

Κορναράκης Κωνσταντίνος-Ιωάννης p3170074

Σπινάρης Γεώργιος p3170151

ΜΕΡΟΣ Α

Χρησιμοποιούμε τον αλγόριθμο ταξινόμησης Quicksort. Ορίζουμε ως πύβος τον μεσαίο αριθμό του πίνακα. Διαιρούμε τον πίνακα σε δύο υποπίνακες. Σε κάθε επανάληψη, αναγνωρίζουμε τον αριθμό από την αριστερή πλευρά ο οποίος είναι μεγαλύτερος από το πύβος, και επίσης αναγνωρίζουμε τον αριθμό από την δεξιά πλευρά που είναι μικρότερο του πύβος. Όταν τελειώσει η αναζήτηση, τότε ανταλλάζουμε τις δύο τιμές. Καλούμε αναδρομικά την μέθοδο και στους δύο πίνακες.

ΜΕΡΟΣ Β ΚΑΙ Γ

Η μέθοδος `remove(int id)` της PQ κάνει πρώτα έναν έλεγχο, αν είναι άδεια η ουρά. Εάν περιέχει στοιχεία, τότε η μέθοδος ψάχνει στο σωρό σειριακά για το τραγούδι με το `id` που της δόθηκε. Αν αποτύχει η αναζήτηση, επιστρέφει ένα μήνυμα στο χρήστη. Αν το τραγούδι βρεθεί, αλλάζει θέσεις το $1^ο$ (με την μεγαλύτερη προτεραιότητα) με το τελευταίο τραγούδι και έπειτα το τραγούδι προς αφαίρεση με το τραγούδι στην τελευταία θέση (που ήταν το $1^ο$). Μετά το τραγούδι στην πρώτη θέση του πίνακα κάνει `sink`, ενώ το τραγούδι με την μεγαλύτερη προτεραιότητα θα κάνει `swim` (αφού είναι μέγιστο θα κάνει πάντα `swim`). Η πολυπλοκότητα είναι $O(N)$.

Το πρόγραμμα `top_k_withPQ` δέχεται δύο ορίσματα από τον χρήστη. Το πρώτο όρισμα είναι ένας ακέραιος `k` που αναπαριστά το πλήθος των καλύτερων τραγουδιών που ζητά ο χρήστης και το δεύτερο όρισμα είναι η λίστα με τα τραγούδια.

Αρχικά γίνεται έλεγχος, εάν τα ορίσματα δεν είναι δύο τότε εμφανίζεται μήνυμα λάθους.

Δημιουργούνται ένας Scanner για το διάβασμα του αρχείου και μία ουρά προτεραιότητας χωρητικότητας k.

Το πρόγραμμα μπαίνει στην συνέχεια σε ένα while όπου αρχικά διαβάζει τη πρώτη «λέξη» η οποία είναι το id του τραγουδιού. Ελέγχει εάν οι επόμενες «λέξεις» δεν αποτελούν ακέραιο αριθμό οπότε είναι μέρος του τίτλου του τραγουδιού και τέλος διαβάζει την τελευταία «λέξη» της γραμμής η οποία αναφέρεται στα likes που έχει το τραγούδι. Τότε γίνονται οι απαραίτητοι έλεγχοι για το id και τον τίτλο του τραγουδιού και εφόσον είναι έγκυρο το τραγούδι, προστίθενται στην ουρά προτεραιότητας. Η διαδικασία αυτή επαναλαμβάνεται μέχρι το τέλος του αρχείου.

Τέλος, εκτυπώνονται τα k καλύτερα τραγούδια καλώντας κάθε φορά την getMax().

ΜΕΡΟΣ Δ

Η κλάση Dynamic_Median χρησιμοποιεί δύο ουρές προτεραιότητας με διαφορετικούς comparators , έτσι ώστε η μία να είναι μεγιστοστρεφής και η άλλη ελαχιστοστρεφής.

Η πρώτη κρατάει τα στοιχεία που είναι μικρότερα του median και η δεύτερη αυτά που είναι μεγαλύτερα. Με την εισαγωγή νέου στοιχείου ενημερώνονται οι δυο ουρές και γίνεται προσαρμογή με τέτοιο τρόπο, έτσι ώστε η διαφορά του πλήθους των στοιχείων κάθε ουράς να μη ξεπερνά το 2.

Η πολυπλοκότητα από την ανάγνωση μίας γραμμής για την εισαγωγή είναι $O(\log N)$, με N το πλήθος των στοιχείων σε μία από της ουρές(η άλλη θα έχει N-1, N ή N+1). Καλεί την insert του Dynamic_Median που καλεί τις insert των ουρών $O(\log N)$ και την rebalance() που και εκείνη με τη σειρά της καλεί getMax() και insert $O(\log n)$ ($2\log N$). Εάν η γραμμή είναι πολλαπλάσιο του 5, καλείται η median() που καλεί την max() $O(1)$.