Δομές Δεδομένων

Εργασία 2

ΤΟΣΚΟΛΛΑΡΙ ΡΟΝΑΛΝΤ

p3160244

Μέρος Α: Στην κλάση Top_k υπάρχουν 4 μέθοδοι όπου την αποτελούν. Στην main μέσα στην οποία γίνεται η επεξεργασία του αρχείου με τα τραγούδια, φτιάχνουμε τον πίνακα που θα έχει τα τραγούδια τον οποίο θα επεξεργαστούμε μέσω της quicksort. Αρχικά διαβάζουμε το αρχείο μέχρι να μην υπάρχουν άλλα τραγούδια για διάβασμα, άρα διαβάζαμε συνολικά Ν τραγούδια. Μετά καλούμε την quicksort. Η quicksort με την σειρά της παίρνει 3 ορίσματα, πρώτα τον πίνακα που έχει να επεξεργαστεί δεύτερον την αρχή του και τρίτον το τέλος του πίνακα. Κάνουμε έλεγχο στα όρια που έχουμε πάρει. Αν η αρχή είναι μεγαλύτερη ή ίση από το τέλος τότε δεν πειράζουμε τον πίνακα, σε αντίθετη περίπτωση χωρίζουμε τον πίνακα στην μέση καλώντας την μέθοδο partition οποία παίρνει τα ίδια ορίσματα με την quicksort, ορίζει ως pivot το τελευταίο στοιχείο της λίστας διότι έχουμε τυχαία δεδομένα και η επιλογή του pivot δεν είναι τόσο σημαντική. Ορίζει σε μια μεταβλητή (int i) να είναι ίση με την αρχή-1(p-1) και μια μεταβλητή (int j) ίση με το τέλος του πίνακα. Η partition φροντίζει πάντα να ισχύει ότι a[p],…..,a[i-1] είναι <= a[i] και ότι a[i+1], …., a[r] είναι >= a[i]. Επιστρέφει μια μεταβλητή που δηλώνει την μέση του πίνακα. Τέλος η quicksort καλεί αναδρομικά τον εαυτό της 2 φορές μια για πρώτο μισό και μία για το άλλο μισό. Αυτό συνεχίζεται αναδρομικά μέχρι να πάρουμε έναν ταξινομημένο πίνακα.

Μέρος Β: Η μέθοδος remove παίρνει ως όρισμα το id ενός τραγουδιού, αρχικά ελέγχει εάν υπάρχουν ακόμα αντικείμενα στο PQ. Σε περίπτωση που έχουμε ακόμα αντικείμενα κάνουμε σειριακή αναζήτηση για όλο το μέγεθος της heap. Αποθηκεύουμε και ελέγχουμε ένα ένα τα αντικείμενα, αν είναι αυτό το αντικείμενο που ψάχνουμε τότε βγαίνουμε από την for, σε αντίθετη περίπτωση απλός κρατάμε την θέση που βρισκόμαστε. Εάν δεν υπάρχει το αντικείμενο τότε γυρνάμε null αλλιώς αποκαθιστούμε την ιδιότητα της σωρού και επιστρέφουμε το αντικείμενο που ψάχνουμε.

Μέρος Γ: Ξεκινάμε το πρόγραμμα ζητώντας από τον χρήστη να εισάγει τον αριθμό των τραγουδιών που θέλει να του εμφανίσουμε, αργότερα φτιάχνουμε τα αντικείμενα και τις μεταβλητές που θα χρειαστούμε, φτιάχνουμε και την ουρά προτεραιότητας που θα χρησιμοποιήσουμε η οποία θα αποθηκεύει αντικείμενα τύπου Song και βάζουμε ως ορίσματα το μέγεθος(2) και το Comparator(SongComparator). Αργότερα διαβάζουμε το αρχείο μέχρι να φτάσουμε στο τέλος του. Σε κάθε γραμμή του αρχείου που διαβάζουμε γίνεται το κατάλληλο tokenize έτσι ώστε να αποθηκεύσουμε τις πληροφορίες του συγκεκριμένου τραγουδιού. Σε περίπτωση που δεν είναι ορθή η γραμμή που διαβάσαμε απλά την αγνοούμε και προχωράμε στην επόμενη γραμμή. Αν όμως είναι ορθό τραγούδι, ελέγχουμε εάν η ουρά μας έχει φτάσει στο μέγιστο αριθμό των τραγουδιών που μπορεί να αποθηκεύσει, αν έχει φτάσει τότε συγκρίνουμε το max της ουράς με το τραγούδι που διαβάζουμε τώρα, εάν το τραγούδι είναι μεγαλύτερο από το max, αφαιρούμαι το max και εισάγουμε το τραγούδι. Αν η ουρά μπορεί να αποθηκεύσει και άλλα τραγούδια απλά κάνουμε insert το τραγούδι. Τέλος απλά εμφανίζουμε τα τραγούδια σε φθίνουσα σειρά.

Μέρος Γ(πολυπλοκότητα): Το πρόγραμμα θα κάνει σίγουρα κ εισαγωγές τραγουδιών, όπου κ θα το δίνει ο χρήστης. Για κάθε εισαγωγή ο χρόνος που χρειάζεται είναι O(logn). Τώρα σε περίπτωση

που ο χρήστης δώσει κ μικρότερο από το συνολικό αριθμό των τραγουδιών τότε θα γίνουν στην χειρότερη περίπτωση maxSongs – k εξαγωγές. Για κάθε εξαγωγή ο χρόνος που χρειάζεται είναι O(logn). Τέλος απλά παίρνουμε κ φορές τα max τραγούδια.

Μέρος Δ: Στο πρόγραμμα αυτό η εύρεση του median γίνεται με την χρήση 2 PQs. Το πρώτο αποθηκεύει σαν max το ‘’μικρότερο’’ τραγούδι ενώ η δεύτερη αποθηκεύει σαν max το ‘’μεγαλύτερο’’ τραγούδι. Διαβάζουμε το αρχείο μέχρι το τέλος του. Κατά την ανάγνωση κάνουμε έλεγχο αν το τραγούδι που έχουμε τώρα είναι μεγαλύτερο από το median (αρχικό median = 0) και μπαίνει στο ανάλογο PQ. Η εισαγωγή στοιχείου σε οποιαδήποτε από τις δύο σωρούς θέλει χρόνο O(logn). Για να βρούμε το median τώρα ελέγχουμε το μέγεθος που έχουν τα PQs και παίρνουμε το median σε αυτό που έχει το μεγαλύτερο μέγεθος. Μόλις το βρούμε απλά εξάγουμε το max του συγκεκριμένου PQ.