

**HY240: Δομές Δεδομένων**  
**Εαρινό Εξάμηνο – Ακαδημαϊκό Έτος 2015-16**  
**Παναγιώτα Φατούρου**

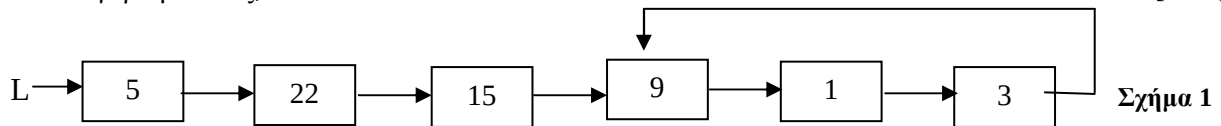
**2<sup>ο</sup> Σετ Ασκήσεων**

**Ημερομηνία Παράδοσης:** Τρίτη, 15 Μαρτίου 2016

**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, 15 Μαρτίου 2016, ώρα 11:00-12:00. Ασκήσεις που παραδίδονται μετά τις 12:00 της Τρίτης, 15/3/2016 θεωρούνται εκπρόθεσμες. Εκπρόθεσμες ασκήσεις γίνονται δεκτές σε ηλεκτρονική μορφή και η αποστολή τους πρέπει να γίνει χρησιμοποιώντας το πρόγραμμα turnin..

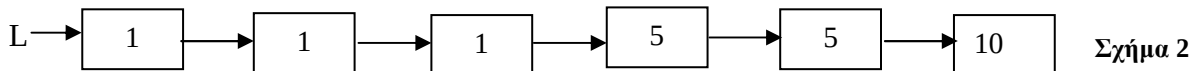
**Άσκηση 1 [50 μονάδες]**

- 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%]

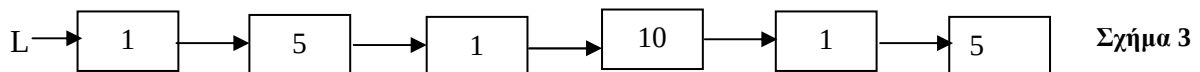


- β. Παρουσιάστε ψευδοκώδικα για έναν αλγόριθμο, ο οποίος θα διαγράφει πολλαπλές εγγραφές του ίδιου στοιχείου από μια απλά-συνδεδεμένη ταξινομημένη λίστα. **Η πολυπλοκότητα του αλγορίθμου που θα σχεδιάσετε θα πρέπει να είναι  $O(n)$** , όπου  $n$  είναι ο αριθμός των στοιχείων στη λίστα.

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 τρεις 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%]



- 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%]



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 απλά-συνδεδεμένη λίστα περιέχει στοιχεία του ακόλουθου τύπου:

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

Προσαρμόστε τον ακόλουθο αλγόριθμο ταξινόμησης, ο οποίος λειτουργεί σε πίνακα, ώστε αυτός να μπορεί να εφαρμοστεί σε μια απλά-συνδεδεμένη λίστα με στοιχεία του τύπου που περιγράφηκαν παραπάνω. Ο αλγόριθμος που θα σχεδιάσετε θα πρέπει να ταξινομεί τη λίστα βάσει του πεδίου pid των κόμβων.

Ο αριθμός,  $n$ , των στοιχείων της λίστας δεν είναι γνωστός. Επομένως, οι βρόγχοι for της SelectionSort() του Σχήματος 4 θα πρέπει να αντικατασταθούν με βρόγχους που πραγματοποιούν διασχίσεις της λίστας (ή μέρους αυτής) βάσει δεικτών. Μη χρησιμοποιήσετε ονόματα όπως  $i, j, k$  για μεταβλητές τύπου δείκτη.

```
Procedure SelectionSort(table A[0..n-1]) {  
    int i, j, k;  
  
    for (i = 0; i < n-1; i++) {  
        j = i;  
        for (k = i+1; k < n; k++)  
            if (A[k] < A[j]) j = k;  
        swap(A[i], A[j]);  
    }  
}
```

Σχήμα 4

## Άσκηση 3 [25 Μονάδες]

Θεωρήστε τη λειτουργία Equal( $L_1, L_2$ ) (ισότητα συνόλων), όπου  $S_1$  και  $S_2$  είναι δύο σύνολα. Παρουσιάστε αλγόριθμο που να υλοποιεί την Equal() δεδομένου ότι τα σύνολα υλοποιούνται με **ταξινομημένες απλά-συνδεδεμένες λίστες**. Η Equal() 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 οι λίστες περιέχουν ακριβώς τα ίδια στοιχεία και FALSE διαφορετικά. Οι λίστες που αναπαριστούν τα  $L_1$  και  $L_2$  θα πρέπει να παραμένουν αναλλοίωτες κατά την εκτέλεση των αλγορίθμων.

**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.