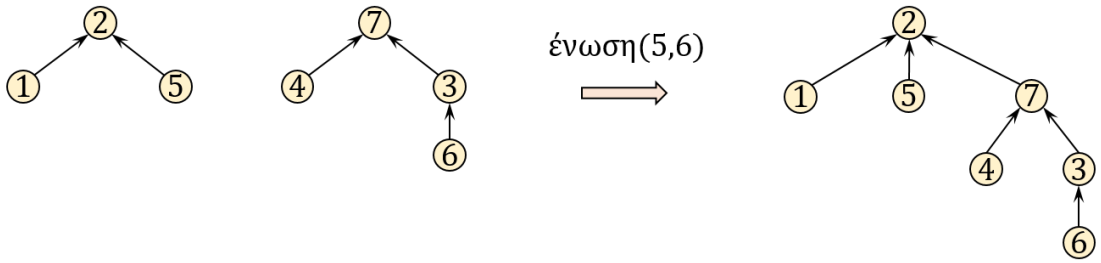
Πανεπιστήμιο Ιωαννίνων – Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Δομές Δεδομένων [ΜΥΥ303] – Χειμερινό Εξάμηνο 2020

2η Εργαστηριακή Άσκηση
Δομές Εύρεσης-Ένωσης
Παράδοση έως Πέμπτη 5/11, 12:00 από το eCourse

ΠΡΟΣΟΧΗ: Γράψτε σε κάθε αρχείο που παραδίδετε τα ονόματα και τους Α.Μ. των μελών της ομάδας σας. Συμπεριλάβετε όλα τα αρχεία σας (κώδικας Java και lab2results.txt) σε ένα zip αρχείο το οποίο να έχει το όνομα ενός μέλους της ομάδας σας με λατινικούς χαρακτήρες.

Σε αυτήν την άσκηση θα υλοποιήσετε και θα συγκρίνετε δύο δομές δεδομένων εύρεσης-ένωσης για τη διαχείριση ξένων συνόλων. Και στις δύο δομές αναπαριστούμε τα σύνολα ως δένδρα με ρίζα τον εκάστοτε αντιπρόσωπο του συνόλου. Στην παρακάτω εικόνα αρχικά έχουμε δύο σύνολα: το πρώτο είναι το {1,2,5} με αντιπρόσωπο το 2 και το δεύτερο είναι το {3,4,6,7} με αντιπρόσωπο το 7.



Η αποθήκευση των συνόλων γίνεται μέσω ενός πίνακα Pi όπου ο γονέας ενός αντικειμένου v δίνεται από την τιμή Pi[v]. Πχ, στο παράδειγμα του σχήματος θα έχουμε Pi[1] == 2 και Pi[6]==3. Αν το στοιχείο v είναι ο αντιπρόσωπος του συνόλου του τότε Pi[v]==v.

Δομή Γρήγορης Ένωσης

Η πρώτη δομή, την οποία πρέπει να συμπληρώσετε στο αρχείο SimpleUnionFind.java, υλοποιεί το βασικό αλγόριθμο γρήγορης ένωσης με τις ακόλουθες συναρτήσεις:

SimpleUnionFind(int N)	Μέθοδος κατασκευής (constructor) της δομής εύρεσης-ένωσης. Αρχικοποιεί τη δομή για N στοιχεία, τα οποία είναι οι ακέραιοι $\{1,2,\dots,N\}$. Αρχικά κάθε στοιχείο k βρίσκεται από μόνο του σε ένα σύνολο $\{k\}$ (με αντιπρόσωπο τον εαυτό του) οπότε θέτουμε Pi[k] = k.
int find(int v)	Επιστρέφει τον αντιπρόσωπο (ρίζα) του συνόλου που περιέχει το αντικείμενο v.
void unite(int v, int u)	Αν τα αντικείμενα v και u βρίσκονται σε διαφορετικά σύνολα τότε ενώνει τα δύο αυτά σύνολα.
int setCount()	Επιστρέφει το πλήθος των συνόλων στη δομή. (Αρχικά έχουμε N σύνολα, όσα είναι και τα στοιχεία της δομής.)

Η μέθοδος SimpleUnionFind() σας δίνεται έτοιμη, και επομένως θα πρέπει να υλοποιήσετε κατάλληλα τις υπόλοιπες μεθόδους.

