Εισαγωγή.

Η εργασία υλοποιεί το 100% της λειτουργικότητας του ερωτήματος Α της εργασίας, ενώ από το δεύτερο ερώτημα υλοποιήθηκαν όλα εκτός από το κομμάτι που αφορά τις καμπύλες, δηλαδή η LSH Frechet στο assignment και ο Mean καμπυλών στο update, καθαρά λόγω έλλειψης χρόνου. Ακολουθείται το 100% των ζητούμενων προδιαγραφών. Έγινε η μέγιστη δυνατή προσπάθεια βελτιστοποίησης πρωτίστως στο κομμάτι της ταχύτητας και εν συνεχεία στο κομμάτι εξοικονόμησης μνήμης. Τα προγράμματα χειρίζονται άριστα τη μνήμη. Οι έλεγχοι με valgrind καταδεικνύουν μηδενικές απώλειες. Ο σχολιασμός του κώδικα είναι πλούσιος. Η μεταγλώττιση δεν έχει warnings όταν μεταγλωττίζουμε με σημαία –Wall.

Το παρόν έγγραφο περιέχει τα ελάχιστα ζητούμενα από την εκφώνηση στοιχεία του readme και μία επιπλέον παράγραφο στο τέλος, στην οποία εξηγούμε συνοπτικά τις πρακτικές βελτιστοποίησης που ακολουθήσαμε.

Στοιχεία φοιτητών.

Θεόδωρος Κωνσταντόπουλος - 201600266

Ιωάννης Ταράτσας - 201700160

https://github.com/GiannisTrt/projectALAP/tree/Project2_/Part_2

Τίτλοι και περιγραφές των προγραμμάτων.

Παραδίδονται δύο κατάλογοι αρχείων. Ο πρώτος περιέχει αρχεία του search που υλοποιεί το ζήτημα Α ενώ ο δεύτερος περιέχει αρχεία του cluster που υλοποιεί το ζήτημα Β. Αναλυτικότερα:

α) search

Περιγραφή: Υλοποιεί το πρώτο υπο-ερώτημα του ζητήματος Α, ακριβώς όπως ζητείται σε όλα τα επίπεδα.

β) cluster

Περιγραφή: Υλοποιεί το ζήτημα Β, ακριβώς όπως ζητείται χωρίς όμως να υλοποιεί Frechet στο assignment και Mean καμπυλών στο update.

Κατάλογος των αρχείων κώδικα / επικεφαλίδων και περιγραφή τους.

Όπως είπαμε παραπάνω παραδίδονται δύο κατάλογοι αρχείων. Στον πρώτο κατάλογο παράγεται το πρόγραμμα search. Περιέχει τα εξής αρχεία:

1) Αρχείο κώδικα search_args.cc. Υλοποιεί το διάβασμα και έλεγχο των ορισμάτων του χρήστη από τη γραμμή εντολών. Εντοπίζει όλα τα σφάλματα και τα εμφανίζει στην οθόνη. Ακολουθεί πιστά τις προδιαγραφές της εκφώνησης.
2) Αρχείο επικεφαλίδας search_args.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
3) Αρχείο κώδικα bucket.cc. Υλοποιεί τη λειτουργικότητα ενός κάδου κατακερματισμού. Χρησιμοποιείται τόσο στο πρόγραμμα lsh όσο και στο πρόγραμμα cube.

