

Πανεπιστήμιο Ιωαννίνων – Τμήμα Μηχανικών Η/Υ και Πληροφορικής  
Δομές Δεδομένων [ΜΥΥ303] – Χειμερινό Εξάμηνο 2024

3<sup>η</sup> Εργαστηριακή Άσκηση  
Δένδρα και Ισομορφισμοί

Παράδοση έως Τετάρτη 6/11, 14:00 από το eCourse

**ΠΡΟΣΟΧΗ:** Γράψτε σε κάθε αρχείο που παραδίδετε τα ονόματα, τους Α.Μ. των μελών της ομάδας σας, καθώς και το group του εργαστηρίου σας. Συμπεριλάβετε όλα τα αρχεία σας (κώδικας Java και lab3results.txt) σε ένα zip αρχείο. Το όνομα που θα δώσετε στο συμπίεσμένο αρχείο θα αποτελείται από το group του εργαστηρίου στο οποίο έχετε τοποθετηθεί καθώς και από το ID της ομάδας σας (π.χ., G1\_ID1.zip).

Μας δίνονται δύο ημιτελή προγράμματα RootedTree.java και RootedTreeIso.java, τα οποία πρέπει να συμπληρώσουμε ώστε να επιτελούν την ακόλουθη λειτουργία. Το πρόγραμμα RootedTreeIso.java δέχεται στην είσοδο δύο δένδρα με ρίζα, όπου και τα δύο έχουν  $N$  κόμβους με αρίθμηση από 0 έως  $N - 1$ . Σκοπός του προγράμματος είναι να ελέγξει αν τα δύο δένδρα είναι **ισομορφικά**, με τη βοήθεια μιας κωδικοποίησης των κόμβων των δύο δένδρων με δυαδικές ακολουθίες, οι οποίες υπολογίζονται από το πρόγραμμα RootedTree.java.

### Ισομορφισμός δένδρων

Έστω  $T_1$  και  $T_2$  δύο δένδρα, με ρίζες τους κόμβους  $r_1$  και  $r_2$ , αντίστοιχα. Συμβολίζουμε με  $V_i$  το σύνολο των κόμβων του δένδρου  $T_i$ . Τα δένδρα  $T_1$  και  $T_2$  είναι **ισομορφικά** αν και μόνο αν υπάρχει μια 1-προς-1 και επί συνάρτηση  $f : V_1 \rightarrow V_2$  τέτοια ώστε  $f(r_1) = r_2$  και για κάθε ζεύγος κόμβων  $x, y \in V_1$  ισχύει  $\{x, y\} \in E_1$  αν και μόνο αν  $\{f(x), f(y)\} \in E_2$ .

Προσέξτε ότι υπάρχουν δένδρα με ρίζα που συμφωνούν μεταξύ τους στο πλήθος των επιπέδων, στο πλήθος των κόμβων σε κάθε επίπεδο και στην ακολουθία βαθμών των κόμβων κάθε επιπέδου, αλλά δεν είναι ισομορφικά μεταξύ τους. Σκοπός την εργασίας είναι να υλοποιηθεί αποδοτικά ο αλγόριθμος για τον έλεγχο ισομορφισμού δύο δένδρων που περιγράφουμε στη συνέχεια.

### Ετικέτες κόμβων

Έστω δένδρο  $T$  με ρίζα  $r$ . Ορίζουμε μια αναδρομική μέθοδο LabelNode, η οποία αναθέτει σε κάθε κόμβο  $v$  του  $T$  μια ετικέτα  $label[v]$ . Η ετικέτα αυτή είναι μια δυαδική ακολουθία, η οποία κωδικοποιεί τη μορφή του υποδένδρου που ορίζεται από τους απογόνους του  $v$ . Για δύο δυαδικές ακολουθίες  $l_1$  και  $l_2$ , συμβολίζουμε τη συνένωσή τους ως  $l_1 + l_2$ . (Π.χ., "110" + "01" = "11001".)

```
LabelNode(v) {  
    if (ο v είναι φύλλο) then { label[v] = "10" }  
    else {  
        for (κάθε παιδί w του v) { l_w = LabelNode(w) }  
        Ταξινόμησε τις ετικέτες των παιδιών του v σε φθίνουσα  
        λεξικογραφική σειρά  $l_1 \geq l_2 \geq \dots \geq l_k$   
        label[v] = "1" + l_1 + l_2 + ... + l_k + "0"  
    }  
    return label[v]  
}
```

### Έλεγχος ισομορφισμού δένδρων

Αποδεικνύεται ότι δύο δένδρα  $T_1$  και  $T_2$  με ρίζες  $r_1$  και  $r_2$ , αντίστοιχα είναι ισομορφικά αν και μόνο αν  $T_1.\text{label}[r_1] == T_2.\text{label}[r_2]$ .

