Διαχείριση Σύνθετων Δεδομένων

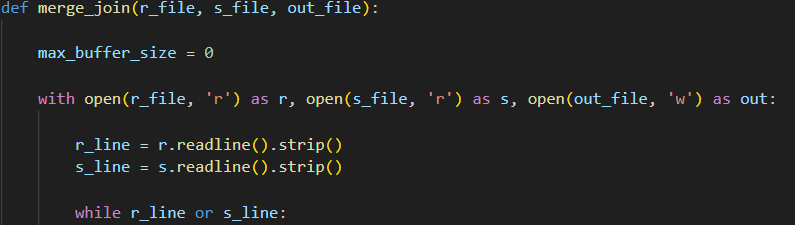
Αναφορά 1ης Εργαστηριακής Άσκησης

ΙΩΑΝΝΗΣ ΜΠΟΥΖΑΣ

ΑΜ:5025

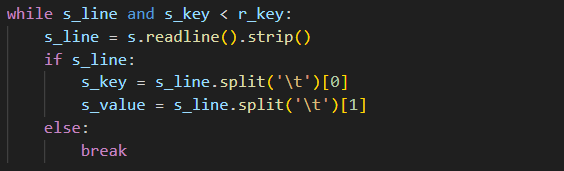
# Merge-Join

Σκοπός του προγράμματος είναι η συνένωση των αρχείων R\_sorted.tsv και S\_sorted.tsv θεωρώντας ότι έχουν κοινό το πρώτο πεδίο τους. Δηλαδή για (“aa”, 33) στο αρχείο R\_sorted και για (“aa”, 45) στο αρχείο S\_sorted θα τυπώσουμε στην έξοδο (“aa”, 33, 45).

Για να το πετύχουμε αυτό υλοποιούμε πιστά τον αλγόριθμο merge-join.Αρχικά ανοίγουμε τα αρχεία R\_sorted και S\_sorted για διάβασμα και το αρχείο out για γράψιμο της εξόδου και διαβάζουμε την πρώτη γραμμή των αρχείων.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

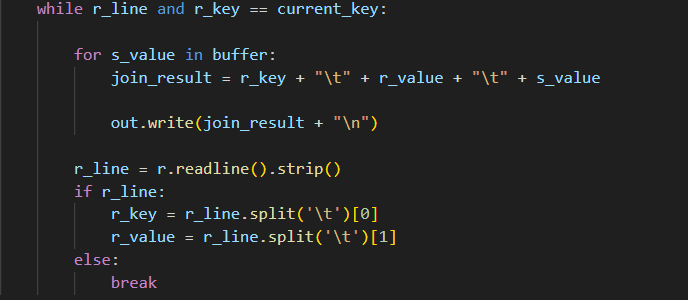
Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.Έπειτα για όσο υπάρχουν γραμμές που στα δυο αρχεία μας αρχικοποιουμε κάθε φορά έναν καινούργιο buffer όπου θα κρατάει τις γραμμές του S που έχουν ίδιο πρώτο πεδίο με μια γραμμή του R και τα r\_key, r\_value καθώς και s\_key, s\_value με τα οποία θα γίνουν οι διάφοροι έλεγχοι.

Με ένα while loop όσο το key των γραμμών του αρχείου του S είναι μικρότερο από αυτών των R προσπαθούμε να αποφύγουμε τις περιττές γραμμές του S που δεν έχουν ίδιο key με μια γραμμή του R.

Όταν τα keys και στο αρχείο S και στο αρχείο R είναι ίδια βάζουμε στον buffer μας ΟΛΕΣ τις γραμμές του S που ταιριάζουν με αυτό το key και υπολογίζουμε το max length του buffer μας. Αλλιώς απλα προχωράμε στην επόμενη γραμμή του αρχείου R.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

Τέλος για όσες γραμμές στο αρχείο R έχουν ίδιο key και για όλα τα στοιχεία του buffer μας τυπώνουμε στο output αρχείο μας το αποτέλεσμα της συνένωσης.

Η main του προγράμματος μας:

Εικόνα που περιέχει κείμενο, γραμματοσειρά, στιγμιότυπο οθόνης, γραμμή

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

# Union

Σε αυτό το πρόγραμμα θέλει να φτιάξουμε την ένωση των δυο αρχείων. Η ένωση είναι όλα τα στοιχεία των δυο αρχείων αλλά στην περίπτωση μας δεν θα γράψουμε τα διπλότυπα στο αρχείο εξόδου.

Ο αλγόριθμος συνεχίζει για όσο υπάρχουν γραμμές σε οποιοδήποτε αρχείο

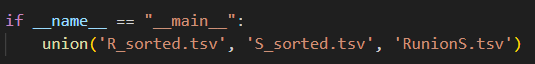
Αν σε ένα αρχείο τελειώσουν οι γραμμές, συνεχίζουμε να γράφουμε τις εναπομείνασες γραμμές από το άλλο αρχείο.

Εάν η τρέχουσα γραμμή από το αρχείο R προηγείται της τρέχουσας γραμμής από το αρχείο S (λεξικογραφικά), γράφει τη γραμμή από το R και προχωράει στην επόμενη γραμμή στο αρχείο αυτό.