4) Αρχείο επικεφαλίδας bucket.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
5) Αρχείο κώδικα hash.cc. Υλοποιεί τη λειτουργικότητα ενός πίνακα κατακερματισμού.
6) Αρχείο επικεφαλίδας hash.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
7) Αρχείο κώδικα list.cc. Υλοποιεί τη λειτουργικότητα μίας λίστας ακεραίων. Όπως θα εξηγήσουμε παρακάτω, το αρχείο φτιάχτηκε για λόγους βελτίωσης της ταχύτητας των προγραμμάτων μας.
8) Αρχείο επικεφαλίδας list.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
9) Αρχείο κώδικα metrics.cc. Υλοποιεί τη λειτουργικότητα της ζητούμενης μετρικής. Ωστόσο επειδή η μετρική μας περνιέται ως όρισμα στις συναρτήσεις που τη χρειάζονται, τα προγράμματά μας είναι απόλυτα επεκτάσιμα και είναι πολύ εύκολο να αλλάζουμε μετρικές προσθέτοντάς τες σε αυτό το αρχείο.
10) Αρχείο επικεφαλίδας metrics.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
11) Αρχείο κώδικα vector.cc. Υλοποιεί τη λειτουργικότητα του πίνακα διανυσμάτων ο οποίος υλοποιείται ως δισδιάστατος πίνακας ακεραίων. Το αρχείο περιέχει συναρτήσεις διαβάσματος του πίνακα από αρχείο διανυσμάτων, εμφάνισης και απελευθέρωσης της σχετικής μνήμης.
12) Αρχείο επικεφαλίδας vector.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
13) Αρχείο κώδικα curve.cc. Υλοποιεί τη λειτουργικότητα του πίνακα καμπυλών ο οποίος υλοποιείται ως δισδιάστατος πίνακας ακεραίων. Το αρχείο περιέχει συναρτήσεις διαβάσματος του πίνακα από αρχείο καμπυλών, εμφάνισης και απελευθέρωσης της σχετικής μνήμης.
14) Αρχείο επικεφαλίδας curve.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
15) Αρχείο κώδικα config.cpp. Ενσωματώνεται στον κώδικά μας, στο πλαίσιο της ενσωμάτωσης της Frechet ως μαύρο κουτί.
16) Αρχείο επικεφαλίδας config.hpp.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
17) Αρχείο κώδικα curve.cpp. Ενσωματώνεται στον κώδικά μας, στο πλαίσιο της ενσωμάτωσης της Frechet ως μαύρο κουτί. Υλοποιεί μια καμπύλη στο format που ο δοσμένος κώδικας χρειάζεται. Στο πλαίσιο της ενσωμάτωσης προκειμένου να καλέσουμε αποτελεσματικά τη frechet, μετατρέπουμε την εκάστοτε καμπύλη από το δικό μας format σε αυτό που ορίζεται στο εν λόγω module.
18) Αρχείο επικεφαλίδας curve.hpp.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
19) Αρχείο κώδικα frechet.cpp. Ενσωματώνεται στον κώδικά μας, στο πλαίσιο της ενσωμάτωσης της Frechet ως μαύρο κουτί. Υλοποιεί τη frechet.
20) Αρχείο επικεφαλίδας frechet.hpp.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
21) Αρχείο κώδικα interval.cpp. Ενσωματώνεται στον κώδικά μας, στο πλαίσιο της ενσωμάτωσης της Frechet ως μαύρο κουτί.
22) Αρχείο επικεφαλίδας interval.hpp.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.

23) Αρχείο κώδικα point.cpp. Ενσωματώνεται στον κώδικά μας, στο πλαίσιο της ενσωμάτωσης της Frechet ως μαύρο κουτί. Υλοποιεί ένα σημείο καμπύλης στο format που ο δοσμένος κώδικας χρειάζεται. Στο πλαίσιο της ενσωμάτωσης προκειμένου να καλέσουμε αποτελεσματικά τη frechet, για να μετατρέπουμε την εκάστοτε καμπύλη από το δικό μας format σε αυτό που ορίζεται στο εν λόγω module, πρέπει να λάβουμε υπόψη μας το format του point.
24) Αρχείο επικεφαλίδας point.hpp.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
25) Αρχείο κώδικα simplification.cpp. Ενσωματώνεται στον κώδικά μας, στο πλαίσιο της ενσωμάτωσης της Frechet ως μαύρο κουτί.
26) Αρχείο επικεφαλίδας simplification.hpp.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
27) Αρχείο επικεφαλίδας types.hpp. Αρχείο επικεφαλίδας με τους τύπους που χρειάζεται και χειρίζεται ο ενσωματωμένος κώδικας.

