

ΑΝΑΦΟΡΑ (Μέρος Ε)

Μέρος Α:

Η κλάση City δημιουργεί στοιχεία τύπου city και χρησιμοποιεί αντίστοιχους setters και getters για να τοποθετηθούν κατάλληλα τα δεδομένα κάθε πόλης.

Στην main δημιουργείται ArrayList με πόλεις και διαβάζοντας το text αρχείο ανά σειρά αποθηκεύεται σε String η κάθε μια. Ύστερα με τους accessors τοποθετούνται τα στοιχεία στο κάθε city. Μετά ζητά έναν αριθμό για την εμφάνιση των αντίστοιχων πόλεων που θέλει να δει ο χρήστης και εάν είναι έγκυρος αποθηκεύονται στον πίνακα reduction οι αναγωγές στα κρούσματα.

Σύμφωνα με αυτόν γίνεται η ταξινόμηση εάν διαφέρουν οι τιμές αλλιώς γίνεται αλφαβητικά και αν και πάλι είναι ίδιες οι τιμές η διαδικασία ολοκληρώνεται από το id. Πιο συγκεκριμένα, θέτουμε ως άξονα το μεσσαίο στοιχείο του πίνακα και ύστερα αρχικοποιούνται δείκτες στα άκρα. Μετά βρίσκει τα στοιχεία που είναι μεγαλύτερα και μικρότερα από τον άξονα και εάν τα αριστερά είναι μεγαλύτερο από τα δεξιά τότε αντιστρέφει τα αντίστοιχα στοιχεία του πίνακα μαζί με τα ονόματα που χρειαστούμε αργότερα. Εάν τα δεξιά είναι ίσα με τα αριστερά τότε αν τα ονόματα δεν είναι ίδια γίνεται αλφαβητική ταξινόμηση αλλιώς ταξινομούνται από τα id και ακολουθούν και τα ονόματα. Η διαδικασία επαναλαμβάνεται για όλον τον πίνακα.

Στο τέλος εμφανίζονται οι k πόλεις με τα περισσότερα κρούσματα αναλογικά

Μέρος Β:

Στην κλάση αυτή υλοποιούνται οι μέθοδοι max, getmax και insert ανάλογα με τις υλοποιήσεις του εργαστηρίου. Η size επιστρέφει το μέγεθος το οποίο επηρεάζεται από την getmax, την insert και την remove, ενώ isEmpty ελέγχοντας το μέγεθος επιστρέφει μήνυμα αληθείας. Τέλος η remove ψάχνει να βρει

σε ποιο στοιχείο του heap υπάρχει αυτό το id ώστε να το αφαιρέσει. (η πολυπλοκότητα δεν είναι η επιθυμητή)

Μέρος Γ:

Στην `DynamicCovid_k_withPQ` εκτός της `main` υλοποιείται και η `getResult` η οποία επιστρέφει την τιμή για τα κρούσματα μετά την αναγωγή και τη στρογγυλοποίηση.

Ξεκινώντας η `main` διαβάζει τον αριθμό των πόλεων που πρέπει να εμφανίσει ταξινομημένα. Με παρόμοια διαδικασία με το Μέρος Α διαβάζει γραμμή γραμμή τα πρώτα k στοιχεία και τα αποθηκεύει στον σωρό κρατώντας όμως στην μεταβλητή `min` τα λιγότερα κρούσματα και στο `mincity` το αντικείμενο `city` που την έχει. Μετά συνεχίζεται η διαδικασία και για τις υπόλοιπες γραμμές συγκρίνοντας το κάθε νέο `city` με το `mincity`. Τέλος ανά 5 γραμμές τυπώνονται οι k πόλεις με τα περισσότερα κρούσματα.

Σε αυτή την περίπτωση έχει μειωθεί η πολυπλοκότητα καθώς δεν χρειάζεται να ανατρέχουμε σε όλα τα `city` για να τα ταξινομήσουμε όπως γίνεται παραπάνω.