Πανεπιστημιο Πελοποννησου
Τμημα Πληροφορικης και Τηλεπικοινωνιων

Προηγμένα θέματα προγραμματισμού — ΣΤ′ εξάμηνο έτους 2020–2021
Διδάσκων: **Ν. Πλατής**

Άσκηση 5
Συλλογές της Java

Αραιοί πίνακες

Ένας **αραιός πίνακας** (sparse matrix) είναι στα μαθηματικά ένας πίνακας ο οποίος έχει μεγάλο ποσοστό μηδενικών στοιχείων· συνήθως πρόκειται για πίνακες μεγάλων διαστάσεων (π.χ. 1000×1000) των οποίων μηδενικά μπορεί να είναι ακόμα και το 90% των στοιχείων.

Είναι προφανές ότι ένας αραιός πίνακας μπορεί να αναπαρασταθεί πλήρως στον υπολογιστή αν είναι γνωστά μόνο τα μη μηδενικά στοιχεία του (και οι θέσεις τους) με μεγάλη οικονομία σε μνήμη. Για παράδειγμα, για έναν πίνακα 1000×1000 πραγματικών αριθμών (`double`) χρειάζονται στην Java $1000 \times 1000 \times 8 = 8.000.000$ bytes· αν όμως μόνο το 10% των στοιχείων είναι μη μηδενικά και για καθένα από αυτά αποθηκεύεται η τιμή (`double`), η γραμμή και η στήλη του (`int`), απαιτώντας $8+4+4 = 16$ bytes, τότε αρκούν $1000 \times 1000 \times 0{,}1 \times 16 =$ 1.600.000 bytes για όλο τον πίνακα.

Τέτοιοι αραιοί πίνακες παρουσιάζονται συχνά σε πρακτικές εφαρμογές (π.χ. σε επίλυση μεγάλων συστημάτων μερικών διαφορικών εξισώσεων) και για το λόγο αυτό έχουν αναπτυχθεί διάφοροι τρόποι αναπαράστασής τους καθώς και αλγόριθμοι χειρισμού τους.

Δύο απλοί (αν και όχι ιδιαίτερα αποτελεσματικοί για όλες τις λειτουργίες) τρόποι αναπαράστασης αραιών πινάκων είναι οι ακόλουθοι:

List of lists (LIL) Για κάθε γραμμή του πίνακα χρησιμοποιείται μία λίστα, της οποίας τα στοιχεία περιέχουν τα στοιχεία (**στήλη και τιμή**) των μη μηδενικών στοιχείων της αντίστοιχης γραμμής. Συνηθίζεται αυτές οι λίστες να είναι **ταξινομημένες ως προς τη στήλη των στοιχείων** για ταχύτερη αναζήτηση των στοιχείων του πίνακα.

Έστω για παράδειγμα ο αραιός πίνακας

$$\begin{bmatrix} 0 & 0 & 3 & 7 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 5 & 0 & 0 \end{bmatrix}$$

Στη μορφή LIL θα αναπαρασταθεί με μία λίστα τεσσάρων λιστών ως εξής:

0	$(2,3) - (3,7)$
1	$(0,1) - (4,2) - (5,3)$
2	
3	$(1,9) - (3,5)$

Dictionary of keys (DOK) Οι μη μηδενικές τιμές του πίνακα αποθηκεύονται σε μία δομή απεικόνισης (map) με *κλειδί* τις συντεταγμένες (**γραμμή, στήλη**) και *τιμή* την **τιμή** του στοιχείου του πίνακα.

Για παράδειγμα, ο παραπάνω αραιός πίνακας θα αναπαρασταθεί στη μορφή DOK με μία δομή απεικόνισης που θα περιέχει τα εξής ζεύγη κλειδιών–τιμών:

$$(0,2) \to 3 \qquad (1,5) \to 3$$
$$(0,3) \to 7 \qquad (3,1) \to 9$$
$$(1,0) \to 1 \qquad (3,3) \to 5$$
$$(1,4) \to 2$$

Ζητούμενα

Στην άσκηση αυτή καλείστε να αναπτύξετε κλάσεις που θα αναπαριστούν αραιούς πίνακες και θα υλοποιούν τις δύο παραπάνω μορφές αναπαράστασής τους. Αναλυτικά υλοποιήστε τα ακόλουθα:

1. Διεπαφή `SparseMatrix`

Αυτή η διεπαφή (interface) αντιπροσωπεύει μία οποιαδήποτε υλοποίηση αραιού πίνακα με στοιχεία τύπου `double`. Περιέχει τις ακόλουθες μεθόδους:

`int rowCount()`
Επιστρέφει το πλήθος γραμμών του πίνακα.

`int colCount()`
Επιστρέφει το πλήθος στηλών του πίνακα.

`double get(int r, int c)`
Επιστρέφει το στοιχείο του πίνακα που βρίσκεται στη γραμμή `r` και στήλη `c`. Προκαλεί μία εξαίρεση τύπου `IndexOutOfBoundsException` με κατάλληλο επεξηγηματικό μήνυμα αν η γραμμή ή η στήλη είναι εκτός ορίων (παρόμοια με την μέθοδο `get()` της `ArrayList`).

`void set(int r, int c, double element)`
Θέτει το στοιχείο της θέσης (`r`, `c`) του αραιού πίνακα στην τιμή `element`. Αν η τιμή είναι μηδέν (με ακρίβεια 10^{-5}) καλεί την επόμενη μέθοδο `zero()`. Προκαλεί επίσης μία εξαίρεση τύπου `IndexOutOfBoundsException` με κατάλληλο επεξηγηματικό μήνυμα αν η γραμμή ή η στήλη είναι εκτός ορίων.

`void zero(int r, int c)`
Μηδενίζει το στοιχείο στη θέση (`r`, `c`) του αραιού πίνακα (προφανώς το αφαιρεί εντελώς από τις αντίστοιχες δομές). Προκαλεί επίσης μία εξαίρεση τύπου `IndexOutOfBoundsException` με κατάλληλο επεξηγηματικό μήνυμα αν η γραμμή ή η στήλη είναι εκτός ορίων.

`void clear()`
Αφαιρεί όλα τα στοιχεία του αραιού πίνακα.

`boolean isEmpty()`
Επιστρέφει `true` αν ο αραιός πίνακας είναι κενός (δεν έχει κανένα στοιχείο μη μηδενικό) ή `false` διαφορετικά.

`String toString()`
Επιστρέφει μία συμβολοσειρά με τα (μη μηδενικά) στοιχεία του πίνακα. Η συμβολοσειρά πρέπει να είναι της μορφής:

```
[ (0, 2: 3,00) (0, 3: 7,00) (1, 0: 1,00) (1, 4: 2,00) (1, 5: 3,00)
(3, 1: 9,00) (3, 3: 5,00) ]
```

Τα στοιχεία του πίνακα θα πρέπει να δίνονται **ταξινομημένα** κατά γραμμές και (για κάθε γραμμή) κατά στήλες.

2. Κλάση `SparseMatrixLIL`

Κλάση που υλοποιεί την διεπαφή `SparseMatrix` αναπαριστώντας τον αραιό πίνακα στη μορφή LIL.

Εκτός από τις μεθόδους της διεπαφής, θα περιέχει μία **μέθοδο δημιουργίας** που θα δέχεται ως παραμέτρους το πλήθος γραμμών και το πλήθος στηλών του αραιού πίνακα.

Αν και η αναπαράσταση καλείται «λίστα από λίστες», μπορείτε να χρησιμοποιήσετε οποιεσδήποτε δομές κρίνετε καταλληλότερες για τις λίστες των γραμμών καθώς και για τη λίστα που θα τις αποθηκεύει. Σε κάθε λίστα γραμμής, τα στοιχεία θα πρέπει μετά από κάθε λειτουργία να είναι **ταξινομημένα** ως προς τη στήλη τους.

Για τα στοιχεία των λιστών των γραμμών δημιουργήστε μία ιδιωτική **εσωτερική** κλάση `ElementInfo`, κάθε αντικείμενο της οποίας θα περιέχει τη στήλη και την τιμή ενός στοιχείου του πίνακα. Ανάλογα με τις δομές που θα χρησιμοποιήσετε για τις λίστες της αναπαράστασης, μπορεί να χρειαστεί να εμπλουτίσετε την κλάση αυτή με άλλες μεθόδους.

3. Κλάση `SparseMatrixDOK`

Κλάση που υλοποιεί την διεπαφή `SparseMatrix` αναπαριστώντας τον αραιό πίνακα στη μορφή DOK.

Εκτός από τις μεθόδους της διεπαφής, θα περιέχει μία **μέθοδο δημιουργίας** που θα δέχεται ως παραμέτρους το πλήθος γραμμών και το πλήθος στηλών του αραιού πίνακα.

Για την κλάση αυτή θα πρέπει να επιλέξετε μία από τις υλοποιήσεις της διεπαφής `Map` της Java.

Για τα κλειδιά της δομής απεικόνισης δημιουργήστε μία ιδιωτική εσωτερική κλάση `ElementPos`, κάθε αντικείμενο της οποίας θα περιέχει τη γραμμή και τη στήλη ενός στοιχείου του πίνακα. Ανάλογα με την υλοποίηση της `Map` που θα επιλέξετε, θα χρειαστεί να προσθέσετε άλλες μεθόδους στην κλάση αυτή.

4. Κλάση `SparseMatrices`

Κλάση που θα περιέχει στατικές μεθόδους για πράξεις μεταξύ αραιών πινάκων. Υλοποιήστε ενδεικτικά μόνο την εξής μέθοδο:

`SparseMatrix add(SparseMatrix a, SparseMatrix b)`
: Υλοποιεί την πρόσθεση δύο αραιών πινάκων (ίδιων διαστάσεων), η οποία ως γνωστόν, ανάγεται απλώς στην πρόσθεση των αντίστοιχων στοιχείων τους. Η μέθοδος δημιουργεί και επιστρέφει τον αραιό πίνακα στον οποίο αποθηκεύεται το άθροισμα των πινάκων `a` και `b`· επιστρέφει `null` αν οι πίνακες δεν έχουν τις ίδιες διαστάσεις.[1]

5. Κλάση `Main`

Η κλάση που θα περιέχει τη μέθοδο `main()` της εφαρμογής, η οποία θα δημιουργεί τουλάχιστον έναν πίνακα της κάθε κλάσης (`SparseMatrixLIL` και `SparseMatrixDOK`) και θα δοκιμάζει όλες τις μεθόδους.

[1] Αυτό ζητείται μόνο για απλοποίηση, σωστότερο θα ήταν σε τέτοια περίπτωση η μέθοδος να προκαλεί μία εξαίρεση.

Βοηθητικά στοιχεία

- Υπενθυμίζεται ότι για να ελέγξουμε αν ένας πραγματικός αριθμός `x` είναι κοντά στο μηδέν με ακρίβεια 10^{-5} πρέπει να ελέγξουμε την απόλυτη τιμή του: $|x| < 10^{-5}$ ή `Math.abs(x) < 1e-5`.
- Όπως και στην προηγούμενη άσκηση, βασικό ζητούμενο είναι να επιλέξετε τις **βέλτιστες** δομές ανάλογα με τα ζητούμενα κάθε κλάσης. Αν επιλέξετε κάποια δομή αλλά δεν μπορείτε να βρείτε κάποια λειτουργία που σας είναι απαραίτητη στα παραδείγματα που έχουμε κάνει, ανατρέξτε στην τεκμηρίωσή της (https://docs.oracle.com/javase/8/docs/api/index.html).

Διαδικαστικά

- Έχει δημιουργηθεί μία «εργασία» στο GitHub, στο URL:

  https://classroom.github.com/g/SFzPdnD4

  Ακολουθείτε την ίδια διαδικασία με τις προηγούμενες ασκήσεις για την υποβολή της άσκησης.
- Χρησιμοποιούνται αυτόματα οι ομάδες των προηγούμενων εργασιών. Αν υπάρχει σοβαρός λόγος αλλαγής ομάδας (π.χ. αποχώρηση ατόμου) παρακαλώ επικοινωνήστε μαζί μου.
- Η άσκηση είναι **προαιρετική**. Θα μετρήσει περίπου για 5% του τελικού βαθμού (τα ακριβή ποσοστά για κάθε άσκηση θα ανακοινωθούν αργότερα).
- Για την παράδοση της εργασίας σας, απλώς θα κάνετε push στο GitHub το project σας.
- **Προθεσμία παράδοσης:** Τετάρτη 9 Ιουνίου, 23:59.
- Για οποιαδήποτε απορία χρησιμοποιήστε την αντίστοιχη **περιοχή συζητήσεων** στο e-class.