Στον δεύτερο κατάλογο παράγεται το πρόγραμμα cluster. Περιέχει τα εξής αρχεία:

1) Αρχείο κώδικα assign.c. Υλοποιεί τους τρεις αλγορίθμους assign (λειτουργικότητα μόνο, όχι δομές δεδομένων, διότι οι δομές δεδομένων υλοποιούνται σε ξεχωριστά αρχεία – modules όπως και στο ζήτημα Α).
2) Αρχείο επικεφαλίδας assign.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
3) Αρχείο κώδικα bucket.c. Υλοποιεί τη λειτουργικότητα ενός κάδου κατακερματισμού. Χρησιμοποιείται τόσο στον lsh όσο και στον υπερκύβο.
4) Αρχείο επικεφαλίδας bucket.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
5) Αρχείο κώδικα cluster_args.c. Υλοποιεί το διάβασμα και έλεγχο των ορισμάτων του χρήστη από τη γραμμή εντολών. Εντοπίζει όλα τα σφάλματα και τα εμφανίζει στην οθόνη. Ακολουθεί πιστά τις προδιαγραφές της εκφώνησης. Επίσης υλοποιεί και το διάβασμα των παραμέτρων από το αρχείο configuration με τις ίδιες προδιαγραφές.
6) Αρχείο επικεφαλίδας cluster_args.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
7) Αρχείο κώδικα constants.c. Αποθηκεύει χρήσιμες σταθερές.
8) Αρχείο επικεφαλίδας hamming.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
9) Αρχείο επικεφαλίδας hamming.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
10) Αρχείο κώδικα hash.c. Υλοποιεί τη λειτουργικότητα ενός πίνακα κατακερματισμού. Χρησιμοποιείται τόσο στον αλγόριθμο lsh όσο και στον αλγόριθμο hypercube.
11) Αρχείο επικεφαλίδας hash.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
12) Αρχείο κώδικα init.c. Υλοποιεί τον αλγόριθμο k-means++ ο οποίος χρησιμοποιείται στην αρχικοποίηση της συσταδοποίησης.
13) Αρχείο επικεφαλίδας init.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
14) Αρχείο κώδικα list.c. Υλοποιεί τη λειτουργικότητα μίας λίστας ακεραίων. Όπως θα εξηγήσουμε παρακάτω, το αρχείο φτιάχτηκε για λόγους βελτίωσης της ταχύτητας των προγραμμάτων μας.

15) Αρχείο επικεφαλίδας list.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
16) Αρχείο κώδικα metrics.c. Υλοποιεί τη λειτουργικότητα της ζητούμενης μετρικής. Ωστόσο επειδή η μετρική μας περνιέται ως όρισμα στις συναρτήσεις που τη χρειάζονται, τα προγράμματά μας είναι απόλυτα επεκτάσιμα και είναι πολύ εύκολο να αλλάζουμε μετρικές προσθέτοντάς τες σε αυτό το αρχείο.
17) Αρχείο επικεφαλίδας metrics.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
18) Αρχείο κώδικα update.c. Υλοποιεί την ενημέρωση (update)της συσταδοποίησης.
19) Αρχείο επικεφαλίδας update.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
20) Αρχείο κώδικα vector.c. Υλοποιεί τη λειτουργικότητα του πίνακα διανυσμάτων ο οποίος υλοποιείται ως δισδιάστατος πίνακας ακεραίων. Το αρχείο περιέχει συναρτήσεις διαβάσματος του πίνακα από αρχείο διανυσμάτων, εμφάνισης και απελευθέρωσης της σχετικής μνήμης.
21) Αρχείο επικεφαλίδας vector.h.Το αρχείο επικεφαλίδας που αντιστοιχεί στο παραπάνω αρχείο κώδικα.
22) Αρχείο κώδικα cube.c. Το αρχείο που φιλοξενεί την main του προγράμματος cluster και υλοποιεί τη λειτουργικότητα του ζητήματος Β.