Το ίδιο κάνουμε και για το αρχείο S.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.Εάν και οι δύο γραμμές είναι ίδιες, γράφουμε μόνο μια γραμμή στην έξοδο για να αποφύγουμε τα διπλότυπα και προχωράμε διαβάζοντας την επόμενη γραμμή και στα δυο αρχεία.

Η main του προγράμματος μας:

# Intersection

H τομή είναι όλα τα στοιχεία των αρχείων R και S τα οποία είναι κοινά και στα δυο αρχεία.

Αρχικά διαβάζουμε μια γραμμή τα δυο αρχεία. Σε κάθε επανάληψη, συγκρίνουμε πρώτα τα κλειδιά και από τις δύο τρέχουσες γραμμές.

Εάν το κλειδί από το αρχείο S είναι μικρότερο από το κλειδί από το αρχείο R, προχωράει στο αρχείο S

Εάν το κλειδί από το R είναι μικρότερο από το κλειδί από το S, προχωράει στο R για τον ίδιο λόγο.

Όταν τα κλειδιά ταιριάζουν, προχωράμε στη σύγκριση των τιμών:

Εάν οι τιμές διαφέρουν, προχωράμε στο κατάλληλο αρχείο για να βρούμε τις πιθανές αντιστοιχίες.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.Όταν τόσο τα κλειδιά όσο και οι τιμές ταιριάζουν, γράφουμε στο output αρχείο την αντίστοιχη γραμμή και προχωράμε τις γραμμές και στα δυο αρχεία.

Η main του προγράμματος μας:

Εικόνα που περιέχει στιγμιότυπο οθόνης, κείμενο, γραμματοσειρά

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

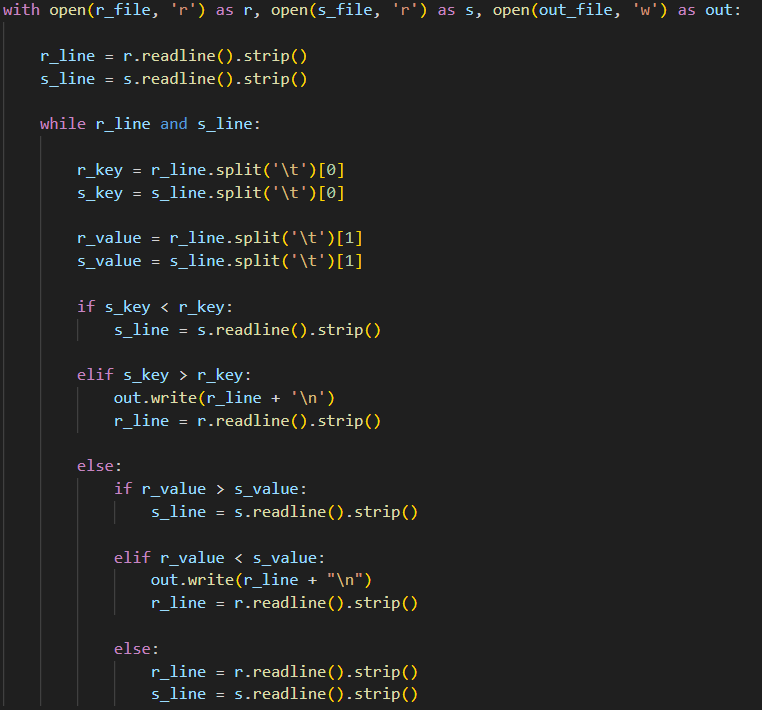
# Set-Difference

Σκοπός του προγράμματος είναι να γράψουμε στο output αρχείο όλα τα στοιχεία του R μείον τα στοιχεία του S. Δηλαδή δεν θα γράψουμε στο output αρχείο τα στοιχεία τα οποία υπάρχουν στο αρχείο του Intersection.

H υλοποίηση είναι παρόμοια με αυτή του Intersection.

Εάν το κλειδί στο αρχείο R είναι μικρότερο από το κλειδί στο αρχείο S, γράφουμε στην έξοδο αυτή την γραμμή και προχωράμε στην επόμενη γραμμή του αρχείου R.

Εάν το κλειδί στο αρχείο S είναι μικρότερο από το κλειδί στο R απλα προχωράμε στην επόμενη γραμμή του αρχείου S

Εάν τα κλειδιά είναι ίσα τοτε βλέπουμε τις αντίστοιχες τιμές. Εάν η τιμή στην γραμμή R είναι μεγαλύτερη από το αυτήν στο αρχείο S τοτε προχωράμε στις επόμενες τιμές του αρχείου S ενώ αν είναι μικρότερη αυτή στο αρχείο R τοτε γράφουμε την γραμμή αυτή στο output αρχείο και συνεχίζουμε στην επόμενη γραμμή του αρχείου R. Αν όλα στην γραμμή είναι ίδια τοτε απλα προχωράμε και στις δυο γραμμές των δυο αρχείων.

