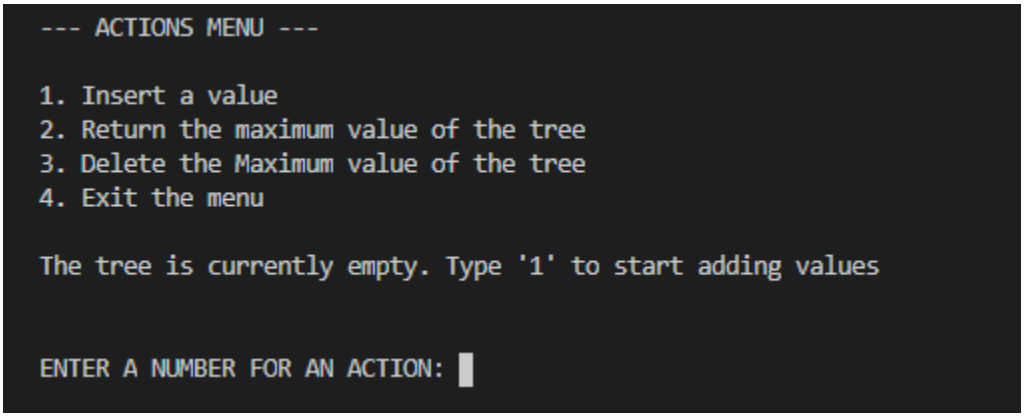
Δομές Δεδομένων: 2η Εργασία

Νίκος Γεωργιάδης – Π19032

Γιώργος Σεϊμένης – Π19204

Κατά την εκτέλεση του προγράμματος της άσκησης (main.exe), εμφανίζεται το μενού των διαθέσιμων επιλογών. Στην περίπτωση που στον εκτελέσιμο κώδικα υπάρχει ήδη κάποιο δέντρο, αυτό θα εμφανιστεί.



Αφού έχει εμφανιστεί το δέντρο σε Προδιάταξη, ύστερα Ενδοδιάταξη και Μεταδιάταξη, οι διαθέσιμες ενέργειες είναι:

- Εισαγωγή μιας μεταβλητής
- Εμφάνιση της μέγιστης μεταβλητής του δέντρου
- Διαγραφή της μέγιστης μεταβλητής του δέντρου
- Τερματισμός του προγράμματος.

Οι ενέργειες αυτές πραγματοποιούνται εισάγοντας τον αριθμό της αντίστοιχης ενέργειας. Η κάθε ενέργεια έχει και τη δική της συνάρτηση στον κώδικα ως μέλος της κλάσης `BinaryTreeNode`. Παρακάτω θα εξηγηθούν οι διαδικασίες, που χρησιμοποιήθηκαν στις συναρτήσεις της κλάσης αυτής.

ΕΙΣΑΓΩΓΗ ΝΕΟΥ ΣΤΟΙΧΕΙΟΥ

Χρησιμοποιώντας τη συνάρτηση `BinaryTreeNode::Insert()` γίνεται η διαδικασία της εισαγωγής. Για την διαδικασία της εισαγωγής, χρησιμοποιείται η εξής μέθοδος:

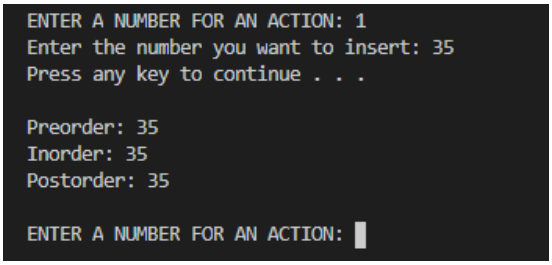
Χρησιμοποιείται ένας δείκτης που περνάει από όλους τους κόμβους του δέντρου (ξεκινώντας από την ρίζα). Για κάθε κόμβο που συναντάει, ισχύουν τα παρακάτω.