Δομή Γρήγορης Ένωσης με Σταθμισμένη Ένωση και Συμπίεση Διαδρομής

Συμπληρώστε στο αρχείο UnionFind.java το βελτιωμένο αλγόριθμο ο οποίος χρησιμοποιεί σταθμισμένη ένωση και συμπίεση διαδρομής. Δίνεται έτοιμη η μέθοδος αρχικοποίησης:

UnionFind(int N)	Μέθοδος κατασκευής (constructor) της δομής εύρεσης-ένωσης. Αρχικοποιεί τη δομή για N στοιχεία, τα οποία είναι οι ακέραιοι $\{1,2,\dots,N\}$. Αρχικά κάθε στοιχείο k βρίσκεται από μόνο του σε ένα σύνολο $\{k\}$ (με αντιπρόσωπο τον εαυτό του) οπότε θέτουμε $\text{Pi}[k] = k$ και $\text{size}[k] = 1$.

Για τη σταθμισμένη ένωση πρέπει να τροποποιήσετε τη unite(int v, int u) ώστε να λαμβάνει υπόψη το μέγεθος κάθε συνόλου. Μπορείτε να ορίσετε ένα πίνακα size[] ο οποίος θα αποθηκεύει το πλήθος των αντικειμένων ενός συνόλου ως εξής. Αν το στοιχείο v είναι ο αντιπρόσωπος του συνόλου του (δηλαδή ισχύει $\text{Pi}[v] == v$) τότε το σύνολο περιέχει $\text{size}[v]$ στοιχεία. Αρχικά $\text{size}[v] = 1$ για όλα τα αντικείμενα v. Αν ενώσουμε τα σύνολα με αντιπροσώπους p και q με $\text{size}[p] \geq \text{size}[q]$ τότε θα έχουμε $\text{size}[p] = \text{size}[p] + \text{size}[q]$.

Το αποτέλεσμα της find(v) με συμπίεση διαδρομής είναι η αλλαγή του γονέα $\text{Pi}[k]$ για όλα τα αντικείμενα k που βρίσκονται στο μονοπάτι από το v έως τη ρίζα του δένδρου (τον αντιπρόσωπο του συνόλου) που περιέχει το v. Ο νέος γονέας αυτών των κόμβων είναι ο αντιπρόσωπος του συνόλου.

Εκτέλεση Προγραμμάτων

Η μέθοδος main() στα αρχεία SimpleUnionFind.java και UnionFind.java εκτελεί μια ακολουθία ενώσεων για $N = 16$ στοιχεία και στη συνέχεια τυπώνει τον γονέα του κάθε αντικειμένου στο δάσος εύρεσης-ένωσης που προκύπτει. Εκτελέστε τα δύο αυτά προγράμματα και αποθηκεύστε τα αποτελέσματα της εκτέλεσης τους στο αρχείο lab2results. txt.

Αφού βεβαιωθείτε ότι οι υλοποιήσεις σας είναι σωστές, συγκρίνετε την απόδοση τους με τη βοήθεια του προγράμματος RandomUnions.java. Το πρόγραμμα αυτό εκτελεί μια τυχαία ακολουθία ενώσεων για N αντικείμενα μέχρι να ενωθούν όλα σε ένα σύνολο. Χρησιμοποιεί πρώτα την απλή δομή εύρεσης-ένωσης (από το αρχείο SimpleUnionFind.java) και στη συνέχεια επαναλαμβάνει την ίδια διαδικασία χρησιμοποιώντας την πιο προηγμένη δομή εύρεσης-ένωσης (από το αρχείο UnionFind.java). Στο τέλος της κάθε εκτέλεσης τυπώνει το πλήθος των ενώσεων που πραγματοποιήθηκαν και τον αντίστοιχο χρόνο (σε ms) που χρειάστηκε η δομή.

Εκτελέστε το πρόγραμμα RandomUnions.java για $N = 10000$, 20000, 40000 και 80000 στοιχεία και αποθηκεύστε τα αποτελέσματα της εκτέλεσης του στο αρχείο lab2results. txt. Για να εκτελέσετε το RandomUnions.java για $N = 10000$ στοιχεία, δώστε στη γραμμή εντολών

```
java RandomUnions 10000
```

Ομοίως και για τις άλλες τιμές του N.

Παραδοτέα

Ανεβάστε στο eCourse ένα zip αρχείο με τα τελικά σας προγράμματα SimpleUnionFind.java και UnionFind.java, το έτοιμο πρόγραμμα RandomUnions.java, καθώς και με το αρχείο των αποτελεσμάτων lab2results. txt. Το zip αρχείο πρέπει να έχει το όνομα ενός μέλους της ομάδας σας με λατινικούς χαρακτήρες.