ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΥΕ041 - ΠΛΕ081: Διαχείριση Σύνθετων Δεδομένων

ΕΡΓΑΣΙΑ 1 – Υλοποίηση Τελεστών (προθεσμία: 4 Απριλίου 2025, 9μ.μ.)

Στόχος της εργασίας είναι η ανάπτυξη προγραμμάτων για την αποτίμηση (evaluation) τελεστών συνένωσης, ένωσης, τομής, διαφοράς και συνάθροισης.
Στην εργασία θα χρησιμοποιήσουμε συνθετικά δεδομένα, τα οποία σας δίνονται στο ecourse. Τα δεδομένα βρίσκονται στα αρχεία R.tsv, R_sorted.tsv, S_sorted.tsv. Τα αρχεία αυτά μπορείτε να τα εκλάβετε σαν σχεσιακούς πίνακες με δύο πεδία A (αλφαριθμητικό 2 χαρακτήρων) και B (ακέραιος). Κάθε γραμμή σε κάθε αρχείο αντιστοιχεί σε μία πλειάδα, όπου οι τιμές των A και B χωρίζονται με tab. Το αρχείο R_sorted.tsv περιέχει τις ίδιες πλειάδες με το R.tsv, αλλά είναι ταξινομημένο. Το αρχείο S_sorted.tsv περιέχει τις πλειάδες μιας σχέσης S ταξινομημένες. Προσοχή: οι πλειάδες δεν είναι μοναδικές σε κάθε αρχείο, πράγμα που σημαίνει ότι μία πλειάδα μπορεί να υπάρχει πάνω από μία φορές (bag semantics). Ανοίξτε τα αρχεία και βεβαιωθείτε ότι κατανοείτε τα περιεχόμενά τους.

Μέρος 1 (merge-join)

Γράψτε ένα πρόγραμμα, το οποίο διαβάζει τα αρχεία R_sorted.tsv και S_sorted.tsv και υπολογίζει και γράφει σε ένα αρχείο RjoinS.tsv τη συνένωση (join) των R και S, θεωρώντας ότι έχουν κοινό το πρώτο πεδίο τους μόνο. Για παράδειγμα, η συνένωση της πλειάδας (`'aa'`, 33) από το R_sorted.tsv με την πλειάδα (`'aa'`, 45) από το S_sorted.tsv, πρέπει να παράγει στην έξοδο την πλειάδα (`'aa'`, 33, 45). Οι πλειάδες της εξόδου γράφονται στο RjoinS.tsv χωρισμένες με tabs, για παράδειγμα:

```
ab  33  45
ab  33  48
ab  90  45
...
```

Προσοχή: Το πρόγραμμά σας θα πρέπει να υλοποιεί πιστά τον αλγόριθμο merge-join. Αυτό σημαίνει ότι
1) Οι γραμμές των αρχείων R_sorted.tsv και S_sorted.tsv θα διαβάζονται μόνο μία φορά
2) Δεν θα διαβάζετε τα πάντα σε πίνακες/λίστες στη μνήμη για να υλοποιήσετε το join χρησιμοποιώντας τις λίστες μετά. Για κάθε γραμμή που διαβάζετε από το κάθε αρχείο θα πρέπει να φροντίζετε να βρίσκετε τα αποτελέσματα που ταιριάζουν με αυτή τη γραμμή.
3) Το μόνο που επιτρέπεται είναι να μαζεύετε σε έναν πίνακα (buffer) τις γραμμές από το S που ταιριάζουν με την τρέχουσα γραμμή του R. Με αυτό τον τρόπο, αν η επόμενη γραμμή του R έχει την ίδια τιμή στο πεδίο του join με την προηγούμενη, δεν χρειάζεται να ξαναδιαβάσουμε τις γραμμές από το S που ταιριάζουν με αυτήν. Στο τέλος του προγράμματος τυπώστε το μέγιστο μέγεθος αυτού του buffer (σε γραμμές).

Προγράμματα που παραβαίνουν τις παραπάνω οδηγίες θα χάνουν βαθμούς.

Μέρος 2 (union)

Γράψτε ένα πρόγραμμα, το οποίο διαβάζει τα αρχεία R_sorted.tsv και S_sorted.tsv και υπολογίζει και γράφει σε ένα αρχείο RunionS.tsv την ένωση (union) των R και S, θεωρώντας ότι έχουν ακριβώς τα ίδια πεδία. Το πρόγραμμα θα πρέπει να διαβάζει μόνο μία φορά τις γραμμές των αρχείων R_sorted.tsv και S_sorted.tsv και ταυτόχρονα να υπολογίζει την ένωσή τους και να γράφει τις πλειάδες της ένωσης στο αρχείο εξόδου, υλοποιώντας παραλλαγή του merge-join αλγορίθμου. Σε αυτή την περίπτωση δεν επιτρέπεται η χρήση buffer. Επειδή τα R_sorted.tsv και S_sorted.tsv μπορεί να έχουν διπλότυπα, το πρόγραμμα θα πρέπει να φροντίζει να εξαλείφει τα διπλότυπα από τις εισόδους καθώς και να αποφεύγει να γράφει διπλότυπα στην έξοδο.
Παράδειγμα εξόδου:

```
aa  11
ab  33
ab  45
...
```

