα δείκτη στο πρώτο στοιχείο της λίστας που υλοποιεί το L_2. Θα πρέπει να επιστρέφει TRUE αν το σύνολο που αναπαριστά η L1 είναι υποσύνολο του συνόλου που αναπαριστά η L2 και FALSE διαφορετικά. Οι λίστες που αναπαριστούν τα L1 και L2 θα πρέπει να παραμένουν αναλλοίωτες κατά την εκτέλεση του αλγορίθμου σας.

Ο αλγόριθμος σας θα πρέπει να έχει πολυπλοκότητα $O(n_1+ n_2)$, όπου n_1 και n_2 είναι το πλήθος στοιχείων των συνόλων L_1 και L_2, αντίστοιχα. Εξηγήστε γιατί ο αλγόριθμός σας επιτυγχάνει την πολυπλοκότητα αυτή.