



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων

Τομέας Υλικού και Αρχιτεκτονικής Υπολογιστών

ΗΥ134 - Εισαγωγή τους Η/Υ

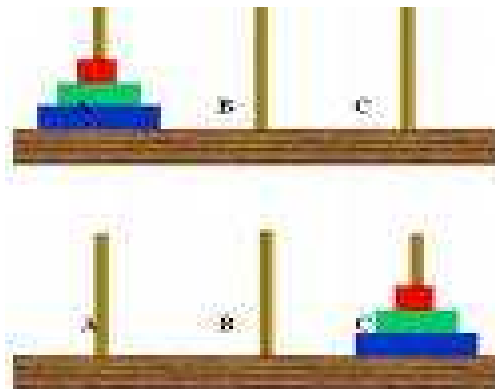
Εργαστηριακή Άσκηση 6

Εαρινό εξάμηνο 2010

Σκοπός της 6^{ης} εργαστηριακής άσκησης είναι να γίνει μια επανάληψη όσων έχουμε διδαχθεί μέχρι τώρα. Αρχίστε τις ασκήσεις αυτές νωρίς, επειδή είναι αρκετά απαιτητικές.

1^η Άσκηση (Hanoi Towers) (50%)

Υλοποιήστε σε assembly κώδικα ένα πρόγραμμα το οποίο θα υλοποιεί τον αλγόριθμο «Ο πύργος του Hanoi».



Εικόνα 1. Ο πύργος του Hanoi

Περιγραφή Αλγορίθμου:

Υπάρχουν n δακτύλιοι και τρεις στύλοι. Αρχικά οι δακτύλιοι βρίσκονται στον πρώτο στύλο, τοποθετημένοι από τον μεγαλύτερο στον μικρότερο. (βλ. Εικόνα 1)

Η άσκηση σας ζητάει να μεταφερθούν οι n δακτύλιοι από τον πρώτο στον τρίτο στύλο, χρησιμοποιώντας το δεύτερο ως βοηθητικό χώρο.

Κανόνες:

Επιτρέπεται η μετακίνηση ενός δακτυλίου τη φορά.

Κάθε μετακίνηση αποτελείται από την αφαίρεση ενός δακτυλίου που βρίσκεται στην κορυφή ενός στύλου και τοποθέτησή του στην κορυφή ενός άλλου δακτυλίου.

Σε καμία φάση δεν μπορεί να τοποθετηθεί μεγαλύτερος δακτύλιος πάνω από μικρότερο.

Περιγραφή της λύσης:

Μετακίνηση N-1 δακτυλίων από τον στύλο A στον στύλο B. (Χρησιμοποιώντας ως βοηθητικό τον στύλο C).

Μετακίνηση του μεγαλύτερου στύλου από τον δακτύλιο A στον δακτύλιο C.

Μετακίνηση N-1 δακτυλίων από τον στύλο B στον στύλο C. (Χρησιμοποιώντας ως βοηθητικό τον στύλο A).

Η έξοδος του προγράμματός σας για N=3 θα πρέπει να είναι ως εξής:

Enter number of disks: 3

Move disk 1 from peg A to peg C

Move disk 2 from peg A to peg B

Move disk 1 from peg C to peg B

Move disk 3 from peg A to peg C

Move disk 1 from peg B to peg A

Move disk 2 from peg B to peg C

Move disk 1 from peg A to peg C

Η υλοποίηση σας πρέπει να γίνει με χρήση αναδρομικής συνάρτησης. Ο ορισμός της συναρτησης είναι *hanoi(int N, int from, int using, int to)*, όπου αρχικά *from = 1*, *using=2*, και *to=3* (τα 1, 2, 3 αντιστοιχούν στα A, B, C). Ο χρήστης θα πρέπει να δίνει το N από την κονσόλα.

Πριν ξεκινήσετε την λύση του προβλήματος σε Assembly, θα ήταν χρήσιμο να την έχετε υλοποιήσει πρώτα σε μορφή ψευτοκώδικα ή ακόμα καλύτερα σε C (η υλοποίηση με αναδρομή σε C είναι πολύ απλή). Για καλύτερη κατανόηση του αλγορίθμου θα μπορούσατε να ψάξετε στο διαδίκτυο, όπου υπάρχει άφθονο υλικό πάνω στο θέμα αυτό.