### Ζητούμενες μέθοδοι

Υλοποιήστε στο πρόγραμμα RootedTreeLabel.java την παρακάτω μέθοδο

String LabelNode(int  $v$ ): Υπολογίζει την ετικέτα  $\text{label}[v]$  του κόμβου  $v$ , αναδρομικά από τις ετικέτες των παιδιών του, όπως περιγράψαμε παραπάνω.

Επίσης, υλοποιήστε στο πρόγραμμα RootedTreeIso.java την ακόλουθη μέθοδο

int[] ComputeMap (RootedTreeT1, RootedTreeT2) Με την προϋπόθεση ότι τα δύο δένδρα  $T_1$  και  $T_2$  είναι ισομορφικά, υπολογίζει μια συνάρτηση ισομορφισμού  $f : V_1 \rightarrow V_2$ , ως πίνακα  $\text{mapNode}[]$ . Δηλαδή, αν έχουμε  $f(v) = u$  τότε  $\text{mapNode}[v] = u$ .

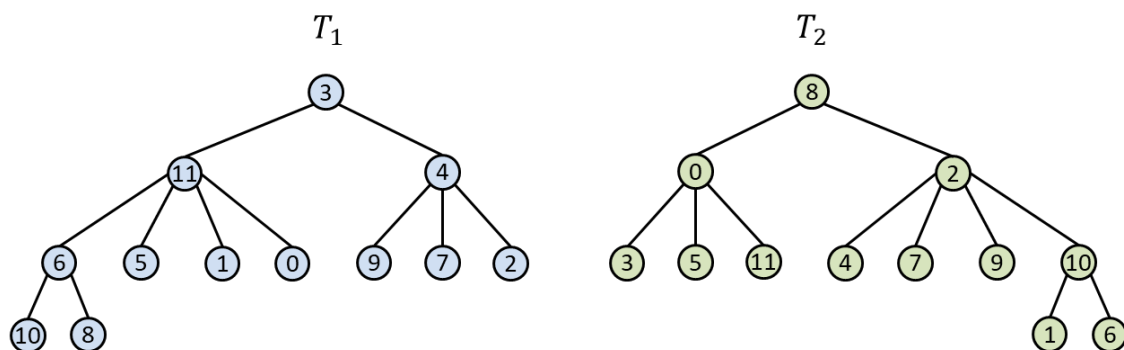
### Αρχεία εισόδου

Τα δένδρα δίνονται ως μια ακολουθία  $2N + 1$  ακέραιων, όπου ο πρώτος ακέραιος αντιστοιχεί στο πλήθος των κόμβων  $N$ , οι επόμενοι  $N$  ακέραιοι  $v_0, v_1, v_2, \dots, v_{N-1}$  δίνουν τον γονέα του κάθε κόμβου στο δένδρο  $T_1$ , και οι τελευταίοι  $N$  ακέραιοι  $u_0, u_1, u_2, \dots, u_{N-1}$  δίνουν τον γονέα του κάθε κόμβου στο δένδρο  $T_2$ . Συγκεκριμένα, ο κόμβος  $v_k$  είναι ο γονέας του κόμβου  $k$  στο δένδρο  $T_1$ , ενώ ο κόμβος  $u_k$  είναι ο γονέας του κόμβου  $k$  στο δένδρο  $T_2$ .

Για τη ρίζα  $r_1$  του δένδρου  $T_1$  ισχύει  $v_{r_1} = r_1$ , και αντίστοιχα για τη ρίζα  $r_2$  του δένδρου  $T_2$  ισχύει  $u_{r_2} = r_2$ . Για παράδειγμα, η ακολουθία εισόδου

$\langle N, v_0, v_1, v_2, \dots, v_{N-1}, u_0, u_1, u_2, \dots, u_{N-1} \rangle$   
 $= \langle 12, 11, 11, 4, 3, 3, 11, 11, 4, 6, 4, 6, 3, 8, 10, 8, 0, 2, 0, 10, 2, 8, 2, 2, 0 \rangle$

αντιστοιχεί στα δύο δένδρα του παρακάτω σχήματος.