- Αν το στοιχείο προς εισαγωγή είναι μεγαλύτερο ενός κόμβου, τότε, αναγκαστικά, ο δείκτης πάει δεξιά
- Αν το στοιχείο προς εισαγωγή είναι μικρότερο, τότε υπάρχει 50% πιθανότητα για τον δείκτη να πάει είτε δεξιά, είτε αριστερά

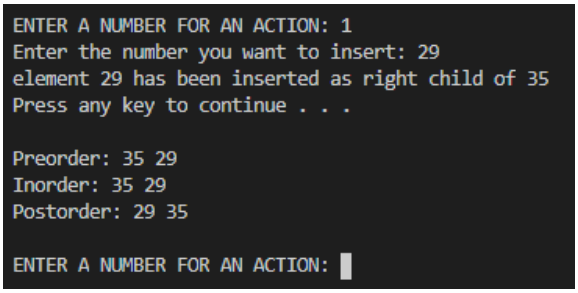
Παράδειγμα:

Έστω ότι στην αρχή του προγράμματος, όπου το δέντρο θα ‘ναι κενό, εισάγουμε πρώτα τον αριθμό 35 (ο οποίος θα είναι και η ρίζα του δέντρου). Εάν, μετά, επιχειρήσουμε να βάλουμε τον κόμβο 29, πάει τυχαία, είτε αριστερά, είτε δεξιά. Στο τέλος της διαδικασίας της εισαγωγής, εξετάζονται οι περιπτώσεις, στις οποίες το δέντρο δεν έχει σωστή ισορροπία (για δέντρα AVL) και χρειάζεται περιστροφή κάποιος κόμβος.

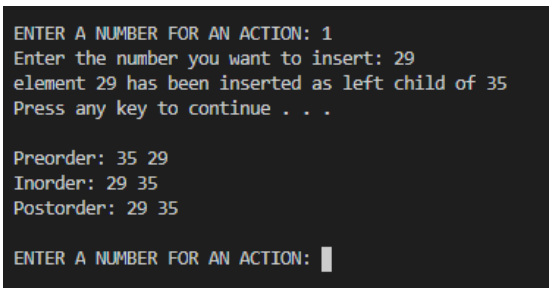
Εικόνα 1: Εισαγωγή του κόμβου 35 (ρίζα).



Εικόνα 2: Σε αυτήν την περίπτωση, ο κόμβος 29 εισήχθη στα δεξιά του 35.



Εικόνα 3: Αντιθέτως, σε αυτήν την περίπτωση, ο κόμβος 29 εισήχθη στα αριστερά του 35.



ΕΥΡΕΣΗ ΜΕΓΙΣΤΟΥ ΣΤΟΙΧΕΙΟΥ

Μέσα στα μέλη της κλάσης `BinaryTreeNode` υπάρχει και ο δείκτης `Maxkey` ο οποίος ευθύνεται για την ανάδειξη του μεγίστου στοιχείου του δέντρου. Σε κάθε συνάρτηση του δέντρου, ο δείκτης αυτός ενημερώνεται. Με αυτόν τον τρόπο, καταφέρνουμε να έχουμε ταχύτητα εύρεσης μεγίστου O(1) σε οποιοδήποτε μέγεθος του δέντρου.

```
Preorder: 56 47 34 27 1 42 55 54 6 43 98 97 91 15 95 22 89 26 100 99 32 77 109 104 127 110 115
Inorder: 1 27 34 42 47 54 55 6 43 56 15 91 22 95 97 89 26 98 32 99 77 100 104 109 110 127 115
Postorder: 1 27 42 34 54 43 6 55 47 15 22 95 91 26 89 97 32 77 99 104 110 115 127 109 100 98 56

ENTER A NUMBER FOR AN ACTION: 2
The maximum value of the tree is: 127
Press any key to continue . . .
```

Στο παραπάνω δέντρο, δεν έχει σημασία που υπάρχουν 27 κόμβοι, ο χρόνος, ο οποίος χρειάζεται για να εμφανιστεί το μέγιστο, είναι O(1).

