

ΕΡΩΤΗΜΑ 10

Στην άσκηση 10 πρέπει να υλοποιήσουμε μια συνάρτηση η οποία να βρίσκει πόσες φορές υπάρχει κάθε κλειδί μέσα σε ένα δέντρο (BST). Με άλλα λόγια να βρούμε τη συχνότητα κάθε κλειδιού.

main10.c

Είναι το main αρχείο (περιέχει τη main). Δέχεται από τη γραμμή εντολής ως ορίσματα τα στοιχεία του BST (θεωρώ ότι ο χρήστης θα δώσει int ορίσματα γιατί γίνεται χρήση "atoi"). Έτσι αρχικά κάνει initialize το BST και μετά insert τα δεδομένα που δώθηκαν. Στη συνέχεια τυπώνονται τα στοιχεία του δέντρου σε φθίνουσα σειρά (inorder traversal) και μετά καλείται η συνάρτηση KeyFrequency.

mods10.c

Είναι το αρχείο που περιέχει όλες τις συναρτήσεις. Η KeyFrequency δέχεται ως όρισμα το πλήθος των στοιχείων που υπάρχουν στο BST. Στη συνάρτηση αρχικά αρχικοποιείται ένας 2D πίνακας. Αν υπάρξει πρόβλημα στη malloc το πρόγραμμα τερματίζει. Ο πίνακας έχει διαστάσεις $n*2$ όπου n = το πλήθος των στοιχείων μέσα στο BST. Έτσι ουσιαστικά κάθε σειρά περιέχει 2 πληροφορίες. Στην αριστερή στήλη υπάρχει η τιμή του κλειδιού και στη δεξιά η συχνότητα με την οποία εμφανίζεται στο δέντρο (πχ το "4->3" σημαίνει ότι το στοιχείο "4" εμφανίζεται 3 φορές στο BST). Ο τρόπος με τον οποίο συμπληρώνεται ο πίνακας είναι ο εξής:

Καλείται η STselect τόσες φορές, όσοι είναι και οι κόμβοι του δέντρου. Τα ορίσματα είναι 0,1,2,...,n-1. Έτσι ουσιαστικά είναι σα να διασχίζουμε όλο το δέντρο σε φθίνουσα σειρά.

Στην 1η γραμμή / 1η στήλη αποθηκεύεται το μικρότερο στοιχείο του δέντρου και στη 2η στήλη ο αριθμός "1" (δηλ. βρέθηκε το "χ" στοιχείο "1" φορά). Μετά βρίσκουμε το 2ο στοιχείο του BST. Αν αυτό είναι ίδιο με το 1ο τότε απλά αυξάνουμε το πλήθος κατά 1. Αν διαφέρει από το 1ο τότε πηγαίνουμε στην επόμενη γραμμή και στην 1η στήλη βάζουμε την τιμή του νέου στοιχείου και στη 2η στήλη τον αριθμό 1. Αυτό επαναλαμβάνεται μέχρι να διασχίσουμε όλο το δέντρο. Αφού συμπληρώσουμε τον πίνακα, εμφανίζουμε τα στοιχεία που περιέχει και τερματίζει η συνάρτηση.

item-implementation.c

Περιέχει κάποιες έτοιμες συναρτήσεις.

mods10.h / Item.h

Είναι αρχεία στα οποία γίνονται δηλώνονται συναρτήσεις, τύποι μεταβλητών (typedef) και μακροεντολές.

Το makefile μου έχει τις εξής λειτουργίες:

- make
 - gcc -c main10.c
 - gcc -c mods10.c
 - gcc -c item-implementation.c
 - gcc -o ask10 main10.o mods10.o gcc -c item-implementation.o
- make clean
 - rm *.o ask10

Για την εκτέλεση του προγράμματος αρκεί να γράψετε:

- make
- ./ask10 orisma1 orisma2 (...)

Όπου:

- orisma1,2,3,... είναι τα στοιχεία του δέντρου που θα δώσει ο χρήστης.

Ενδεικτική εκτέλεση:

- make
- ./ask10 23 11 2 6 2 3 3 3 8 3 9 54 45 66 66

Τα δέντρα που θα μπορούσαν να χρησιμοποιηθούν ώστε η πολυπλοκότητα να γίνει $O(n \log m)$ είναι τα δέντρα αναζήτησης m δρόμων (multi-way search trees).

Υ.Γ.: Έχουν χρησιμοποιηθεί οι συναρτήσεις που παρέχονται στη σελίδα του μαθήματος.