Πανεπιστήμιο Ιωαννίνων

ΔΙΑΧΕΙΡΙΣΗ ΣΥΝΘΕΤΩΝ ΔΕΔΟΜΕΝΩΝ

ΕΡΓΑΣΙΑ 3η – Top-k queries

ΝΤΟΝΤΗΣ ΒΑΣΙΛΕΙΟΣ

ΑΜ: 3300

Διάβασμα rnd.txt αρχείου και αρχικοποίηση πινάκων

Αρχικά, ξεκινάμε διαβάζοντας το αρχείο rnd.txt και καθώς το διαβάζουμε παίρνουμε τα απαραίτητα δεδομένα για να γεμίσουμε τον πίνακα με όλα τα σκορ των αντικειμένων. Αρχικοποιούμε με None σε κάθε θέση τον πίνακα R και αρχικοποιούμε με None σε όλες τις θέσεις και τους πίνακες RloweBoundScore και RtotalScore, τους πίνακες που κρατάνε τα σκορ με τα κάτω όρια των αντικειμένων και το πλήρες σκορ αντίστοιχα. Για κάθε γραμμή που διαβάζουμε αφού διαβάσουμε το ID του αντικειμένου, τοποθετούμε στην θέση ID του πίνακα R το αντίστοιχο rating που διαβάσαμε.

Διάβασμα αρχείων με Round Robin και διαχείριση των δεδομένων

Εκτελούμε μια open with και για τα δύο αρχεία και αρχικοποιούμε την παρακάτω μεταβλητές:

- sequentialAccessesCounter : μετρητής συνολικών σειριακών προσπελάσεων
- heap : πίνακας που περιέχει τα top-k αντικείμενα και τα ID τους.
- seq1LastRating : μεταβλητή που κρατάει το τελευταίο αντικείμενο που διαβάστηκε από το αρχείο seq1
- seq2LastRating : μεταβλητή που κρατάει το τελευταίο αντικείμενο που διαβάστηκε από το αρχείο seq2
- flag : σημαία που γίνεται αληθής, όταν φτάσουμε τα k διαβάσματα αντικειμένων

Στην συνέχεια διαβάζουμε ένα-ένα τις γραμμές των αρχείων εναλλάξ. Για κάθε γραμμή ξεκινάμε ελέγχοντας αν το αντικείμενο είναι το κ-οστό αντικείμενο για να αλλάξουμε την σημαία σε αληθή και να ταξινομήσουμε τον πίνακα μέσω της sort (θα αναλυθεί παρακάτω). Αφού διαβάζουμε το ID και το rating του αντικειμένου, ελέγχουμε αν το κάτω όριο του είναι None, ελέγχοντας έτσι αν το έχουμε ξαναδεί. Αν δεν το έχουμε ξαναδεί, ενημερώνουμε τον πίνακα με τα κάτω όρια και αν δεν έχουμε διαβάσει κ αντικείμενα το τοποθετούμε χωρίς έλεγχο στον πίνακα με τα top-κ αντικείμενα. Αν έχουμε διαβάσει κ ή περισσότερα αντικείμενα, ενημερώνουμε το threshold σε 5 + rating + τελευταίο_rating_που_διαβάστηκε_από_το_άλλο_αρχείο και ελέγχουμε αν το threshold είναι μεγαλύτερο από το μικρότερο αντικείμενο του heap. Αν ισχύει, τότε ελέγχουμε αν το κάτω όριο του αντικειμένου που διαβάζουμε είναι μεγαλύτερο από το μικρότερο rating του μικρότερου κ αντικειμένου και αν ισχύει το αντικαθιστούμε με το αντικείμενο που διαβάζουμε, αντικαθιστούμε επίσης και τα IDs και sortάρουμε τον πίνακα heap. Αν το threshold δεν είναι μεγαλύτερο από το min-k αντικείμενο, δημιουργούμε μια σημαία αρχικοποιημένη ψευδής και τον πίνακα heapArray που κρατάει τα IDs των top-k αντικειμένων. Διατρέχουμε με μια for τον πίνακα κάτω ορίων και ελέγχουμε για τα αντικείμενα που έχουμε διαβάσει και δεν είναι στα top-k αντικείμενα, ελέγχουμε αν το κάτω όριο αυτού του αντικειμένου αθροισμένο με το rating που διαβάσαμε είναι μεγαλύτερο από το κάτω όριο του μικρότερου αντικειμένου του πίνακα heap αθροισμένο και αυτό με το rating. Αν είναι αληθής η συνθήκη αλλάζουμε την σημαία σε αληθή σε σταματάμε με break την for. Ελέγχουμε αν πρέπει να μπει στο heap και αν μπει ενημερώνουμε τον πίνακα και τον ταξινομούμε. Αν η σημαία είναι ψευδής, αυτό σημαίνει ότι τελειώσαμε και έτσι ενημερώνουμε τον μετρητή σειριακών προσπελάσεων και αφού τυπώσουμε τα αποτελέσματα μέσω της printer(θα αναλυθεί παρακάτω), σταματάμε το πρόγραμμα.