ΔΙΑΓΡΑΦΗ ΜΕΓΙΣΤΟΥ ΣΤΟΙΧΕΙΟΥ

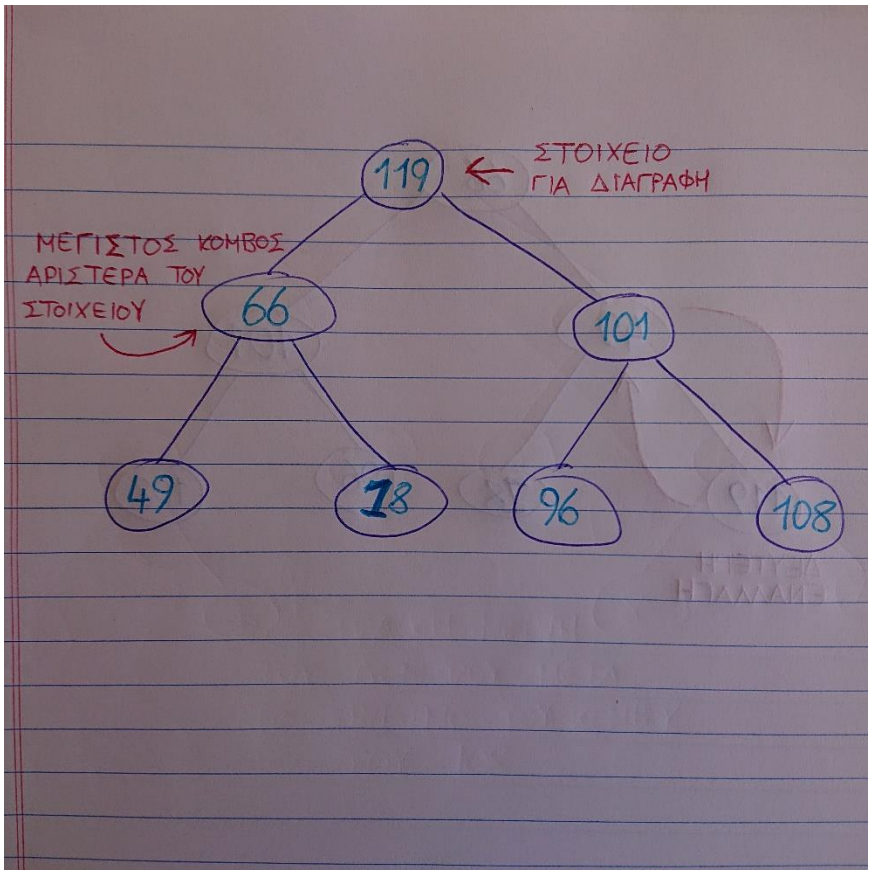
Η διαδικασία της διαγραφής μεγίστου γίνεται με αναδρομικό τρόπο, με τη χρήση της συνάρτησης `deleteNode`. Μόλις η συνάρτηση βρει τον κόμβο του μεγίστου, ακολουθεί η εξής τεχνική:

- Αν ο κόμβος του μεγίστου είναι φύλλο, τότε σβήνεται ο κόμβος ως έχει.
- Αν ο κόμβος του μεγίστου έχει ως αριστερό, ή δεξί παιδί ένα φύλλο, τότε εναλλάσσεται ο κόμβος του μεγίστου με το φύλλο. Μετά, σβήνεται το φύλλο.
- Αν ο κόμβος του μεγίστου έχει δύο παιδιά, η συνάρτηση βρίσκει τον μέγιστο κόμβο από τα αριστερά του μεγίστου. Αφού βρει το μέγιστο από τα αριστερά, εναλλάσσει τις θέσεις του μεγίστου, με του μεγίστου από τα αριστερά και η διαδικασία επαναλαμβάνεται μέχρι ο κόμβος να φτάσει σε σημείο που είναι φύλλο. Όταν φτάσει σε σημείο να είναι φύλλο, επειδή η συνάρτηση είναι αναδρομική, θα εκτελεστούν οι δύο παραπάνω περιπτώσεις.

