

Εργασία 2

Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

Θεοφανόπουλος Μιχαήλ, 111 520 18 00 053

Δημησιάνος Σπυρίδων, 111 520 18 00 042

Οργάνωση αρχείων και φακέλων:

Φάκελος search: Περιέχει τα αρχεία mainSEARCH.cpp και mainSEARCH.h, τα οποία περιέχουν τη main συνάρτηση του Α σκέλους της εργασίας.

Φάκελος cluster: Περιέχει τα αρχεία mainCluster.cpp και mainCluster.h, τα οποία περιέχουν τη main συνάρτηση του Β σκέλους της εργασίας.

Φάκελος brute : Περιέχει τα αρχεία bruteFunctionsCUBE.cpp και bruteFunctionsLSH.cpp , τα οποία περιέχουν τις υλοποιήσεις των συναρτήσεων του HyperCube και αντιστοίχα του LSH με brute τρόπο (brute έρεση αποστάσεων μέσα απο τα αρχεία των δεδομένων)

Φάκελος data : Περιέχει το αρχείο items.cpp , το οποίο περιέχει τις υλοποιήσεις των μεθόδων της κλάσης item, καθώς και τα αρχεία Curve.cpp, Grid.cpp, GridContinuous.cpp τα οποία ομοίως περιέχουν τις υλοποιήσεις των μεθόδων των αντίστοιχων κλάσεων.

Φάκελος src : Περιέχει τα αρχεία util.cpp utilCLUSTER.cpp utilLSH.cpp , τα οποία περιέχουν το καθένα τις αντιστοιχες βοηθητικές συναρτήσεις για την υλοποίηση του Α και Β σκέλους της εργασίας αντίστοιχα. Εκτός από αυτά, περιέχει και τα αρχεία που χρειαστήκαμε από τον έτοιμο κώδικα για το μαύρο κουτί.

Φάκελος structures : Περιέχει τα αρχεία Cluster.cpp, G.cpp, H.cpp, Hash.cpp, HashTable.cpp και HyperCube.cpp τα οποία περιέχουν τις υλοποιήσεις των μεθόδων των αντίστοιχων κλάσεων.

Φάκελος tests : Περιέχει τα αρχεία με τα δεδομένα της εργασίας (queries , dataset) όπως επίσης και το cluster.conf που χρησιμοποιείται για την εισαγωγή των δεδομένων που απαιτούνται για το Clustering.

Φάκελος headers : Περιέχει όλα τα Header files , εκτος των headers των main συναρτήσεων που βρίσκονται στους αντιστοιχους cluster και search φακέλους.

Επίσης, έχουμε τα αρχεία Test.cpp, Test.h, TestMain.cpp τα οποία υλοποιούν 4 συναρτήσεις cpp unit testing πάνω στις κλάσεις curve και item.

Τέλος, έχουμε το **makefile** και το αρχείο **output.txt** για την συλλογή των εξόδων του προγράμματος.

Εκτέλεση του προγράμματος:

Η μεταγλώττιση του προγράμματος γίνεται με τη χρήση της εντολής:

> make

Η εκτέλεση του A σκέλους γίνεται με τη χρήση της εντολής:

> make out1

Η εκτέλεση του B σκέλους γίνεται με τη χρήση της εντολής:

> make out2

Η εκτέλεση του CPPUNIT γίνεται με τη χρήση της εντολής:

> make out3

Αν δεν επιθυμείται η χρήση του makefile για την εκτέλεση των προγραμμάτων, το πρότυπο των ορισμάτων είναι ως εξής:

> ./outSEARCH -i /tests/dataset.csv -q /tests/query.csv -k [int] -L [int] -M [int] -probes [int] -o output.txt -algorithm [string] -metric [string] -delta [double] για το A σκέλος.

> ./outCLUSTER -i /tests/dataset.csv -c /tests/cluster.conf -o output.txt -update [string] -assignment [string] -complete [optional] -silhouette [optional]

Περιγραφή των κλάσεων:

Η εργασία μας πατάει πάνω στον κώδικα από τη προηγούμενη εργασία μας, οπότε οι κλάσεις είναι σχεδόν οι ίδιες. Οι καινούργιες κλάσεις είναι οι εξής:

- Κλάση curve, αντιπροσωπεύει μια καμπύλη και περιέχει έναν πίνακα από item objects. Χρησιμοποιείται κυρίως στον discrete frechet όπου πρέπει να κρατάμε τα points στον 2d χώρο, οπότε μια καμπύλη έχει έναν πίνακα από items του τύπου [(1,2.5),(2,7.3)...].
- Κλάση Grid, αντιπροσωπεύει ένα grid το οποίο χρησιμοποιείται στο snapping για να μετατοπίσει τη καμπύλη. Κάθε Grid object κρατάει ένα double t για κάθε διάσταση, το οποίο επιλέγεται τυχαία uniformly και χρησιμοποιείται στον ξεχωριστό τύπο που έχει το κάθε Grid με σκοπό να μετατοπίσει το διάνυσμα πάνω στον άξονα.
- Κλάση GridContinuous, είναι το Grid αλλά χρησιμοποιείται στον continuous frechet. Η μόνη διαφορά είναι πως έχει ένα double t και έναν τύπο, καθώς στον continuous frechet ασχολούμαστε μόνο με μια διάσταση.
- Κλάση Continuous, αντιπροσωπεύει τη βασική δομή που χρησιμοποιούμε για την υλοποίηση του αλγορίθμου continuous frechet με LSH. Διατηρεί όσες μεταβλητές χρειάζεται για την αρχικοποίηση των δομών του καθώς και έναν πίνακα με τα αντικείμενα που γίνονται inserted(στην αρχική μορφή τους), ένα Grid και ένα LSH structure όπου γίνονται inserted τα αντικείμενα μετά το hashing τους.
- Κλάση Discrete, αντιπροσωπεύει τη βασική δομή που χρησιμοποιούμε για την υλοποίηση του αλγορίθμου discrete frechet με LSH. Διατηρεί όσες μεταβλητές χρειάζεται για την αρχικοποίηση των δομών του καθώς και έναν πίνακα με τα αντικείμενα που γίνονται inserted(στην αρχική τους μορφή), έναν πίνακα από Grids και έναν πίνακα από LSH structures όπου γίνονται inserted(στην αρχική τους μορφή) τα αντικείμενα μετά το hashing τους.

Γενική προσέγγιση:

Ο κώδικας περιέχει σχόλια με σκοπό να γίνεται εύκολη η διαδικασίας ανάγνωσής του.

Η προσέγγιση που ακολουθήσαμε είναι ίδια με αυτή της πρώτης εργασίας μας. Κατά την αρχικοποίηση των δομών (πχ. LSH, Hypercube, Cluster κλπ) η διαδικασία και οι τιμές τις οποίες επιλέξαμε να δίνουμε (πχ. w) είναι ακριβώς οι ίδιες. Καθώς μας είχαν δωθεί έτοιμα τα αρχεία dataset και query τα οποία θα χρησιμοποιηθούν στην εξέταση της εργασίας, έχουμε κάνει σε ορισμένα σημεία του Clustering κάποιες αλλαγές (πχ. Update exit statement, Assign exit statement) καθώς θέλαμε να βρούμε τα conditions που θα βελτιστοποιούν την εκτέλεση, με βάση πάντα τα αρχεία που δόθηκαν στο e-class του μαθήματος.

Σας ευχόμαστε καλή διόρθωση!