Η ρουτίνα `main()` στο πρόγραμμα `TreeIso.java` αποθηκεύει τους γονείς  $v_0, v_1, v_2, \dots, v_{N-1}$  του δένδρου  $T_1$  σε ένα πίνακα `parent1` με  $N$  ακέραιους, όπου `parent1[k] = v_k` για  $0 \leq k \leq N - 1$ . Ομοίως, αποθηκεύει τους γονείς  $u_0, u_1, u_2, \dots, u_{N-1}$  του δένδρου  $T_2$  σε ένα πίνακα `parent2` με  $N$  ακέραιους, όπου `parent2[k] = u_k` για  $0 \leq k \leq N - 1$ .

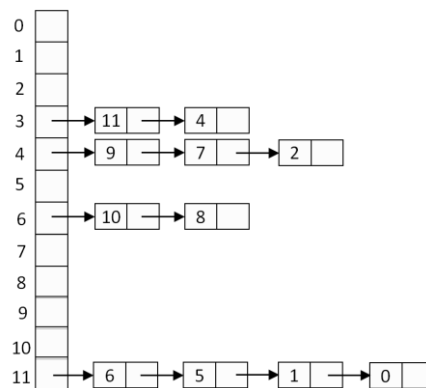
## Πανεπιστήμιο Ιωαννίνων – Τμήμα Μηχανικών Η/Υ και Πληροφορικής Δομές Δεδομένων [ΜΥΥ303] – Χειμερινό Εξάμηνο 2024

Στο παραπάνω παράδειγμα έχουμε  $\text{parent1} = [11, 11, 4, 3, 3, 11, 11, 4, 6, 4, 6, 3]$  και  $\text{parent2} = [8, 10, 8, 0, 2, 0, 10, 2, 8, 2, 2, 0]$ . Προσέξτε ότι για την ρίζα  $r_1 = 3$  του δένδρου  $T_1$  έχουμε  $\text{parent1}[3] = 3$  και για την ρίζα  $r_2 = 8$  του δένδρου  $T_2$  έχουμε  $\text{parent2}[8] = 8$ .

### Κατασκευή δένδρου

Συμπληρώστε στο αρχείο `RootedTree.java` την ακόλουθη μέθοδο κατασκευής ενός δένδρου. (Δείτε το συνοδευτικό αρχείο «Υλοποίηση δένδρου με λίστες παιδιών».)

`RootedTree(int N, int[] parent)` Λαμβάνει ως ορίσματα το πλήθος των κόμβων  $N$  και ένα πίνακα `parent` και κατασκευάζει τις λίστες των παιδιών κάθε κόμβου στο δένδρο. Μπορείτε να χρησιμοποιήσετε όποια δομή θέλετε για την αποθήκευση της κάθε λίστας παιδιών, π.χ. λίστες γειτνίασης, όπως η ακόλουθη που αντιστοιχεί στο δένδρο  $T_1$  του παραπάνω σχήματος.



Το αρχείο **In.java** σας δίνεται έτοιμο και δε χρειάζεται κάποια αλλαγή. Θα πρέπει να υλοποιήσετε τις μεθόδους σας στα αρχεία **RootedTree.java** και **RootedTreeIso.java**, αλλά μπορείτε να ορίσετε και δικές σας δομές σε άλλα αρχεία Java αν το επιθυμείτε.

### Εκτέλεση Προγραμμάτων

Για να εκτελέσετε το πρόγραμμα σας `RootedTreeIso.java` με είσοδο το αρχείο `twoTreesA.txt`, δώστε στη γραμμή εντολών `java RootedTreeIso < twoTreesA.txt`. Εκτελέστε το συμπληρωμένο πρόγραμμα σας `RootedTreeIso.java` με είσοδο τα αρχεία `twoTreesA.txt` και `twoTreesB.txt`. Το πρόγραμμα ελέγχει τα δύο δένδρα με ρίζα που δίνονται στην είσοδο είναι ισομορφικά, και αν είναι τυπώνει μια κατάλληλη συνάρτηση ισομορφισμού. Αποθηκεύστε τα αποτελέσματα της κάθε εκτέλεσης στο αρχείο `lab3results.txt`.

### Παραδοτέα

Ανεβάστε στο eCourse ένα zip αρχείο με τα τελικά προγράμματα σας σε Java καθώς και με το αρχείο των αποτελεσμάτων `lab3results.txt`. Το zip αρχείο πρέπει να έχει όνομα που περιλαμβάνει το group του εργαστηρίου στο οποίο έχετε τοποθετηθεί καθώς και από το ID της ομάδας σας (π.χ., `G1_ID1.zip`).