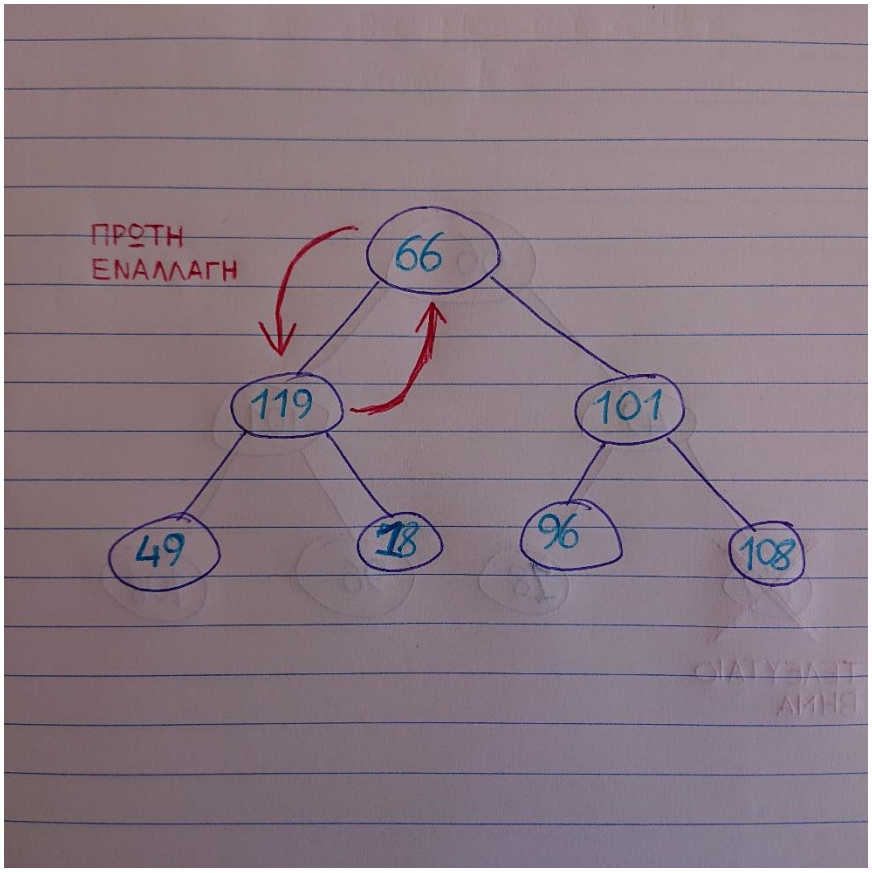
Κάθε φορά που η διαδικασία της διαγραφής φτάνει στο τέλος της, ελέγχονται όλα τα ενδεχόμενα περιστροφής των κόμβων, σε περίπτωση που οι κόμβοι δεν έχουν την σωστή ισορροπία (διάταξη AVL δέντρων).

Παράδειγμα:

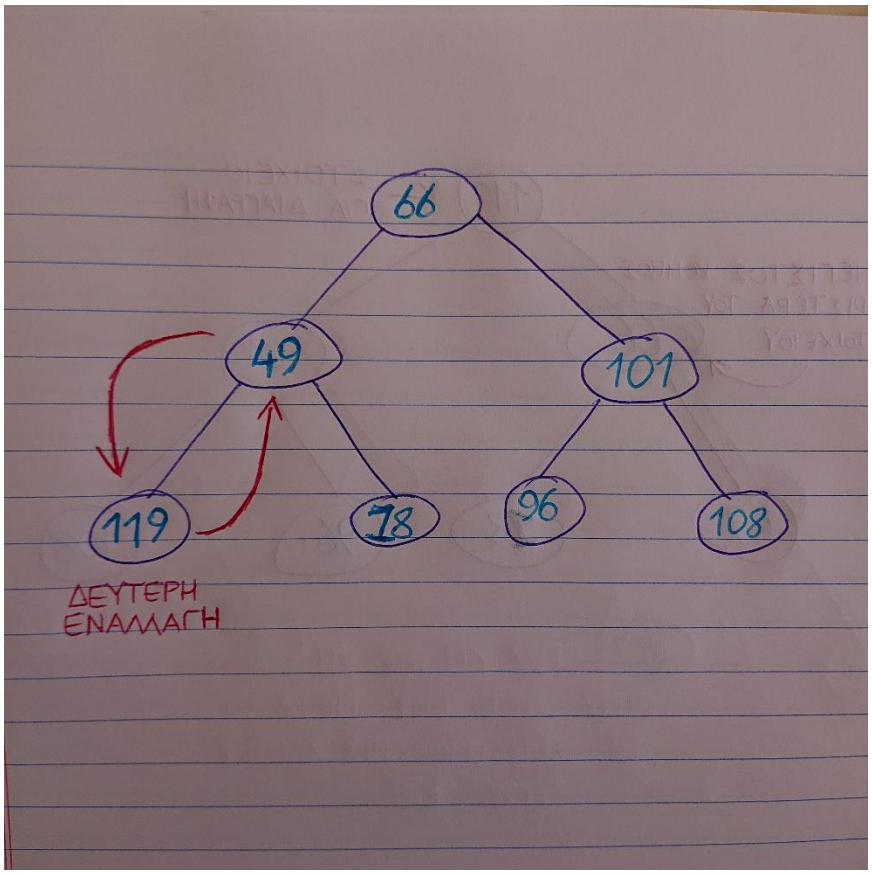
Έστω το εξής δέντρο, με μέγιστο τον κόμβο 119.