Οδηγίες μεταγλώττισης των προγραμμάτων.

Για κάθε ένα από τα δυο ζητήματα παραδίδονται σχετικά makefile. Για να μεταγλωττιστούν τα δύο προγράμματα του ζητήματος Α, εκτελούμε την εντολή make στον πρώτο κατάλογο. Για να μεταγλωττιστεί το πρόγραμμα του ζητήματος Β, εκτελούμε την εντολή make στον πρώτο κατάλογο.

Οδηγίες χρήσης των προγραμμάτων.

Τα προγράμματα εκτελούνται ακριβώς σύμφωνα με τις προδιαγραφές της εκφώνησης. Η διάταξη των ζευγών σημαία – όρισμα, μπορεί να αλλάζει σύμφωνα με τις επιθυμίες του χρήστη.

Θα πρέπει στον κατάλογο του ζητήματος Α να υπάρχουν αρχεία configuration με ονόματα lsh.conf, cube.conf και search.conf. Είτε αυτά που παραδίδονται, είτε άλλα στο ίδιο format. Επίσης θα πρέπει να υπάρχουν αρχεία εισόδου και επερωτήσεων είτε αυτά που μας δώσατε, είτε άλλα, αλλά στο ίδιο format.

Στον κατάλογο του ζητήματος Β, θα πρέπει να υπάρχει αρχείο configuration με ονόματα cluster.conf. Είτε αυτό που παραδίδεται, είτε άλλο στο ίδιο format. Επίσης θα πρέπει να υπάρχει αρχείο εισόδου και επερωτήσεων είτε αυτό που μας δώσατε, είτε άλλο, αλλά στο ίδιο format.

Πρακτικές βελτιστοποίησης κώδικα.

- Όπως εξηγήσαμε και στην εισαγωγή ακολουθήσαμε ορισμένες πρακτικές βελτιστοποίησης οι οποίες αξίζει να αναφερθούν.
- Τα buckets στη φάση του διαβάσματος, υλοποιούνται ως λίστες ακεραίων (αριθμών διανυσμάτων). Ωστόσο όταν ολοκληρώνεται το διάβασμά τους, μετατρέπονται σε

πίνακες (για την ακρίβεια structs που περιέχουν δυναμικό πίνακα και το αντίστοιχο μήκος) προκειμένου η μετέπειτα πρόσβαση να είναι πολύ πιο γρήγορη.

- Χρησιμοποιούνται δηλώσεις register μεταβλητών στους αριθμοδείκτες.
- Χρησιμοποιείται σημαία –O3 στη μεταγλώττιση.
- Στη φάση του update θα πρέπει να υπολογίζουμε τις αποστάσεις όλων των διανυσμάτων από όλα τα υπόλοιπα διανύσματα ενός cluster. Προκειμένου να ελαχιστοποιήσουμε τους υπολογισμούς, φτιάχνουμε ένα δισδιάστατο τετραγωνικό πίνακα που περιέχει όλες τις αποστάσεις δηλαδή το στοιχείο στη θέση (i,j) περιέχει την απόσταση του i-οστού διανύσματος του cluster από το j-οστό. Μάλιστα για να γίνεται κάθε υπολογισμός μονάχα από μία φορά, υπολογίζουμε τις τιμές μονάχα του άνω τριγώνου του πίνακα, αφού το κάτω είναι συμμετρικό του άνω, στη λογική ότι η απόσταση του i-οστού διανύσματος από το j-οστό, είναι ίση με την απόσταση του j-οστού διανύσματος από το i-οστό. Τα στοιχεία της διαγωνίου είναι μηδενικά αφού η απόσταση ενός διανύσματος από τον εαυτό του είναι μηδενική.
- Η μετρική διοχετεύεται σαν όρισμα στις σχετικές συναρτήσεις. Με αυτόν τον τρόπο επιτυγχάνεται η μέγιστη δυνατή παραμετροποίηση στο κομμάτι αυτό.