

- Σε αυτή την εργασία τροποποιήθηκε η κλάση Node από την προηγούμενη εργασία ώστε να μπορεί να χρησιμοποιηθεί για την δημιουργία λίστας διπλής σύνδεσης.

Ά μέρος

Για την υλοποίηση του 1^{ου} μέρους χρησιμοποιήθηκε η λίστα διπλής σύνδεσης ώστε να μπορεί να τροποποιηθεί δυναμικά κατά την ανάγνωση του αρχείου. Η ταξινόμηση της λίστας έγινε τροποποιώντας κατάλληλα τον αλγόριθμο της quicksort.

Αρχικά ,στη μέθοδο που υλοποιεί τη διαμέριση `partition(DoubleLinkedList , Node<T> left ,Node<T> right)` , επιλέγοντας το τελευταίο στοιχείο της λίστας που εξετάζεται κάθε φορά ως στοιχείο διαμέρισης ελέγχουμε με τη σειρά κάθε στοιχείο από τα αριστερά μέχρι να φτάσουμε στο τέλος της λίστας. Για τους σκοπούς της διαμέρισης χρησιμοποιείται ο δείκτης `i` ,δεξιά του οποίου τοποθετούνται όλα τα στοιχεία μικρότερα του στοιχείου διαμέρισης και αριστερά του ,όλα τα στοιχεία που είναι μεγαλύτερα από αυτό .

Τέλος ,ο δείκτης `i` παίρνει την τιμή του στοιχείου διαμέρισης και επιστρέφεται.

Η διαδικασία αυτή συνεχίζεται για κάθε υπολίστα που καθορίζεται από το στοιχείο διαμέρισης.

Η πολυπλοκότητα του αλγορίθμου είναι ίδια με της quicksort για πίνακες ($O(n \log n)$ μέση περίπτωση ή $O(N^2)$ όταν το στοιχείο διαμέρισης είναι το μικρότερο στοιχείο της λίστας).

Β' Μέρος

Η remove στη κλάση PQ τρέχει σε $O(N)$, αφού δεν χρησιμοποιείται κάποια βοηθητική δομή που επιτρέπει την άμεση εύρεση του συγκεκριμένου στοιχείου, συνεπώς πρέπει να διασχίσει όλο το σωρό μέχρι να βρει το στοιχείο με το id που δόθηκε.

Γ' Μέρος

Στο πρόγραμμα Top_k_WithPQ αρχικά εισάγονται στην ουρά τα πρώτα k τραγούδια που διαβάζονται, δηλαδή μέχρι να φτάσουμε στον περιορισμό μεγέθους. Στη συνέχεια κάθε επιπλέον τραγούδι που διαβάζεται, συγκρίνεται με το τραγούδι που έχει τα λιγότερα likes στην λίστα. Για να γίνει αυτό έχει προστεθεί στην κλάση PQ ένα πεδίο minElement καθώς και δύο μέθοδοι οι :

- FindMin() που βρίσκει εκ νέου το ελάχιστο στοιχείο μόνο στην περίπτωση που αυτό αφαιρεθεί μέσω της remove.
- getMin() που επιστρέφει το ελάχιστο στοιχείο χωρίς να το αφαιρεί.
- Σημειώνεται πως το πεδίο minElement ενημερώνεται αυτόματα κάθε φορά που εισάγεται ένα νέο στοιχείο στη λίστα.

Αν το τραγούδι που ελέγχεται έχει περισσότερα likes από το minElement, το ελάχιστο στοιχείο πρέπει να διαγραφεί και να μπει το τραγούδι που ελέγξαμε, ώστε να μην παραβιαστεί ο περιορισμός μεγέθους. Διαφορετικά δεν γίνεται τίποτα και συνεχίζεται κανονικά το διάβασμα του αρχείου.

Λόγω της remove, στην χειρότερη περίπτωση κάθε τραγούδι που διαβάζεται θα είναι μεγαλύτερο από το χειρότερο τραγούδι της λίστας, που σημαίνει ότι θα πρέπει να καλέσουμε τη remove (όπως και την insert) τόσες φορές όσα και τα τραγούδια στο αρχείο πλην k

που είναι τα τραγούδια που εισάγουμε στην αρχή, δηλαδή $N-k$ φορές. Προφανώς η πολυπλοκότητα αυξάνεται όσο μικραίνει το k .

Αυτό σε συνδυασμό με την πολυπλοκότητα της `insert` και της `remove` δείχνει πως είναι προτιμότερο να χρησιμοποιηθεί το πρόγραμμα `Top_k` και η `quicksort` σε τέτοιες περιπτώσεις, από το να γίνουν $N-k$ διαδοχικές κλήσεις στις μεθόδους εισαγωγής και αφαίρεσης της `PQ`, με πολυπλοκότητα $O(\log n)$ και $O(N)$ αντίστοιχα.