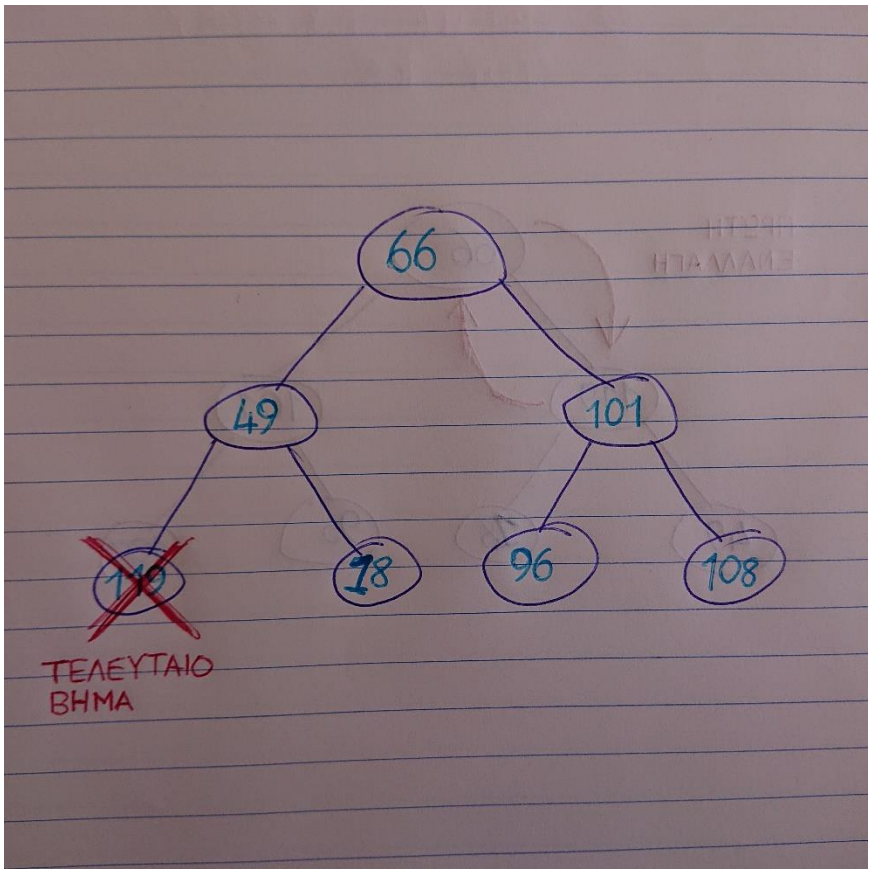
Όταν η συνάρτηση βρει τον κόμβο 119, αρχίζει να ψάχνει το μέγιστο στοιχείο από τα αριστερά του κόμβου (σε αυτήν την περίπτωση, το 66).