Μέρος 3 (intersection)

Γράψτε ένα πρόγραμμα, το οποίο διαβάζει τα αρχεία R_sorted.tsv και S_sorted.tsv και υπολογίζει και γράφει σε ένα αρχείο RintersectionS.tsv την τομή (intersection) των R και S, θεωρώντας ότι έχουν ακριβώς τα ίδια πεδία. Το πρόγραμμα θα πρέπει να διαβάζει μόνο μία φορά τις γραμμές των αρχείων R_sorted.tsv και S_sorted.tsv και ταυτόχρονα να υπολογίζει την τομή τους και να γράφει τις πλειάδες της τομής στο αρχείο εξόδου, υλοποιώντας παραλλαγή του merge-join αλγορίθμου. Σε αυτή την περίπτωση δεν επιτρέπεται η χρήση buffer. Επειδή τα R_sorted.tsv και S_sorted.tsv μπορεί να έχουν διπλότυπα, το πρόγραμμα θα πρέπει να φροντίζει να εξαλείφει τα διπλότυπα από τις εισόδους καθώς και να αποφεύγει να γράφει διπλότυπα στην έξοδο.
Παράδειγμα εξόδου:

```
bb  94
bh  10
cl  41
...
```

Μέρος 4 (set-difference)

Γράψτε ένα πρόγραμμα, το οποίο διαβάζει τα αρχεία R_sorted.tsv και S_sorted.tsv και υπολογίζει και γράφει σε ένα αρχείο RdifferenceS.tsv τη διαφορά (difference) των R και S, θεωρώντας ότι έχουν ακριβώς τα ίδια πεδία. Το πρόγραμμα θα πρέπει να διαβάζει μόνο μία φορά τις γραμμές των αρχείων R_sorted.tsv και S_sorted.tsv και ταυτόχρονα να υπολογίζει τη διαφορά τους και να γράφει τις πλειάδες της διαφοράς στο αρχείο εξόδου, υλοποιώντας παραλλαγή του merge-join αλγορίθμου. Σε αυτή την περίπτωση δεν επιτρέπεται η χρήση buffer. Επειδή τα R_sorted.tsv και S_sorted.tsv μπορεί να έχουν διπλότυπα, το πρόγραμμα θα πρέπει να φροντίζει να εξαλείφει τα διπλότυπα από τις εισόδους καθώς και να αποφεύγει να γράφει διπλότυπα στην έξοδο.
Παράδειγμα εξόδου:

```
aa  11
ab  33
ab  90
...
```

Μέρος 5 (Ομαδοποίηση και συνάθροιση)

Γράψτε ένα πρόγραμμα, το οποίο διαβάζει το αταξινόμητο αρχείο R.tsv και υπολογίζει και γράφει σε ένα αρχείο Rgroupby.tsv το αποτέλεσμα της ομαδοποίησης των πλειάδων της R με βάση το πρώτο πεδίο της και της συνάθροισης χρησιμοποιώντας τη συνάρτηση sum με βάση το δεύτερο πεδίο της. Για παράδειγμα, οι πλειάδες (`'ab'`, 33), (`'ab'`, 90) ομαδοποιούνται και γράφεται στην έξοδο η πλειάδα (`'ab'`, 123).

Αυτή τη φορά η υλοποίηση θα γίνει στην κύρια μνήμη. Αρχικά θα διαβάσετε όλο το αρχείο σε έναν πίνακα (λίστα). Μετά θα υλοποιήσετε και θα τρέξετε τον κλασσικό αλγόριθμο sort-merge στη μνήμη, ο οποίος όμως θα είναι αλλαγμένος ως εξής: αν δύο πλειάδες που συγχωνεύονται είναι ίδιες ως προς το πρώτο πεδίο, τότε κατά τη συγχώνευση δημιουργείται μία μόνο πλειάδα από αυτές, η οποία έχει σαν δεύτερο πεδίο το άθροισμα των δεύτερων πεδίων των συγχωνευμένων πλειάδων.
Παράδειγμα εξόδου:

```
aa   11
ab   123
ac   54
ad   46
...
```

Παραδοτέα: Κάντε turnin στο assignment1@mye041 τα προγράμματά σας και ένα PDF αρχείο το οποίο τεκμηριώνει τα προγράμματα και περιέχει διευκρινίσεις ως προς τη λειτουργία τους.

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) εάν αυτό απαιτείται.

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 (σε greeklish) και το ΑΜ σε κάθε αρχείο που υποβάλετε.

6) Ο έλεγχος των προγραμμάτων σας μπορεί να γίνει σε άλλα αρχεία εισόδου από αυτά που σας δίνονται, άρα θα πρέπει ο κώδικάς σας να μην εξαρτάται από τα συγκεκριμένα αρχεία εισόδου που σας δίνονται.