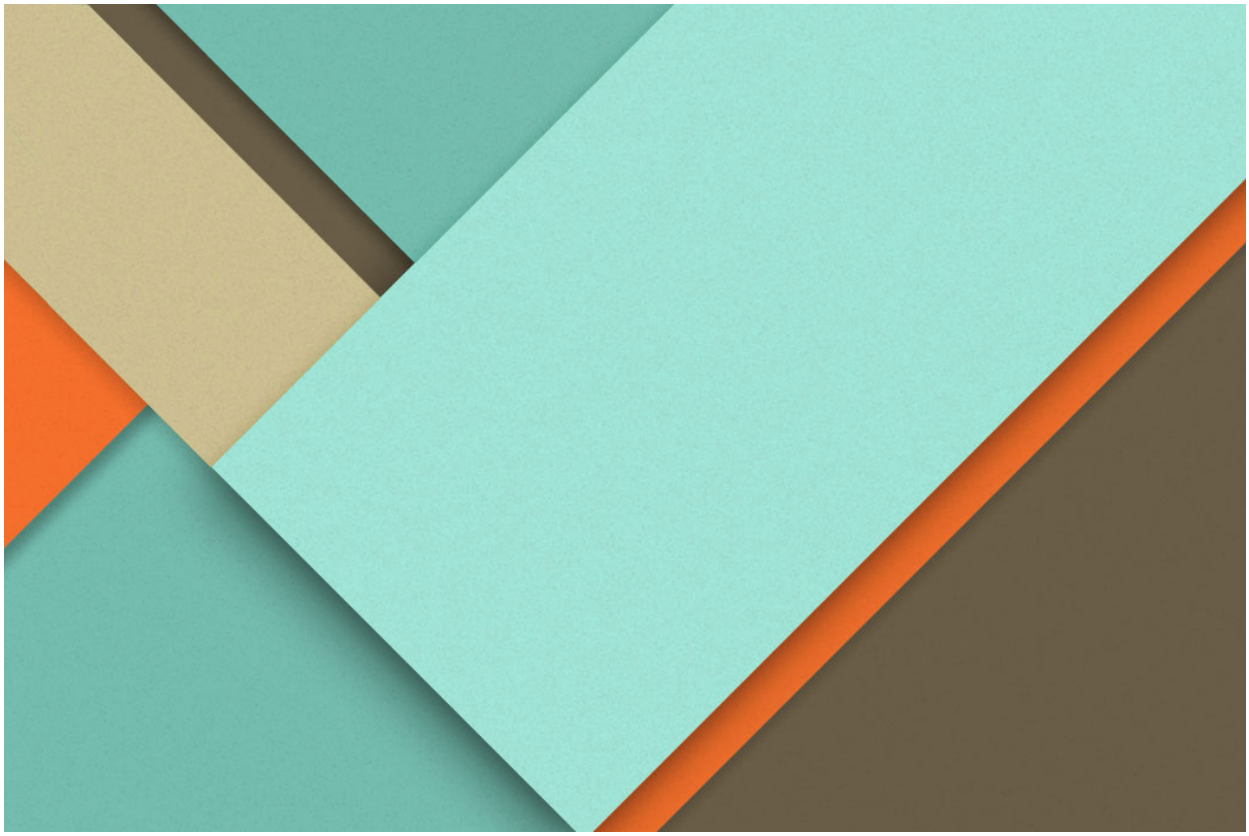

Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα Εργασία 1

Ονοματεπώνυμο: Μπαρμπάρ Ηλίας-Ελιάς
ΑΜ: 1115201200118

Συνοπτική Περιγραφή

Το πρόγραμμα υλοποιει τις τεχνικές LSH και HyperCube δοθεντος ενος συνολου εισοδου και ενος συνολου αναζητησης για δυο διαφορετικες μετρικες (cosine - euclidean) για να βρει: 1. Τους R-κοντινους γειτονες, τον approximate nearest neighbor και την αποσταση απο αυτον. Και στη συνεχεια να συγκρινει αυτα τα αποτελεσματα με το brute-force counterpart τους τον Exact Nearest Neighbor

Κατάλογος Αρχείων

myVector.hpp:

Ενα απλο αρχειο επικεφαλιδας που εχει το struct myVector που χρησιμοποιουμε ανα την ασκηση για τα διανυσματα.