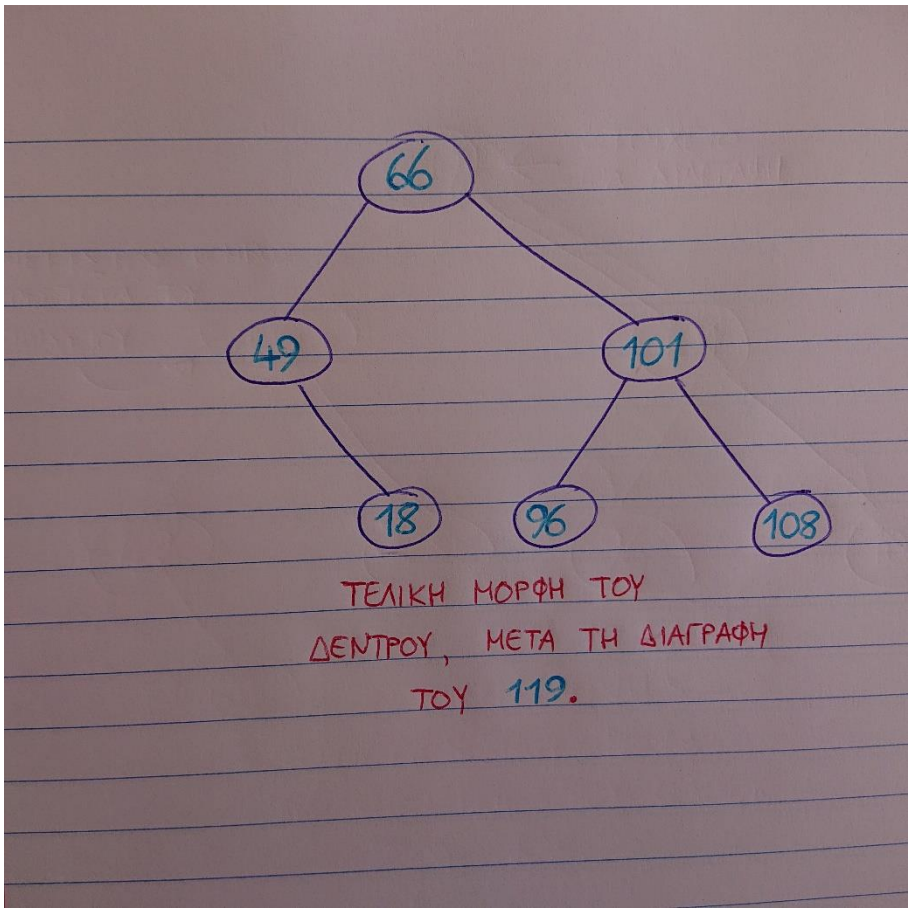
Τώρα που έγινε η εναλλαγή, το 119 έχει επίσης δύο παιδιά. Τώρα, θα βρει το (μοναδικό) μέγιστο παιδί από τα αριστερά, το 49. Όταν το βρει, θα γίνει και η δεύτερη εναλλαγή, μέχρι το 119 να γίνει επιτέλους φύλλο.



Τώρα που το 119 έγινε φύλλο, θα σβηστεί.



Και η τελική μορφή του δέντρου, θα είναι ως εξής.



Κάθε φορά που καλείται η `deleteNode`, γίνεται έλεγχος για τον αν χρειάζεται να γίνει κάποιο rotation (σε αυτήν την περίπτωση δεν χρειάστηκε πουθενά).

Εκτέλεση του προγράμματος:

Εντός του πηγαίου κώδικα υπάρχουν σχόλια στα αγγλικά, που εξηγούν τις τεχνικές που χρησιμοποιήθηκαν τόσο για τις συναρτήσεις της κλάσης όσο και για την εμφάνιση του μενού στην κονσόλα. Η μεταγλώττιση του πηγαίου κώδικα έγινε με τη χρήση του MinGW.

~ Οι φοιτητές,

Νίκος Γεωργιάδης (Π19032) & Γιώργος Σεϊμένης (Π19204).