π.χ. http://en.wikipedia.org/wiki/Hanoi_tower

2^η Άσκηση (Insertion Sort) (50%)

Το insertion sort (ταξινόμηση με εισαγωγή) είναι ένας απλός αλγόριθμος ταξινόμησης κατά τον οποίο κάθε στοιχείο του αρχικού πίνακα εισέρχεται ένα προς ένα στην τελική του θέση στον ταξινομημένο πίνακα.

Ας δούμε την όλη διαδικασία εισαγωγής ενός νέου στοιχείου X σε έναν μερικώς ταξινομημένο πίνακα Π1.

| | | |
|----------|-------|----|
| $\leq X$ | $> X$ | Π1 |
|----------|-------|----|

Κατ' αρχάς, μπορούμε να χωρίσουμε όλα τα στοιχεία του πίνακα Π1 σε αυτά που είναι μικρότερα ή ίσα του X και σε αυτά που είναι μεγαλύτερα του X. Επειδή ο πίνακας Π1 είναι ταξινομημένος, η πρώτη ομάδα στοιχείων θα είναι στα αριστερά της δεύτερης ομάδας (υποθέτουμε ότι ο πίνακας είναι ταξινομημένος κατά αύξουσα σειρά).

Είναι προφανώς ότι για να διατηρήσουμε την ιδιότητα του πίνακα, δηλ. να παραμείνει ταξινομημένος μετά την εισαγωγή του X, θα πρέπει το X να τοποθετηθεί ανάμεσα στις δύο ομάδες, όπως φαίνεται και στο σχήμα.

| | | | |
|----------|---|-------|----|
| $\leq X$ | X | $> X$ | Π2 |
|----------|---|-------|----|

Αυτή είναι η βασική ιδέα του αλγόριθμου ταξινόμησης με εισαγωγή (insertion sort), ενός από τους σημαντικότερους και απλούστερους αλγόριθμους ταξινόμησης. Υπάρχουν τρία βήματα που πρέπει να ακολουθήσετε για την εισαγωγή του X σε έναν ήδη ταξινομημένο πίνακα:

α) να βρείτε την θέση που θα πρέπει να τοποθετηθεί ο X στον πίνακα Π1. Έστω ότι η θέση αυτή είναι η i . Τότε θα ισχύει ότι $X(j) \leq X, \forall j < i$ και $X(k) > X, \forall k > i$

β) να μετακινήσετε όλα τα στοιχεία του πίνακα που είναι $> X$ μία θέση προς τα δεξιά για να κάνουν χώρο για το νέο στοιχείο X

γ) να τοποθετήσετε το X στην θέση i του πίνακα: $X(i) = X$

Η άσκηση αυτή σας ζητάει να γράψετε ένα πρόγραμμα σε assembly το οποίο θα υλοποιεί τον αλγόριθμο ταξινόμησης με εισαγωγή σε έναν πίνακα που δίδεται από τον χρήστη. Το πρόγραμμα αρχικά θα ζητά από τον χρήστη να εισάγει έναν θετικό ακέραιο αριθμό ($n > 0$). Στην συνέχεια ο χρήστης θα εισάγει από την κονσόλα n αριθμούς και θα τους αποθηκεύει σε έναν πίνακα στο τμήμα **.data** της μνήμης. Χρησιμοποιώντας τον αλγόριθμο όπως περιγράψαμε παραπάνω, το πρόγραμμα σας θα πρέπει να δημιουργεί και στην συνέχεια να εκτυπώνει τον ταξινομημένο πίνακα.

Σημείωση

Είναι σημαντικό η υλοποίηση των ασκήσεων να γίνει με κλήση συναρτήσεων ακριβώς όπως περιγράφει η εκφώνηση.

Κάθε συνάρτηση πρέπει να εξασφαλίζει ότι οι λεγόμενοι *Saved Temporary* καταχωρητές διατηρούν το περιεχόμενο που είχαν πριν την κλήση της.

Μην κάνετε άσκοπα χρήση της στοίβας. Η χωρίς όρια δέσμευση περιττής μνήμης αποτελεί προγραμματιστικό λάθος.