Hfamily.hpp:

Ενα αρχειο επικεφαλιδας με την κλαση Hfamily την οποια κληρωνομουν οι κλασεις euclideanFamily και cosineFamily τις οποιες διαχωριζει και υποστηριζει με virtual συναρτησεις.

cosineH.hpp:

Αυτο το αρχειο επικεφαλιδας περιεχει τις κλασεις cosineH το οποιο εχει την υλοποιηση ενος μελους μιας οικογενειας H με βαση την μετρικη cosine και την κλαση cosineFamily ή οποια ειναι μια απο τις δυο κατηγοριες οικογενιων που εχουμε υλοποιησει και περιεχει την λογικη με την οποια οι cosine συναρτησεις παραγουν τα απαιτουμετα που πρεπει να παραγει μια Hfamily ασχετως μετρικης. (Τιμες για τις διαφορες H που παιρνει μια g, αποσταση)

cosineH.cpp:

Καθε cosineH φτιαχνεται βαζωντας D αριθμους κανονικης κατανομης σε ενα vector. Επισης υπολογιζεται η ποσοτητα του H δομενου ενας διανυσματος.
Οσο για την οικογενεια απλως φτιαχνει K τετοια σημεια, στελνει πισω αυτα τα σημεια σε vector<cosineH>.

euclideanH.hpp:

Αντιστοιχα και η euclideanH. Επιπλεον περιεχη και την κλαση Rfamily η οποια χρειαζεται για τον υπολογισμο της συναρτησης φ (καλο θα ηταν να μεταφερθει στο LSH)

euclideanH.cpp:

Οπως και στο cosine.cpp γινονται οι ιδιες διαδικασιες, απλως στη δημιουργια εδω θελουμε και ενα t απο ομοιομορφη κατανομη

gfunction.hpp:

Μια μικρη κλαση, η Gfunction, η οποια απλος περιεχει μια οικογενεια H και οποτε χρειαζεται κατι μεταφερει τον ελεγχο πιο κατω και γυρναει τα απαιτουμετα στοιχεια.

gfunction.cpp:

Κανει ακριβως οτι λενε τα ονοματα των συναρτησεων.

LSH.hpp:

Αυτο το αρχειο επικεφαλιδας εχει τρεις κλασεις HashNode και HashTable οι οποιες εχουν την πολυ απλη, κλασσικη υλοποιηση για hashtable και την κλαση LSH. Η LSH για τις κυριες συναρτησεις της rangeSearch και approxNNsearch απλως μεταφερει τον ελεγχο πιο κατω. Και η κυρια χρηση της ειναι για να κραταει το πληθος των hashtable και να τα κανει parse για τις διαφορες συναρτησεις. Τωρα το HashTable εχει αρχικα τις μεθοδολογιες υπολογισμου hash function με βαση την μετρικη (euclideanHashFunction - cosineHashFunction) τις οποιες χρησιμοποιει για τις τρεις κυριες συναρτησεις (putVector, approxNNsearch, rangeSearch)

LSH.cpp:

Οι συναρτησεις εχουν φτιαχτει αρκετα μινιμαλιστικα (θελω να πιστευω) οποτε δεν ειναι και περιπλοκες στην εξηγηση. Για τα hash function ζηταει απο την συναρτηση g να της φερει τα h απο τα οποια υπολογιζει με τις συναρτησεις που ξερουμε απο την θεωρια το hashvalue. Εφοσον εχει το hashvalue το χρησιμοποιει ειτε για να προσθεσει μια τιμη στο hashtable που γινεται με τον πιο απλο τροπο (parse + add). Ειτε να βρει την μικροτερη αποσταση, ειτε τους R γειτονες.

HyperCube.hpp:

Το HyperCube εχει ακριβως την ιδια δομη με το LSH + του HyperH που ειναι ενα key-value pair για ενα hashtable που κραταει το hypercube το οποιο περιεχει ολες τις αντιστοιχησεις απο συναρτησεις h σε 0 ή 1, το οποιο ειναι χρησιμο για την ωρα μονο για την euclidean αλλα στο μελλον ενδεχομενως και για αλλες μετρικες.

HyperCube.cpp:

Ακριβως τα ιδια με την LSH το μονο που διαφερει ειναι ο υπολογισμος του hashvalue

ExactNN.hpp / cpp:

Η γνωστη και πολυ απλη υλοποιηση του exact nearest neighbor με τον πιο απλο τροπο που υπαρχει (δηλαδη χωρις καμια βελτιωση)

Μεταγλώττιση

Για την μεταγλώττιση υπαρχει αρχειο makefile το οποιο εχει την εξης δομή:

```
all: lsh HC

lsh:
	g++ mainLSH.cpp LSH.cpp euclideanH.cpp cosineH.cpp gfunction.cpp ExactNN.cpp readFile.cpp -o lsh


HC:
	g++ mainHC.cpp HyperCube.cpp euclideanH.cpp cosineH.cpp gfunction.cpp ExactNN.cpp readFile.cpp -o HC
```