Γεώργιος Δημόπουλος

Α.Μ. 2964

3η σειρά ασκήσεων στο μάθημα: Διαχείριση Σύνθετων Δεδομένων (κ.Μαμουλής)

ΜΕΡΟΣ 1ο:

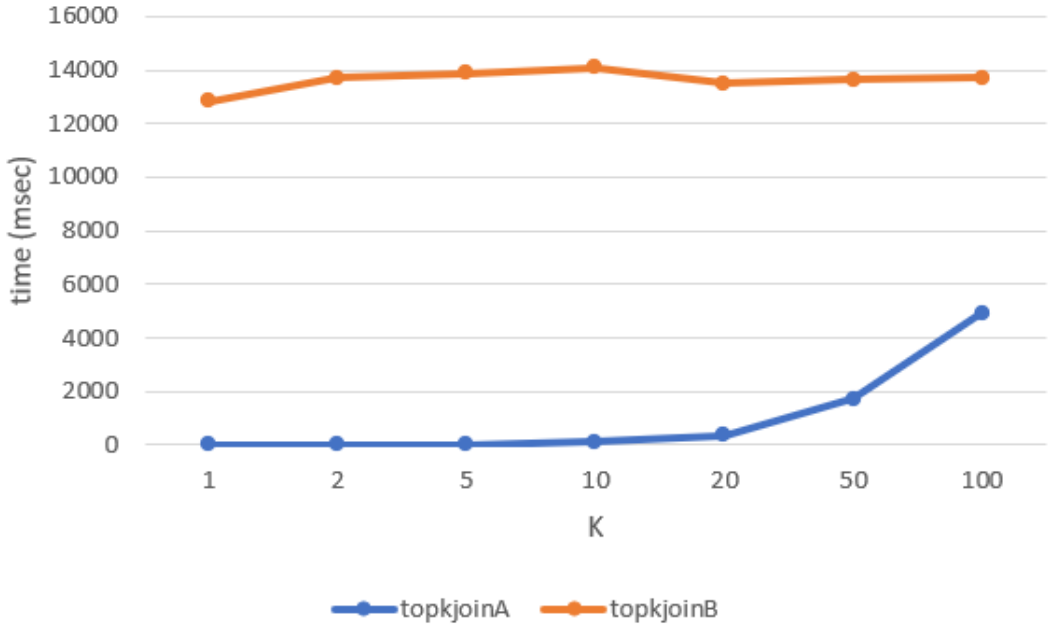
Σε αυτό το μέρος θα υλοποιήσουμε τον αλγόριθμο A top-k join, όπως είναι στις διαφάνειες του μαθήματος. Επειδή ξέρω ότι τα αρχεία είναι ταξινομημένα με βάση το πεδίο instance_weight, θα θέσω από την αρχή για p1_max, p2_max την τιμή που έχει η πλειάδα της πρώτης γραμμής των αρχείων. Τώρα ξεκινάω με το αρχείο male. Παίρνω τις έγκυρες πλειάδες και βρίσκω το κατώφλι. Έπειτα βάζω την πλειάδα σε ένα λεξικό με κλειδί το age. Αμέσως μετά ελέγχω αν το age βρίσκεται σαν κλειδί στο λεξικό του female. Αν βρίσκεται το βάζω μέσα στην max heap. Τώρα όσο η ουρά δεν είναι άδεια και η τιμή του πρώτου στοιχείου της ουράς είναι μεγαλύτερη από το κατώφλι, επιστρέφω το μεγαλύτερο στοιχείο της ουράς. Έπειτα παίρνω την πλειάδα από το female. Ενημερώνω το κατώφλι. Βάζω την πλειάδα σε ένα λεξικό με κλειδί το age. Μετά ελέγχω αν το age βρίσκεται σαν κλειδί στο λεξικό του male. Αν βρίσκεται το βάζω μέσα στην max heap. Όσο η ουρά δεν είναι άδεια και η τιμή του πρώτου στοιχείου της ουράς είναι μεγαλύτερο από το κατώφλι, επιστρέφω το μεγαλύτερο στοιχείο της ουράς. Να σημειώσω ότι για να κάνω την ουρά max heap απλά τοποθετώ την τιμή που με ενδιαφέρει με αρνητικό πρόσημο. Τέλος το πρόγραμμα μου τυπώνει το συνολικό χρόνο που κάνει για να τρέξει, καθώς και τις έγκυρες πλειάδες που διαβάζονται από το αρχείο male και female.

ΜΕΡΟΣ 2ο:

Για αυτό το μέρος διαβάζω όλο το αρχείο males και βάζω όλες τις έγκυρες πλειάδες σε ένα λεξικό με κλειδί το age. Όταν τελειώσω διαβάζω την κάθε γραμμή του female. Για κάθε έγκυρη πλειάδα ελέγχω αν το age βρίσκεται σαν κλειδί στο λεξικό male. Αν βρίσκεται το βάζω στην min heap. Αυτό το κάνω μέχρι να γεμίσει η min heap με K στοιχεία. Όταν γεμίσει για κάθε νέο ζευγάρι που θέλει να μπει στην min heap, το ελέγχω αν είναι μεγαλύτερο από την τιμή του πρώτου στοιχείου της min heap. Αν είναι μεγαλύτερο κάνω pop την ρίζα και push το νέο ζευγάρι. Με αυτόν τον τρόπο διαβάζοντας και όλο το αρχείο female θα έχω στην ουρά τα καλύτερα top K ζευγάρια. Μετά αυτό που κάνω είναι σαν μια μορφή ταξινόμησης από το μεγαλύτερο προς το μικρότερο με την συνάρτηση *nlargest*. Τέλος το πρόγραμμα μου τυπώνει και το συνολικό χρόνο τον οποίο χρειάζεται για να τρέξει.

ΜΕΡΟΣ 3ο:

Το διάγραμμα που προκύπτει από την εκτέλεση των δύο αλγορίθμων είναι το εξής:



Όπως βλέπουμε από το παραπάνω διάγραμμα ο Α αλγόριθμος είναι πιο γρήγορος από τον Β αλγόριθμο. Ο Β αλγόριθμος τρέχει σε σταθερό χρόνο ανεξάρτητα από το Κ. Προφανώς για μικρό αριθμό Κ μας συμφέρει ο αλγόριθμος Α. Όμως όσο αυξάνεται ο αριθμός Κ, ο χρόνος του αλγορίθμου Α τείνει να φτάσει τον χρόνο του αλγορίθμου Β και μετά από κάποιο όριο και να τον ξεπεράσει. Οπότε ο αλγόριθμος Α μας συμφέρει για ένα συγκεκριμένο Κ, ενώ ο αλγόριθμος Β τρέχει σε σταθερό χρόνο για οποιοδήποτε Κ.

Παρακάτω παρουσιάζεται ο αριθμός των έγκυρων γραμμών για τον αλγόριθμο Α:

K	**Έγκυρες γραμμές για το αρχείο male**	**Έγκυρες γραμμές για το αρχείο female**
1	44	44
2	134	134
5	437	437
10	1145	1145
20	2656	2656
50	6030	6030
100	10139	10139