Μέρος Α

Στο αρχείο Influenza\_k.java έχουμε υλοποιήσει την συνάρτηση quicksort(City[]) cits, int start, int end) η οποία αποτελεί μια εξειδικευμένη εκδοχή του αλγορίθμου QuickSort, προκειμένου να μπορούμε να επεξεργαστούμε τα δεδομένα τύπου City.

Χαρακτηριστικά αλγορίθμου:

1. Time complexity: O(nlogn) - Space complexity: O(n)
2. Τελευταίο στοιχείο ως pivot
3. Ταξινόμηση τέτοια ώστε το πρώτο στοιχείο να έχει το μικρότερο πλήθος κρουσμάτων ανά 50,000 κατοίκους (Σε περίπτωση ισοβαθμίας μεταξύ πόλεων, θεωρούμε πιο υψηλά στην

κατάταξη το στοιχείο που προηγείται αλφαβητικά και σε περίπτωση που και τα ονόματα ταυτίζονται, προηγείται αυτό με το μικρότερο ID).

Μέρη Β-Γ

\*\*\*Β)

Η συνάρτηση remove(int id) έχει Time complexity -> O(logn). Προκειμένου όμως να είναι κάτι τέτοιο δυνατό και δεδομένου ότι η συνάρτηση sink(int index) και swim(int index) που χρησιμοποιούμε έχουν επίσης time complexity -> O(logn), τότε η αναζήτηση του στοιχείου προς διαγραφή θα πρέπει να γίνεται σε χρόνο καλύτερο του n. Στην προκειμένη περίπτωση χρησιμοποιούμε έναν πίνακα με 1000 θέσεις για τον οποίο ισχύσει το εξής:

1. Key -> Το id της χώρας (π.χ μια χώρα με id = 20 θα βρίσκεται στην θέση 20 του πίνακα)
2. Value -> Η θέση του στοιχείου στο οποίο αντιστοιχεί το id στον πίνακα heap (π.χ αν η χώρα με id 20 βρίσκεται στην θέση 3 του πίνακα heap, τότε θα ισχύει id[20]=3)

Με αυτό τον τρόπο βρίσκουμε το στοιχείο που ψάχνουμε σε χρόνο O(1), το αντικαθιστάμε με το τελευταίο και εκτελούμε sink και swim (δεν ξέρουμε αν το νέο στοιχείο θα είναι μικρότερο του πατέρα/μεγαλύτερο από τα παιδιά του) για να ξαναδημιουργήσουμε ένα σωστό heap.

Φυσικά η επιλογή των 1000 θέσεων δεν είναι αυθαίρετη, καθώς τα idE[1, 999] και άρα μπορεί να έχουμε το πολύ 999 ids (η επιπλέον θέση 0 δεν χρησιμοποιείται γιατί δεν υπάρχει id = 0).

Α\*?)

Όταν τρέχουμε το αρχείο Influenza\_k.java από το terminal θα πρέπει να δώσουμε ως όρισμα το k και το file path που περιέχει τις πόλεις και τα δεδομένα που θα χρησιμοποιήσουμε. Σε περίπτωση που δοθούν περισσότερα ή λιγότερα ορίσματα το πρόγραμμα τερματίζει με exit code 1 και ένα αντίστοιχο μήνυμα εμφανίζεται στον χρήστη. Αν το k δεν είναι integer το πρόγραμμα και πάλι τερματίζει, ενώ αν όλα τα ορίσματα είναι σωστά τότε τα αποθηκεύουμε απλώς σε 2 μεταβλητές και τα χρησιμοποιούμε κανονικά στο πρόγραμμα.