

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΠΛ231: Δομές Δεδομένων και Αλγόριθμοι

Εαρινό Εξάμηνο 2013

Φροντιστήριο 7 - ΛΥΣΕΙΣ

Άσκηση 1

Σας δίνεται ένας πίνακας από n ακεραίους με αρκετές επαναλήψεις στοιχείων έτσι που ο αριθμός των διακριτών στοιχείων του πίνακα να είναι της τάξης $O(\lg n)$. Να κατασκευάσετε ένα αλγόριθμο ταξινόμησης ο οποίος να ταξινομεί τον πίνακα χρησιμοποιώντας αριθμό συγκρίσεων της τάξης $O(n \lg \lg n)$. (Υπόδειξη: Μπορείτε να χρησιμοποιήσετε κάποια βοηθητική δενδρική δομή.)

Θα επεκτείνουμε τη δομή δεδομένων AVL-δένδρο έτσι ώστε να λαμβάνει υπόψη τον αριθμό επαναλήψεων στοιχείων ως εξής. Σε κάθε θέση του δένδρου φυλάγουμε ζεύγη της μορφής (k, n) όπου k είναι το αποθηκευμένο κλειδί και n είναι το πλήθος των επαναλήψεων του κλειδιού k μέσα στο δένδρο. Οι πράξεις AVL-δένδρου επηρεάζονται ως εξής:

- $\text{Insert}(k, T)$: Αναζήτησε στο δένδρο το κλειδί k . Αν υπάρχει σε ζεύγος (k, n) ανανέωσε την πληροφορία αυτή έτσι ώστε να λάβει υπόψη τη νέα εμφάνιση του k , δηλαδή $(k, n) := (k, n + 1)$. Διαφορετικά, αν το k δεν υπάρχει στο δένδρο, εφάρμοσε τη γνωστή διαδικασία εισαγωγής σε AVL-δένδρο για το ζεύγος $(k, 1)$. Προφανώς η διαδικασία Insert εξακολουθεί να διατηρεί χρόνο εκτέλεσης της τάξης $O(\lg n)$ όπου n είναι ο αριθμός των κόμβων του δένδρου
- $\text{InOrder}(T)$: Τύπωσε τα στοιχεία του δένδρο σε αύξουσα σειρά με τη διαφορά ότι για κάθε κόμβο με κλειδί (k, n) τυπώνουμε το στοιχείο k , n φορές. Προφανώς η διαδικασία InOrder εξαρτάται από την πολλαπλότητα των στοιχείων του δένδρου τα οποία αναπαριστούν τα στοιχεία του πίνακα με n στοιχεία συνεπώς έχει χρόνο εκτέλεσης της τάξης $O(n)$.

Υποθέτοντας υλοποίηση της δομής με όνομα MultiAVL ορίζουμε τον πιο κάτω αλγόριθμο.

```
int A[n];
MultiAVL tree;
for (i = 0; i < n; i++)
    Insert(A[i], tree);
InOrder(tree);
```

Ο αλγόριθμος αυτός τοποθετεί τα στοιχεία του πίνακα σε ένα MultiAVL-δένδρο μέσω η διαδοχικών κλήσεων της Insert και στη συνέχεια τα τυπώνει σε αύξουσα σειρά μέσω κλήσης της InOrder.

Παρατηρούμε ότι αφού ο πίνακας A περιέχει μόνο $\lg n$ διαφορετικά στοιχεία, ο αριθμός κόμβων του δένδρου B αφού δημιουργηθεί θα είναι $\lg n$. Επομένως ο χρόνος εκτέλεσης της Insert θα είναι της τάξης $O(\lg \lg n)$ και της InOrder $O(n)$. Έτσι, ο χρόνος εκτέλεσης του πιο πάνω αλγόριθμου ταξινόμησης είναι $O(n \lg \lg n + n) = O(n \lg \lg n)$, όπως ζητείται από την άσκηση.

Άσκηση 2

Εισάγετε τα στοιχεία 65, 76, 71, 79, 82, 73, 84, 72, 77, 83 σε ένα 2-3 δέντρο.

