ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΥΕ041 - ΠΛΕ081: Διαχείριση Σύνθετων Δεδομένων

(ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2018-19)

ΕΡΓΑΣΙΑ 3 – Ερωτήσεις κορυφαίων κ και κορυφογραμμής

Προθεσμία: 27 Μαΐου 2019

Στο ecourse θα βρείτε τα αρχεία δεδομένων που θα χρησιμοποιήσετε γι αυτή την εργασία. Τα δεδομένα προέρχονται από την ιστοσελίδα στατιστικών του NBA (https://www.basketball-reference.com), κατέβηκαν από το https://www.kaggle.com/drgilermo/nba-players-stats και επεξεργάστηκαν. Τα αρχεία που θα χρησιμοποιήσετε είναι στατιστικά από τις επιδόσεις παικτών που αγωνίστηκαν στο NBA το 2017. Στο αρχείο 2017_ALL.csv μπορείτε να δείτε τα στοιχεία των παικτών που μας ενδιαφέρουν (όνομα και ομάδα) και κάποια στατιστικά: total rebounds (TRB), assists (AST), steals (STL), blocks (BLK), points (PTS). Για καθεμιά από τις 5 κατηγορίες των στατιστικών (TRB, AST, STL, BLK, PTS) υπάρχει και ένα αρχείο, το οποίο ταξινομεί τους παίκτες σε φθίνουσα σειρά με βάση την επίδοσή τους στο στατιστικό. Π.χ. στο 2017_TRB.csv, ο παίκτης με id 138 έχει πάρει τα περισσότερα rebounds (1116), ακολουθεί ο παίκτης 294 με 1114 rebounds, κλπ.

Μέρος 1: Top-k ερωτήματα

Γράψτε ένα πρόγραμμα που να υλοποιεί τον NRA, ώστε να βρίσκει τους κορυφαίους κ παίκτες με βάση τις επιδόσεις τους σε κάποια από τα 5 στατιστικά. Γι αυτό το σκοπό θα χρησιμοποιήσετε τα αρχεία 2017_TRB.csv, κλπ. με τους ταξινομημένους παίκτες. Συγκεκριμένα, στη **γραμμή διαταγών** ο χρήστης θα δίνει ένα πίνακα με τα στατιστικά που τον ενδιαφέρουν και τον ζητούμενο αριθμό κορυφαίων κ. Για παράδειγμα δίνοντας στη γραμμή διαταγών τις παραμέτρους [2,5] και 10, το πρόγραμμα πρέπει να υπολογίζει τους κορυφαίους 10 παίκτες της σαιζόν 2017 με βάση την επίδοσή τους σε assists και πόντους.

Η συνάρτηση συνάθροισης θα είναι το άθροισμα των **ομαλοποιημένων επιδόσεων** σε καθένα από τα δοθέντα στατιστικά. Π.χ. αν η παράμετρος είναι [2,5], τότε το σκορ του κάθε παίκτη στο στατιστικό 2 (assists) θα είναι ο αριθμός των assists του παίκτη διά το μέγιστο αριθμό assists (906: υπάρχει στην πρώτη γραμμή του 2017_AST.csv). Όμοια, το σκορ του κάθε παίκτη στο στατιστικό 5 (points) είναι οι πόντοι διαιρεμένοι με τον μέγιστο αριθμό πόντων (δηλ. 2558). Προσθέτοντας τα αυτά 2 σκορ παίρνουμε το συνολικό σκορ του παίκτη και στα δύο στατιστικά. Ζητάμε τους 10 παίκτες με τα μεγαλύτερα συνολικά σκορ.

Το πρόγραμμά σας θα πρέπει να ακολουθεί τη λογική του αλγορίθμου NRA. Δηλαδή θα πρέπει να διαβάζει ταυτόχρονα τα σχετικά αρχεία (π.χ. στο παράδειγμά μας τα αρχεία 2017_AST.csv και 2017_PTS.csv) και να κρατάει για κάθε παίκτη που βρίσκει σε ένα hash

map το κάτω όριο του συνολικού του σκορ. Μετά από κάθε γύρο από προσπελάσεις στα αρχεία θα πρέπει να ενημερώνει τους κ καλύτερους με βάση το κάτω όριο και να ελέγχει αν το σύνολο αυτό μπορεί να οριστικοποιηθεί (αν δεν υπάρχει κάποιος άλλος παίκτης που να μπορεί να τους ξεπεράσει στην καλύτερη περίπτωση). Κάνετε χρήση του ορίου T ώστε να μειώσετε τις συγκρίσεις στο growing phase του αλγορίθμου. Το πρόγραμμα θα πρέπει να τυπώνει στην έξοδο τους κ κορυφαίους μαζί με τα συνολικά τους σκορ, καθώς επίσης και τον αριθμό των γραμμών που έχουν διαβαστεί από τα αρχεία (number of accesses).

Για τον έλεγχο ορθότητας υλοποιήστε έναν απλό αλγόριθμο top-k ο οποίος θα διαβάζει το αρχείο 2017_ALL.csv, θα υπολογίζει το συνολικό σκορ του κάθε παίκτη και στο τέλος θα τυπώνει τους 10 παίκτες με το μεγαλύτερο συνολικό σκορ. Αν τα προγράμματά σας είναι σωστά θα πρέπει να τυπώνουν τα ίδια αποτελέσματα σε κάθε περίπτωση.

Μέρος 2: Ερωτήσεις κορυφογραμμής

Ζητείται να υλοποιήσετε έναν απλό αλγόριθμο που θα ακολουθεί τη λογική του BNL (block nested loops) αλγορίθμου στις σημειώσεις. Γράψτε ένα πρόγραμμα το οποίο θα δέχεται από τη γραμμή διαταγών σαν παράμετρο έναν πίνακα με τα στατιστικά που μας ενδιαφέρουν (π.χ. [2,5] υπονοεί ότι μας ενδιαφέρουν assists και points) και υπολογίζει και τυπώνει το skyline των παικτών που δεν κυριαρχούνται (not dominated) από κανέναν άλλο παίκτη με βάση τα στατιστικά αυτά (π.χ. με βάση assists και points αν έχουμε δώσει [2,5]).

Ο αλγόριθμος θα διαβάζει έναν-έναν τους παίκτες από το αρχείο 2017_ALL.csv και θα κρατάει σε μια δομή τους παίκτες που είναι στο skyline μέχρι στιγμής. Για κάθε νέο παίκτη που διαβάζει θα ελέγχει αν κυριαρχείται ο παίκτης από κάποιον που είναι στο skyline μέχρι τώρα (σε αυτή την περίπτωση ο παίκτης δεν μπαίνει στο skyline). Αν ο παίκτης δεν κυριαρχείται, τότε μπαίνει στο skyline και σβήνουμε από αυτό εκείνους τους παίκτες που κυριαρχούνται από το νέο παίκτη. Θεωρείστε ότι υπάρχει αρκετή μνήμη για να κρατήσουμε το skyline (άρα δεν χρειάζονται πολλά περάσματα στο αρχείο ή προσωρινά αρχεία).

Οδηγίες για τις υποβολές:

1) Μπορείτε να χρησιμοποιήστε έτοιμες δομές array, vector ή λιστών που προσφέρονται από τη γλώσσα προγραμματισμού που θα χρησιμοποιήσετε (Python, C, C++ ή Java).
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) 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 ότι μπορεί να βοηθήσει στη βαθμολόγηση.
4) Μην ξεχνάτε να βάζετε το όνομά σας (σε greeklish) και το ΑΜ σε κάθε αρχείο που υποβάλετε.
5) Κάντε turnin την εργασία στο assignment3@mye041