Αν έχουμε ξαναδεί το αντικείμενο, ενημερώνουμε τον πίνακα με τα πλήρη σκορ και αν δεν έχουμε διαβάσει κ αντικείμενα το τοποθετούμε χωρίς έλεγχο στον πίνακα με τα top-κ αντικείμενα. Αν έχουμε διαβάσει κ ή περισσότερα αντικείμενα, ενημερώνουμε το threshold σε 5 + rating + τελευταίο_rating_που_διαβάστηκε_από_το_άλλο_αρχείο και ελέγχουμε αν το threshold είναι μεγαλύτερο από το μικρότερο αντικείμενο του heap. Αν ισχύει, τότε ελέγχουμε αν το κάτω όριο του αντικειμένου που διαβάζουμε είναι μεγαλύτερο από το μικρότερο rating του μικρότερου κ αντικειμένου και αν ισχύει το αντικαθιστούμε με το

αντικείμενο που διαβάζουμε, αντικαθιστούμε επίσης και τα IDs και sortάρουμε τον πίνακα heap. Αν το threshold δεν είναι μεγαλύτερο από το min-k αντικείμενο, δημιουργούμε μια σημαία αρχικοποιημένη ψευδής και τον πίνακα heapArray που κρατάει τα IDs των top-k αντικειμένων. Διατρέχουμε με μια for τον πίνακα κάτω ορίων και ελέγχουμε για τα αντικείμενα που έχουμε διαβάσει και δεν είναι στα top-k αντικείμενα, ελέγχουμε αν το κάτω όριο αυτού του αντικειμένου αθροισμένο με το rating που διαβάσαμε είναι μεγαλύτερο από το κάτω όριο του μικρότερου αντικειμένου του πίνακα heap αθροισμένο και αυτό με το rating. Αν είναι αληθής η συνθήκη αλλάζουμε την σημαία σε αληθή σε σταματάμε με break την for. Ελέγχουμε αν πρέπει να μπει στο heap και αν μπει ενημερώνουμε τον πίνακα και τον ταξινομούμε. Αν η σημαία είναι ψευδής, αυτό σημαίνει ότι τελειώσαμε και έτσι ενημερώνουμε τον μετρητή σειριακών προσπελάσεων και αφού τυπώσουμε τα αποτελέσματα μέσω της printer(θα αναλυθεί παρακάτω), σταματάμε το πρόγραμμα.

Πριν τελειώσουμε με το αντικείμενο αυτού του αρχείου, ενημερώνουμε το seq1LastRating με το αντικείμενο που διαβάσαμε, έτσι ώστε να το χρησιμοποιήσουμε για το άλλο αρχείο και ενημερώνουμε και τον μετρητή των σειριακών προσπελάσεων.

Για το άλλο αρχείο ακολουθούμε ακριβώς την ίδια τακτική, οπότε δεν θα αναλυθεί.

Συναρτήσεις που χρησιμοποιήθηκαν

1. ***Printer(array, accesses):*** Τυπώνει στον χρήστη τον αριθμό των σειριακών προσπελάσεων και αφού αντιστρέψει τον πίνακα με τα top-k αντικείμενα, τα τυπώνει ένα-ένα, με τα ID τους και τις βαθμολογίες τους.
2. ***Sort(array):*** Διατρέχοντας τα αντικείμενα του πίνακα, κρατάμε στην μεταβλητή minimumScore το σκορ του αντικειμένου που διατρέχουμε και στην μεταβλητή minimumScoreIndex την θέση του αντικειμένου. Στην συνέχεια διατρέχουμε τα υπόλοιπα αντικείμενα της λίστας και αν κάποιο είναι μικρότερο, ενημερώνουμε τις μεταβλητές minimumScore και minimumScoreIndex. Όταν τελειώσει το τρέξιμο των υπόλοιπων αντικειμένων, αλλάζουμε την θέση του αντικειμένου που ελέγχουμε στην for, με αυτό που προέκυψε ως το μικρότερο. Έτσι, διατρέχοντας όλα τα αντικείμενα, έχουμε μια ταξινομημένη λίστα σύμφωνα με τις βαθμολογίες, δεσμευμένα όμως μαζί με το ID τους . Τέλος επιστρέφουμε τον ταξινομημένο πίνακα. Η χρήση της γίνεται επειδή θέλουμε να αντιστοιχούμε τα ID με την βαθμολογία καθώς τα χρειαζόμαστε.

ΤΕΛΟΣ