ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΥΕ041 - ΠΛΕ081: Διαχείριση Σύνθετων Δεδομένων

(ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2023-24)

**ΕΡΓΑΣΙΑ 3 – Ερωτήσεις κορυφαίων κ και κορυφογραμμής**

**Προθεσμία: 31 Μαΐου 2024, 9μ.μ.**

Στο ecourse θα βρείτε τα αρχεία δεδομένων που θα χρησιμοποιήσετε γι' αυτή την εργασία. Τα δεδομένα προέρχονται από το https://kdd.ics.uci.edu/databases/census-income/census-income.html και αποτελούν δημογραφικά δεδομένα από τις ΗΠΑ. Τα δεδομένα έχουν χωριστεί σε δύο αρχεία τα οποία **είναι ταξινομημένα** με βάση το πεδίο **instance weight** σε **φθίνουσα σειρά**: το αρχείο **males_sorted** έχει όλες τις εγγραφές που αντιστοιχούν σε άντρες (πεδίο sex = Male) και το αρχείο **females_sorted** έχει όλες τις εγγραφές που αντιστοιχούν σε γυναίνες (πεδίο sex = Female). Σε κάθε εγγραφή έχει προστεθεί μία κολώνα στην αρχή, η οποία περιέχει το id της εγγραφής (δηλαδή τη σειρά της) στο original αρχείο πριν το διαχωρισμό και την ταξινόμηση. Για παράδειγμα, το 135085 στην πρώτη γραμμή του males_sorted σημαίνει ότι η γραμμή αυτή βρίσκεται στη θέση 135085 στο original αρχείο census-income.data, ενώ οι τιμές που ακολουθούν το 135085 είναι οι τιμές των πεδίων age, class of worker, κλπ. Κάντε unzip τα αρχεία και μετά χρησιμοποιήστε τα.

Στόχος είναι να βρούμε τα top-K ζευγάρια με το υψηλότερο άθροισμα των πεδίων instance weight, λαμβάνοντας υπόψη μόνο τα ζευγάρια που έχουν **την ίδια ηλικία**. Αποκλείονται άτομα που είναι παντρεμένα (η τιμή του πεδίου marital status ξεκινάει με "Married") και ανήλικα άτομα (η τιμή του πεδίου age είναι μικρότερη του 18). Οι γραμμές των αρχείων που αντιστοιχούν σε μη έγκυρα άτομα πρέπει να αγνοούνται.

Για παράδειγμα, το top-1 ζευγάρι είναι το (135085, 67141) με ηλικία 49 και άθροισμα instance weight 18656.3+7129.24= 25785.54

**Μέρος 1: Αλγόριθμος A top-k join**

Γράψτε ένα πρόγραμμα που να υλοποιεί τον αλγόριθμο top-k join που διδάχθηκε στο μάθημα (διαφάνειες 51-55, αρχείο topk.pptx). Ο αλγόριθμος περιγράφεται και ως HRJN στην εργασία: https://cs.uwaterloo.ca/~ilyas/papers/rank_join2.pdf (προσοχή: υλοποιήστε τον αλγόριθμο HRJN και όχι τον HRJN*)

Ο αλγόριθμος θα πρέπει να διαβάζει την επόμενη **έγκυρη** γραμμή από το males_sorted ή το female_sorted (εναλλάξ) και θα ενημερώνει τα αντίστοιχα hash tables (hash maps ή dictionaries) τα οποία θα κρατούν τις γραμμές που έχουμε δει μέχρι τώρα από το κάθε αρχείο, οργανωμένες με βάση το πεδίο age. (Στα hash tables κρατείστε από τις γραμμές μόνο τα πεδία που μας ενδιαφέρουν.) Θα ενημερώνει κάθε φορά το κατώφλι (threshold) T και θα βρίσκει τα αποτελέσματα του join της

τρέχουσας γραμμής με το hash table του άλλου αρχείου. Τα μέχρι στιγμής αποτελέσματα θα οργανώνονται σε ένα max heap Q και θα δίνονται στην έξοδο εκείνα που είναι πάνω από το κατώφλι T. Ο αλγόριθμος θα πρέπει να υλοποιηθεί σαν generator function, δηλαδή η αντίστοιχη συνάρτηση θα πρέπει να επιστρέφει σε κάθε κλήση το επόμενο join result. Π.χ. τα πρώτα 5 join results είναι:

```
1. pair: 135085,67141 score: 25785.54
2. pair: 135085,44307 score: 24247.12
3. pair: 135085,111291 score: 23657.66
4. pair: 135085,12112 score: 23644.20
5. pair: 135085,183898 score: 23046.54
```

Δηλαδή, το ζευγάρι (`135085,67141`) είναι το κορυφαίο με συνολικό σκορ `25785.54`, το δεύτερο κορυφαίο είναι το (`135085,44307`) με συνολικό σκορ `24247.12`, κλπ.

Το πρόγραμμά σας θα πρέπει να παίρνει στη γραμμή διαταγών έναν ακέραιο K και να τυπώνει τα K πρώτα ζεύγη όπως παραπάνω.

Το πρόγραμμά σας θα πρέπει να τυπώνει και το χρόνο εκτέλεσης του.

