ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ 2019 – 2020
ΑΣΚΗΣΗ 1

Νικόλαος Μακρυγεώργος	**1115201500238**
Αλέξανδρος Ζαφειρίου-Κωστούρος	**1115201500040**
Παναγιώτης Σταυρόπουλος	**1115201500150**

Μεταγλώττιση και εκτέλεση :

Η μεταγλώττιση γίνεται μέσω του makefile γράφοντας make όπου παράγεται το εκτελέσιμο με όνομα main
Η εκτέλεση γίνεται με ορίσματα στην γραμμή εντολών τα δύο αρχεία με τις εγγραφές και το μέγεθος κάθε αρχείου σε γραμμές.
Τα ορίσματα μπορούν να δοθούν με οποιαδήποτε σειρά.

π.χ.
main -f1 table_R_1000000 -s1 1000000 -f2 table_S_1000000 -s2 1000000

Διαγραφή εκτελέσιμου και αντικειμενικών αρχείων γράφοντας make clean

Τα αρχεία πρέπει να περιέχουν μόνο το key καθώς το payload το δίνουμε εμείς.
Απο τα dataset που δόθηκαν κρατήσαμε μόνο την πρώτη στήλη (ως key) και διαγράψαμε την δεύτερη για να δουλέψει στο πρόγραμμα μας.
Έχουμε ανεβάσει τα αρχεία tiny και small στο git μετά την αφαίρεση της δεύτερης στήλης.

Μετρήσεις Χρόνου:

Για 2 σχέσεις με 1000000 εγγραφές η κάθε μια έχουμε 0,464 seconds.
Για 2 σχέσεις με 2000000 εγγραφές η κάθε μια έχουμε 0,642 seconds.
Για 2 σχέσεις με 5000000 εγγραφές η κάθε μια έχουμε 1,722 seconds.
Για 2 σχέσεις με 10000000 εγγραφές η κάθε μια έχουμε 3,744 seconds.

Περιγραφή Προγράμματος:

main.c :

Αρχικά καλούμε την συνάρτηση take_arguments() η οποία παίρνει τα ορίσματα απο την γραμμή εντολών και τα βάζει σε μεταβλητές
Τα ορίσματα είναι τα ονόματα δύο αρχείων τα οποία περιέχουν τις εγγραφές για τις σχέσεις και δύο ακέραιοι που είναι το πλήθος των γραμμών των αρχείων αντίστοιχα.
Στην συνέχει δημιουργούνται(με malloc) οι πίνακες – σχέσεις και αρχικοποιούνται με τις τιμές απο τα αρχεία μέσω της συνάρτησης create_init_relations().
Ακολουθούν δύο κλήσεις της συνάρτησεις recurseFunc() μια για κάθε σχέση η οποία ταξινομεί τις σχέσεις.
Μετα τις ταξινομήσεις των δυο πινάκων – σχέσεων καλέιται η Sort_Merge_Join() όπου κάνει το join των σχέσεων και εισάγει τα αποτελέσματα στην λίστα.
Στο τέλος εκτυπώνουμε την λίστα που περιέχει τα αποτελέσματα του join και κάνουμε διαγραφή όλων των πινάκων- σχέσεων και της λίστας.
Επίσης εκτυπώνεται ο cpu time και τα αποτελέσματα της λίστα εκτυπώνονται σε αρχείο με όνομα results.csv.

Μέσα στην main περιέχονται και οι κατάλληλες εντολές για τα tests τα οποία επίσης εκτυπώνονται στο τέλος.

recurseFunc():

Για την ταξινόμηση των σχέσεων δημιουργήσαμε την αναδρομική συνάρτηση recurseFunc().
Παίρνει ορίσματα τον πίνακα – σχέση R, τον βοηθητικό πίνακα R', το διάστημα του πίνακα που θα γίνει η ταξινόμηση και έναν ακέραιο που μας λέει με βάση ποια bit του key θα δημιουργηθεί το ιστόγραμμα.
Εντός της συνάρτησης:
Γίνεται έλεγχος αν το διάστημα που δόθηκε είναι μεγαλύτερο του 4096
(4096 x 16 = 65536 = 64KB), αν όχι καλύται η quicksort για το διάστημα αυτο, αν ναι τότε σημαίνει οτι για το διάστημα αυτο θα πρέπει να δημιουργηθεί ο R' αρα υπολογίζουμε το ιστοστόγραμα, το p_sum και φτιάχνουμε τον R'.
Στην συνέχεια για κάθε ένα bucket που δημιουργήθηκε στον R' καλύται η αναδρομική συνάρτηση όπου επαναλαμβάνεται η ίδια διαδικασία.Τα διαστήματα για κάθε bucket υπολογίζονται με βάση το ιστόγραμμα και το p_sum.

Sort_Merge_Join():

Παίρνει σαν όρσμα τις δύο σχέσεις και την λίστα όπου θα κάνουμε εισαγωγή τα αποτελέσματα του join.
Το join γίνεται διασχίζοντας παράλληλα τους πίνακες.
Έχουμε δύο δείκτες στην αρχή κάθε σχέσης.Όσο το στοιχείο κάθε σχέση που δείχνει ο δείκτης είναι μικρότερο απο το στοιχείο της άλλης σχέσης που δείχνει ο δείκτης τότε ο δείκτης αυξάνεται για να πάμε στο επόμενο στοιχείο.Αντίστοιχα γίνεται και για τον άλλον δείκτη της δεύτερης σχέσης.
Οταν τα key της μιας και της άλλης σχέσεις είναι ίδια τότε κάνουμε εισαγωγή στην λίστα τα payload των keys.

Η λίστα που χρησιμοποιήθηκε :

Έγινε υλοποίηση μιας λίστας με πληροφοριακό κόμβο με δύο δείκτες στον αρχικό και τον τελευταίο κόμβο της λίστας.Με αυτό τον τρόπο έχεουμε εισαγωγή εγγραφής και κόμβου σε O(1) χρόνο.
Για κάθε εισαγωγή εγγραφής (κάθε εγγραφή θεωρείτε δύο uint64_t αριθμοί που συμβολίζουν την γραμμή απο το ένα και το άλλο αρχείο αντίστοιχα οπου γίνεται το join) πάμε στον τελευταίο κόμβο της λίστας αμέσως εφόσον έχουμε δείκτη τέλους και κάνουμε την εισαγωγή.Στην περίπτωση που δεν υπάρχει χώρος για νέα εγγραφή εισάγουμε νέο κόμβο και κάνουμε εισγωγή της εγγραφής.

Κάθε κόμβος περιέχει έναν 2D πίνακα uint64_t, ένα int για το μέγεθος του πίνακα, ένα int για τη θέση της επόμενη εγγραφής στον πίνακα και ένα δείκτη για τον επόμενο κόμβο.

Εφόσον υπάρχουν αποτελέσματα στο join η εκτύπωση γίνεται σε αρχείο με όνομα results.csv
Η εκτύπωση και η διαγραφής της λίστα χρειάζεται χρόνο O(n).

Tests:

Tests έγιναν για τον αν οι σχέσεις έχουν ταξινομηθεί σωστά.

Υπάρχουν σχόλια στον κώδικα όπου είναι απαραίτητο.
Επίσης θα διευκρινιστεί οποιδήποτε απορία στην προφορική εξέταση.