

6^η Προγραμματιστική Εργασία

Ημερομηνία παράδοσης: 4.12.2022
Συζήτηση λύσης - επίλυση αποριών: 7.12.2022

Στόχοι: Εξοικείωση με τη χρήση δισδιάστατων πινάκων και με τις μεθόδους ταξινόμησης.

Άσκηση 1.

Θέλουμε να συμπληρώσουμε έναν δισδιάστατο 10×10 πίνακα με τυχαίες ακέραιες τιμές στο διάστημα $[1, 100]$ και στη συνέχεια να υπολογίσουμε το ελάχιστο στοιχείο κάθε γραμμής και κάθε στήλης του πίνακα. Πιο συγκεκριμένα:

Για την υλοποίησή σας, αρχικά θα χρειαστεί να ορίσετε μια καθολική ακέραια σταθερά N , η οποία αντιστοιχεί στη διάσταση του πίνακα (10 όπως αναφέρθηκε παραπάνω). Ακολουθώντας, υλοποιήστε τις παρακάτω συναρτήσεις:

- Η συνάρτηση `fillTable(int A[N][N])` γεμίζει με τυχαίες ακέραιες τιμές στο διάστημα $[1, 100]$ τον δοθέν $N \times N$ πίνακα A (με χρήση της συνάρτησης `rand()`).
- Η συνάρτηση `minRow(int A[N][N], int min[N])` δέχεται ως παραμέτρους έναν δισδιάστατο $N \times N$ πίνακα A , και έναν μονοδιάστατο $1 \times N$ πίνακα min και αποθηκεύει στην i -οστή θέση του πίνακα min το ελάχιστο της i -οστής γραμμής του πίνακα A .
- Η συνάρτηση `minColumn(int A[N][N], int min[N])` δέχεται ως παραμέτρους έναν δισδιάστατο $N \times N$ πίνακα A , και έναν μονοδιάστατο $1 \times N$ πίνακα min και αποθηκεύει στην i -οστή θέση του πίνακα min το ελάχιστο της i -οστής στήλης του πίνακα A .

Ενσωματώστε τις συναρτήσεις σε κατάλληλο δοκιμαστικό πρόγραμμα, το οποίο θα τις καλεί και θα εμφανίζει τα αποτελέσματά τους.

Άσκηση 2.

Στην τάξη μελετήσαμε διάφορους τρόπους ταξινόμησης ενός μονοδιάστατου πίνακα (με συγκρίσεις). Στην άσκηση αυτή θα υποθέσουμε ότι τα προς ταξινόμηση στοιχεία του πίνακα είναι ακέραιοι αριθμοί στο διάστημα $[0, N-1]$, όπου το N είναι μια καθολική σταθερά (έστω 100). Η ταξινόμηση ενός τέτοιου πίνακα A μπορεί να γίνει χωρίς να κάνουμε κάποια σύγκριση ως εξής:

- Δοθέντος του πίνακα A , δημιουργούμε ένα $1 \times N$ μονοδιάστατο πίνακα B , τον οποίο τον αρχικοποιούμε ως εξής: Το στοιχείο $B[i]$ στη θέση i του πίνακα του πίνακα B έχει την τιμή j αν και μόνο αν ο πίνακας A έχει ακριβώς j στοιχεία με τιμή i (όπου $0 \leq i \leq N-1$ και $0 \leq j \leq N$).
Παράδειγμα: Αν $N=8$, και $A = [1, 4, 1, 2, 7]$ τότε $B = [0, 2, 1, 0, 1, 0, 0, 1]$

- Παρατηρήστε ότι ο υπολογισμός του πίνακα B μπορεί να γίνει θέτοντας αρχικά όλα του τα στοιχεία στην τιμή 0 και ακολούθως κάνοντας μια διαπέραση του πίνακα A ανανεώνουμε τις τιμές του πίνακα B ως εξής: Για $i=0, \dots, N-1$, αν $A[i]=k$, τότε αυξάνουμε την τιμή του $B[k]$ κατά 1.
- Έχοντας υπολογίσει τον πίνακα B , τότε η ταξινόμηση του πίνακα A αντιστοιχεί στην εκτύπωση των στοιχείων του πίνακα B ως εξής: Για $i=0, \dots, N-1$ εκτυπώνουμε $B[i]$ φορές τον ακέραιο i . Για το παράδειγμα που δώσαμε παραπάνω, αυτό αντιστοιχεί σε: 1, 1, 2, 4, 7.

Να υλοποιήσετε την συνάρτηση `sortAndPrint(int[] A, int n, int max)`, η οποία δέχεται ως παράμετρο έναν πίνακα A μεγέθους n , του οποίου τα στοιχεία ανήκουν στο διάστημα $[0, \text{max}-1]$, και ακολουθώντας τον κανόνα που δόθηκε προηγουμένως εκτυπώνει στο τερματικό τον πίνακα A ταξινομημένο.

Ενσωματώστε τη συνάρτηση σε κατάλληλο δοκιμαστικό πρόγραμμα, το οποίο θα την καλεί και θα εμφανίζει το αποτέλεσμα της.