ΕΡΓΑΣΙΑ 2-ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Α Ασκηση:

Η κλάση City υλοποιεί τη διεπαφή CityInterface και προσφέρει μια δομή για την αναπαράσταση πληροφοριών σχετικά με μια πόλη, όπως το αναγνωριστικό, το όνομα, τον πληθυσμό, και τα κρούσματα γρίπης.

Ιδιοτητες:

- ID: Το αναγνωριστικο της πολης
- name: Το ονομα της πολης
- population: Ο πληθυσμος της πολης
- InfluenzaCases:Ο αριθμος κρουσματων της πολης

Με τις μεθοδους get set για κάθε ιδιοτητα εχουμε προσβαση στις ιδιοτητες και μας επιτρεπεται αναγνωση και ενημερωση τους.

Με την compareTo συγκρινουμε δυο πολεις βασεις της πυκνοτητας της αυτό επιτυγχανεται μεσω της εντολης compare και μεσα στην compareTo παιρνουμε σαν περιπτωση αν είναι ισα σε πυκνοτητα οι πολεις να ελεγχουμε τα ονοματα.

Η compareTo καλειται city.compareTo(otherCity).

Με την calculateDestiny υπολογιζουμε την πυκνοτητα κρουσματων ανα 50.000 κατοικους.

Με βαση την παραπανω υλοπιηση προχωρησαμε στο Influenza_k για την ταξινομηση πολεων με βαση την πυκνοτητα κρουσματων.

Αυτό το πετυχαίνουμε διαβαζοντας τα δεδομενα από ένα txt αρχειο με την χρηση BufferedReader καθως κάθε γραμμη είναι για μια πολη

Η τακινομηση εχει υλοποιηθει με την χρηση της QuickSort,(ΑΜ 3200128 3200135),

Ο αλγοριθμος της QuickSort χρησιμοποιηται για την γρηγορη ταξινομηση των πολεων με βασει την πυκνοτητα,για την υλοποιηση εχουμε διαιρέσει την λιστα σε υπολιστες και καλουμε αναδρομικα την QuickSort.

Για να μπορεσει να τρεξει το προγραμμα πρεπει να του δωσουμε το txt αρχειο και των αριθμο των πολεων που θελουμε να εμφανισει μεσω command line.

Β Ασκηση:

Η κλάση PQ υλοποιεί μια προτεραιότητα ουρά (priority queue) με χρήση ενός πίνακα (heap). Η προτεραιότητα καθορίζεται από τη σχέση σύγκρισης των πόλεων με βάση την πυκνότητα των κρουσμάτων.

Ιδιοτητες:

- heap: Ο πίνακας που αναπαριστά την προτεραιότητα ουρά.
- size: Το πληθος των στοιχειων στην ουρα
- DEFAULT_CAPACITY: Η προκαθορισμένη χωρητικότητα του πίνακα.
- AUTOGROW_SIZE: Το πλήθος που αυξάνεται ο πίνακας κάθε φορά που χρειάζεται επέκταση.

Μεθοδοι:

- getHeap: Επιστρεφει τον πινακα heap
- size: επιστρεφει τον αριθμο στοιχειων στην ουρα
- grow: Αυτή η ι μέθοδος χρησιμοποιείται για να αυξήσει το μέγεθος του πίνακα heap όταν ο υπάρχον πίνακας είναι γεμάτος. Δημιουργεί ένα νέο πίνακα μεγαλύτερου μεγέθους και αντιγράφει τα στοιχεία από τον παλιό πίνακα στον νέο.
- swap: ανταλλασσει τα στοιχεια της θεσης i με τα στοιχεια της θεσεις j του πινακα
- isEmpty: ελεγχει αν ο πινακας είναι αδειος
- insert: Εισαγει μια πολη στην ουρα αυξανοντας το μεγεθος αν είναι αναγκαιο και καλοντας την swim για να διατηρηθει η ταξη
- min: Επιστρεφει την πολύ με την μικροτερη πικνοτητα (Πρωτο στοιχειο του πινακα)
- getmin: Επιστρεφει και αφαιρει την πολύ με την μικροτερη πικνοτητα
- remove: αφαιρεί την πόλη με το καθορισμένο αναγνωριστικό από την ουρά και εφαρμόζει τη μέθοδο sink για να διατηρήσει την τάξη
- swim: Εφαρμοζει τον αλγοριθμο αναδυσης για να διατηρηση την ταξη μετα από μια εισαγωγη

- sink: Εφαρμοζει τον αλγοριθμο καταδυσης για να διατηρηση την ταξη μετα από μια αφαιρεση

Γ Ασκηση:

• Σε αυτή τη ασκηση χρησιμοποιουμε αρχικα τους μεθοδους BufferedReader ,FileReaderαπό από τις βιβλιοθηκες

```
import java.io.FileNotFoundException;
import java.io.FileReader;, αντιστοιχα
```

• Διαβαζουμε γραμμη προς γραμμη το κειμενο,δημιιουργουμε σε
κάθε γραμμη μια λιστα με την κάθε λεξη της γραμμης κι δημιουργουμε κάθε φορα ένα αντικειμενο City με γνωρισματα τις
λεξεις από κάθε γραμμη όπως αναφερει η εκφωνηση.
• Κάθε αντικειμενο City το προσθετουμε στη Ουρα προτεραιοτητας
pq που εχουμε αρχικοποιησει.
• Αφοτου τελειωσουμε την αναγνωση του αρχειου ,εμαφανιζουμε
τις k χωρες με τα λιγοτερα κρουσματα ανα 50000 πληθυσμο επιλεγοντας κάθε φορα από την pq την ελαχιστη τιμη (με τη μεθοδο getMin()).

Δ Ασκηση:

• Σε αυτή τη ασκηση κανουμε αρχικα την αναγνωση του αρχειου
όπως στις προηγουμενες ασκησεις.
• Αρχικοποιουμε 2 ουρες προτεραιοτητας MaxPQ,MinPq οπου στη
πρωτη θα εχουμε τα μεγαλυτερες κι στη δευτερη τα μικροτερες
πυκνοτητες.
• Όταν το νέο αντικειμενο City εχει μεγαλυτερη πυκνοτητα από την
ελαχιστη τιμη της ουρας MinPq το τοποθετουμε εκει αλλιως στη ουρα προτ. MaxPQ.
• Ελεγχουμε αν δεν εχει χαλασει η ομοιομορφια στις 2 ουρες .
• Υπολογιζουμε το median ως εξης:

• Αν το μεγεθος των 2 ουρων είναι ισο τοτε,παιρνουμε το πηλικο
των 2 αθροισματων των κρουσματων κι των πληθυσμων των2 ελαχιστων χωρων σε πυκνοτητα από την MinPQ K MaxPQ διαιροντας το με το 2.
• Αν δεν είναι ιδιο το μεγεθος τοτε,παιρνουμε το πηλικο της ελαχιστης πυκνοτητας χωρα από την MinPq.