# Sort-Merge

Σκοπός αυτού του προγράμματος είναι να υλοποιήσουμε τον γνωστό αλγόριθμο merge-sort αλλά με μια αλλαγή. Αρχικά θα διαβάσουμε όλο το unsorted πρόγραμμα R σε μια λίστα και μετά 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.

Αρχικά έχουμε την βοηθητική συνάρτηση read\_line που χρησιμεύει για να διαβάζουμε την κάθε γραμμή του πίνακα μας και να εξάγουμε το κλειδί και την τιμή από κάθε γραμμή.

Η συνάρτηση merge\_sort\_and\_sum υλοποιεί την κλασικό merge sort αλγόριθμο. Χωρίζει αναδρομικά τον αρχικό πίνακα στα μισά, ταξινομεί κάθε μισό ανεξάρτητα και στη συνέχεια συνδυάζει τα αποτελέσματα χρησιμοποιώντας την συνάρτηση merge\_and\_group.

Έπειτα επεξεργαζόμαστε παράλληλα τους δύο ταξινομημένους υποπίνακες, επιλέγοντας πάντα την γραμμή με το μικρότερο κλειδί.

Για κάθε γραμμή, παρακολουθούμε το τρέχον κλειδί και διατηρούμε ένα άθροισμα για το κλειδί αυτό.

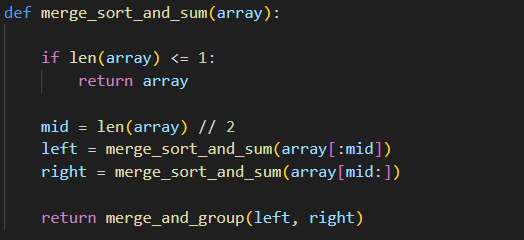
Όταν βρίσκουμε νέο κλειδί, βγάζουμε το προηγούμενο κλειδί και το άθροισμά του πριν αρχίσουμε να αθροίζουμε τις τιμές για το νέο κλειδί.

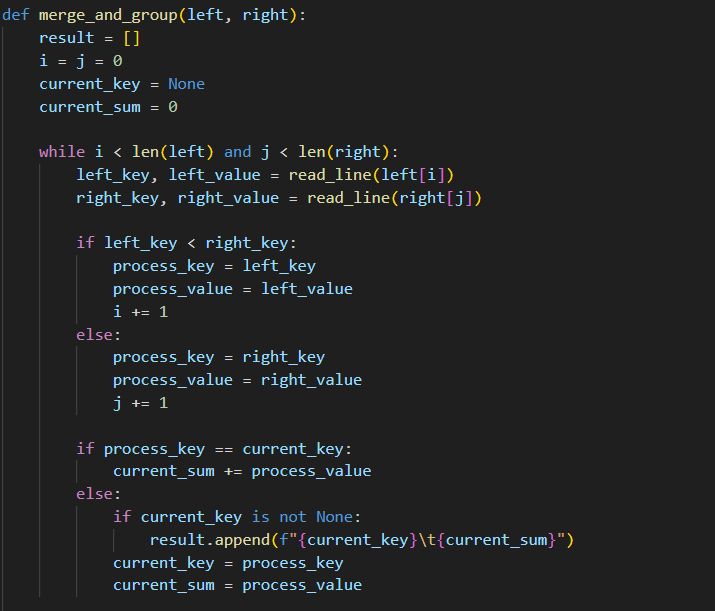
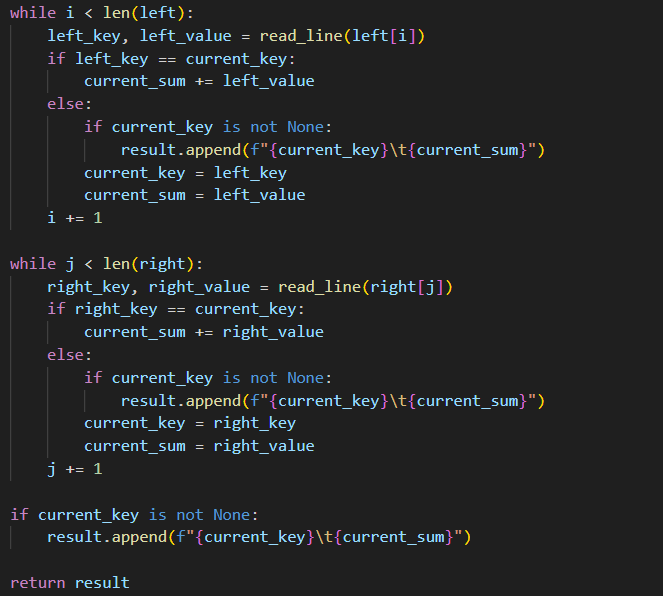
Άμα τελειώσει κάποιος υποπίνακας , συνεχίζουμε για τα υπόλοιπα στοιχεία του άλλου υποπίνακα, συνεχίζοντας να τις τιμές για αντίστοιχα κλειδιά.

Τέλος, ότι κλειδί έχει μείνει μαζί με το αποτέλεσμα του το βάζουμε στο αποτέλεσμα.

Εικόνα που περιέχει κείμενο, γραμματοσειρά, στιγμιότυπο οθόνης, σχεδίαση

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.





Η main του προγράμματός μας:

