

# Δομές Δεδομένων - Εργασία 3

**Στυλιανός Μουζακίτης : 3120111**

## Παράρτημα α.

Μέθοδος **remove**(int AFM):

Ψάχνουμε και αποθηκεύουμε τον κόμβο που θέλουμε να αφαιρέσουμε σύμφωνα με το ΑΦΜ.

Σε περίπτωση που δεν τον βρούμε επιστρέφουμε **null**.

Σε περίπτωση που ο αφαιρούμενος κόμβος έχει 2 παιδιά, βρίσκουμε τον διαδοχό του(κατά την εύρεση του διαδόχου, ενημερώνουμε το πεδίο N, στους κόμβους που συναντάμε) και αντιγράφουμε τα στοιχεία του στον αφαιρούμενο κόμβο. Με αυτόν τον τρόπο έχουμε 2 κόμβους με τα στοιχεία του διαδόχου. Έπειτα μεταφερόμαστε στον διάδοχο κόμβο, έτσι ώστε να είναι αυτός ο κόμβος που θα επεξεργαζόμαστε από εδώ και πέρα.

Αποθηκεύουμε τον πατέρα του κόμβου που επεξεργαζόμαστε και αποθηκεύουμε ως παιδί το αριστερό παιδί του κόμβου, αν υπάρχει, αλλιώς το δεξί(αν υπάρχει).

Αν κόμβος δεν έχει πατέρα σημαίνει ότι επεξεργαζόμαστε τη ρίζα του δέντρου και έτσι την αφαιρούμε αλλιώς συνδέουμε τον πατέρα με το παιδί, ανάλογα με το πού βρίσκεται ο κόμβος που επεξεργαζόμαστε, και το παιδί(αν υπάρχει) με τον πατέρα.

Επειδή όμως το πεδίο N των προγόνων του αφαιρούμενου κόμβου δεν έχει ενημερωθεί, κάνουμε μια αναζήτηση αυτού και το ενημερώνουμε στους κόμβους που συναντάμε.

Ύστερα αποσυνδέουμε τον κόμβο και μειώνουμε κατά ένα το μέγεθος του πίνακα συμβόλων.

Μέθοδος **insert**(Suspect item):

Εισάγει έναν ύποπτο σύμφωνα με το ΑΦΜ του ως φύλλο του δέντρου.

Αν το αντικείμενο είναι **null**, πετάμε exception.

Ψάχνουμε αν υπάρχει ήδη κάποιος ύποπτος με το ίδιο ΑΦΜ για να διαβεβαιώσουμε ότι το ΑΦΜ είναι μοναδικό.

Ξεκινάμε από τη ρίζα και συγκρίνουμε τα ΑΦΜ κατεβαίνοντας στο αριστερό η δεξί υποδέντρο ανάλογα με το αποτέλεσμα κρατώντας τον τελευταίο πατέρα κάθε φορά και ενημερώνοντας το πεδίο N των κόμβων που συγκρίνουμε. Δημιουργούμε τον νέο μας κόμβο και τον τοποθετούμε ως αριστερό ή ως δεξί παιδί του τελευταίου πατέρα είτε ως ρίζα αν δεν υπήρχε πατέρας.

Μέθοδος **insertTopSuspect**(Suspect item):

Κάνει την ίδια επεξεργασία με την μέθοδο **insert**, μόνο που τώρα εισάγει έναν ύποπτο συγκρίνοντας την ποσότητα savings – taxed\_income, του κάθε υπόπτου.

Μέθοδος **insert\_at\_root**(Suspect item):

Εισάγει έναν ύποπτο σύμφωνα με το ΑΦΜ του ως ρίζα του δέντρου.

Χρησιμοποιεί την αναδρομική μέθοδο **insert\_at\_rootRecursive**, η οποία εισάγει αναδρομικά τον ύποπτο ως φύλλο και ύστερα κάνει τα κατάλληλα rotations(καλώντας την **rotateLeft** ή την **rotateRight**), οι οποίες με τη σειρά τους ενημερώνουν το πεδίο N του κόμβου περιστροφής και του κόμβου παίρνει τη θέση του.

Μέθοδος **load**(String filename):

Ανοίγουμε το αρχείο εισόδου χρησιμοποιώντας BufferedReader και διαβάζουμε και αποθηκεύουμε τα στοιχεία υπόπτων χρησιμοποιώντας StringTokenizer.  
Δημιουργούμε αντικείμενα τύπου Suspect με τα στοιχεία που αποθηκεύσαμε και τους εισάγουμε στον πίνακα συμβόλων μας.

Μέθοδος **getMeanSavings**():

Υπολογίζει το μέσο ποσό καταθέσεων με βάση τους καταθέτες που είναι αποθηκευμένοι στο δέντρο. Χρησιμοποιεί την αναδρομική μέθοδο **getMeanSavingsRecursive**, η οποία διασχίζει το δέντρο προ-διατεταγμένα προσθέτοντας κάθε φορά τις καταθέσεις των υπόπτων. Ύστερα καλεί την μέθοδο **countOfSt()**, η οποία χρησιμοποιώντας την αναδρομική μέθοδο **countOfStRecursive**, που παρόμοια διασχίζει το δέντρο προ-διατεταγμένα, επιστρέφει τον αριθμό των κόμβων ολόκληρου του δέντρου. Έτσι διαιρεί τις συνολικές καταθέσεις με τον αριθμό των υπόπτων και επιστρέφει το μέσο ποσό των καταθέσεων.

Μέθοδος **printTopSuspects**(int k):

Δημιουργούμε ένα καινούριο δέντρο.

Καλούμε την μέθοδο **topTreeCreator**, η οποία, χρησιμοποιώντας την αναδρομική μέθοδο **topTreeCreatorRecursive**, που διασχίζει το αρχικό δέντρο προ-διατεταγμένα εισάγοντας κάθε φορά τον κόμβο που συναντάει στο καινούριο δέντρο μέσω της **insertTopSuspect**, επιστρέφει το νέο δέντρο ταξινομημένο, σύμφωνα με ποσότητα savings – taxed\_income.

Ύστερα, καλούμε την μέθοδο **printkMax**, η οποία τυπώνει τους k πιο ύποπτους καταθέτες, καλώντας κάθε φορά την μέθοδο **selectMax**, για να βρεί το k-οστό μεγαλύτερο στοιχείο, η οποία με τη σειρά της χρησιμοποιεί την αναδρομική μέθοδο **selectMaxR**. Η τελευταία μέθοδος υλοποιεί την εξής αναδρομή:

*Έστω ότι το δεξί υποδέντρο έχει t κλειδιά*

- *An  $k-1 < t$  ψάξε το k-οστό κλειδί στα δεξιά*
- *An  $k-1 > t$  ψάξε το (k-t-1)-οστό κλειδί στα αριστερά*
- *An  $k-1 = t$  τότε το k-οστό κλειδί είναι στη ρίζα.*

Μέθοδος **printTreeByAFM**(PrintStream stream):

Τυπώνει τα στοιχεία όλων των κόμβων του δέντρου, που είναι ταξινομημένοι ως προς το ΑΦΜ, χρησιμοποιώντας την αναδρομική μέθοδο **toStringRecursive**, που διασχίζει το δέντρο ενδο-διατεταγμένα.

Μέθοδος **searchByAFM**(int AFM):

Ψάχνει στο δέντρο για την ύπαρξη υπόπτου με το συγκεκριμένο ΑΦΜ (επιστρέφει null αν δεν υπάρχει). Χρησιμοποιεί την αναδρομική μέθοδο **searchByAFMRRecursive**, η οποία συγκρίνοντας τα ΑΦΜ κατεβαίνει στο αριστερό ή δεξί υποδέντρο ανάλογα με το αποτέλεσμα.

Μέθοδος **searchByLastname**(String last\_name):

Ψάχνει στο δέντρο για την ύπαρξη υπόπτου με το συγκεκριμένο επίθετο.Επειδή ενδέχεται να υπάρχουν περισσότεροι του ενός ατόμου με το ίδιο επίθετο, η μέθοδος επιστρέφει μια λίστα μονής σύνδεσης με αντικείμενα τύπου Suspect, χρησιμοποιώντας την αναδρομική μέθοδο **searchByLastnameRecursive**, η οποία διασχίζει το δέντρο προ- διατεταγμένα και εισάγει στη λίστα όλους τους υπόπτους με αυτο το επίθετο.

## Παράρτημα b.

Ξεκινάμε το πρόγραμμά μας με την εμφάνιση, στο comand line, του κυρίου Μενού.

```
1: INSERTION/DELETION of suspects
2: SEARCH suspects
3: PRINT suspects
4: LOAD a file of suspects
5: CALCULATE MEAN of savings of suspects
0: EXIT the program
Please select your action by typing the number of the choice:
```

Ο χρήστης διαλέγει τί θέλει να κάνει πληκτρολογώντας μια από τις εμφανιζόμενες επιλογες(0,1,2,3,4,5,).

Αν δεν πληκτρολογήσει μια έγκυρη επιλογή, το σύστημα τον προτρέπει να ξαναπροσπαθήσει μέχρι να βρει μια που να είναι, με το ακόλουθο μήνυμα.

```
Please select a valid choice:
```

Με την επιλογή 0, τερματίζεται το πρόγραμμα.

Με την επιλογη 5, υπολογίζει κατευθείαν το μέσο ποσό των καταθέσεων(Αν είναι άδειο το δέντρο επιστρέφει 0.0)

Με την επιλογή 4, ζητείται από τον χρήστη να εισάγει το μονοπάτι του επιθυμητού αρχείου εισόδου(χωρίς αυτάκια “”) βάζοντας στο τέλος της πληκτρολόγησης του, το ελληνικό ερωτηματικό(;).

π.χ.

```
Give the path of the file, write ; at the end of the name to continue:
```

```
i.e:path_of_file.txt;
C:\\Users\\stelios\\Desktop\\domes3 pros paradosh\\ergasia3\\src\\treeFile.txt;
```

Με κάθε άλλη επιλογή εμφανίζεται το αντίστοιχο υπομενού και ανάλογα με την επόμενη επιλογή, εκτελείται η λειτουργία ακολουθώντας τα βήματα του αντίστοιχου υπομενού.

Στον χρήστη δίνεται πάντα η επιλογή να γυρίσει στο αρχικό μενού πληκτρολογώντας το 0.