**Μέρος 2: Αλγόριθμος B top-k join**

Γράψτε ένα πρόγραμμα που να υλοποιεί έναν εναλλακτικό αλγόριθμο top-k join. Ο αλγόριθμος διαβάζει εξολοκλήρου το αρχείο males_sorted και βάζει τις πλειάδες (μόνο τις έγκυρες) σε ένα hash table (με κλειδί το age). Μετά, διαβάζει μία προς μία τις πλειάδες από το females_sorted και για καθεμιά από αυτές βρίσκει τις πλειάδες με τις οποίες κάνει join χρησιμοποιώντας το hash table. Από τα αποτελέσματα του join που προκύπτουν κρατάμε σε ένα min-heap τα μέχρι στιγμής κορυφαία K. Μόλις ολοκληρωθεί ο αλγόριθμος το heap θα πρέπει να έχει τα κορυφαία K ζευγάρια.

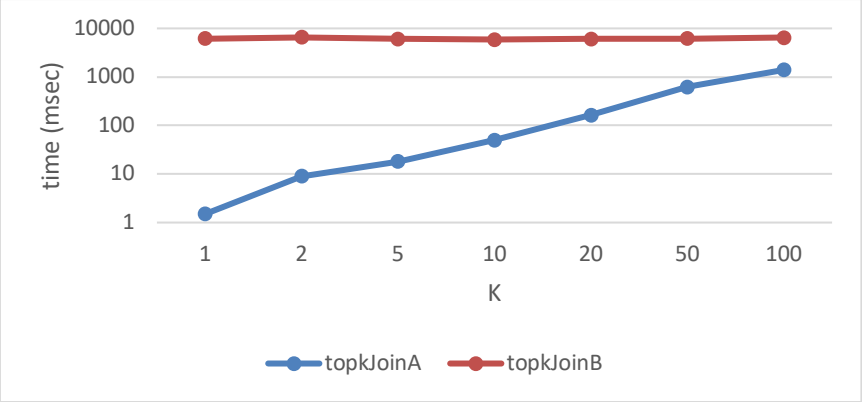
Το πρόγραμμά σας θα πρέπει να παίρνει στη γραμμή διαταγών έναν ακέραιο K και να τυπώνει τα K κορυφαία ζευγάρια. Τα αποτελέσματα πρέπει να είναι τα ίδια με αυτά του αλγορίθμου A (αλλιώς ένα από τα 2 προγράμματα είναι λανθασμένο).

Το πρόγραμμά σας θα πρέπει να τυπώνει και το χρόνο εκτέλεσης του.

**Μέρος 3: Γραπτό μέρος**

Εκτελέστε τα δύο προγράμματα για τις ακόλουθες τιμές του K: 1, 2, 5, 10, 20, 50, 100

Φτιάξτε ένα διάγραμμα, το οποίο δείχνει τον χρόνο εκτέλεσης του καθενός από τους αλγορίθμους για καθεμιά από τις τιμές του K. Για παράδειγμα:




Εξηγήστε τη συμπεριφορά των δύο αλγορίθμων. Επίσης, για κάθε τιμή του K, γράψτε τον αριθμό των **έγκυρων γραμμών** που έχει διαβάσει ο αλγόριθμος A από κάθε αρχείο.

**Έγκυρη γραμμή:** περιέχει πληροφορίες για ένα άτομο που δεν είναι παντρεμένο και είναι τουλάχιστον 18 ετών

Επίσης εξηγήστε τα πλεονεκτήματα και τα μειονεκτήματα του αλγορίθμου A έναντι του αλγορίθμου B.

**Παραδοτέα:** Κάντε turnin στο assignment3@mye041 τα προγράμματά σας και ένα PDF αρχείο το οποίο τεκμηριώνει τα προγράμματα, περιέχει οδηγίες εκτέλεσης/χρήσης και το γραπτό μέρος (3) της εργασίας.

**Οδηγίες για τις υποβολές:**

1) Μπορείτε να χρησιμοποιήσετε δομές όπως priority queue ή heap από τις βιβλιοθήκες της γλώσσας προγραμματισμού (π.χ. το module heapq της Python).
2) Αν χρησιμοποιήσετε Java, το πρόγραμμά σας θα πρέπει να γίνεται compile και να τρέχει και εκτός Eclipse στους υπολογιστές του εργαστηρίου. **Μην χρησιμοποιείτε packages**.
3) Αν χρησιμοποιήσετε Python, μην χρησιμοποιήστε τη βιβλιοθήκη pandas και μην υποβάλετε κώδικα για interactive programming (π.χ. ipython)
4) Υποβάλετε τις εργασίες σας σε ένα **zip** αρχείο (**όχι rar**) το οποίο πρέπει να περιλαμβάνει όλους τους κώδικες καθώς και ένα αρχείο τεκμηρίωσης το οποίο να περιγράφει τη μεθοδολογία σας και να περιλαμβάνει το PDF αρχείο. **Μην υποβάλετε αρχεία δεδομένων**.
5) Μην ξεχνάτε να βάζετε το όνομά σας και το ΑΜ σε κάθε αρχείο που υποβάλετε.