



Εργασία στο Μάθημα «Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα»

1η Εργασία: Αναζήτηση και Συσταδοποίηση Διανυσμάτων στη
C/C++

Ακαδ. Έτος 2021-2022
Ομάδα Χρηστών 52

	ΟΝΟΜΑΤΕΠΩΝΥΜΟ	A.M.	EMAIL
1	ΧΡΙΣΤΟΦΟΡΟΣ ΚΛΑΔΗΣ	1115201600068	sdi1600068@di.uoa.gr
2	ΑΡΙΣΤΟΤΕΛΗΣ ΒΕΡΓΙΝΗΣ	1115201700014	sdi1700014@di.uoa.gr

Github link: https://github.com/sdi160068/project_emiri_01

Περιεχόμενα

Περιεχόμενα	2
Περιγραφή του Προγράμματος	3
LSH	3
CUBE	3
Κατάλογος των αρχείων κώδικα / επικεφαλίδων και περιγραφή τους	3
vector_list.c / vector_list.h	4
vector.c / vector.h	4
random.c / random.h	4
mod.c / mod.h	4
loading.c / loading.h	4
hash.c / hash.h	4
data.c / data.h	4
HT.c / HT.h	4
cluster.c / cluster.h	5
Cluster_main.c	5
Οδηγίες μεταγλώττισης του προγράμματος	5
Οδηγίες χρήσης του προγράμματος	5

Περιγραφή του Προγράμματος

LSH

Αφού γίνει έλεγχος των παραμέτρων που δίνει ο χρήστης από την γραμμή εντολών, το πρόγραμμα φτιάχνει έναν πίνακα L θέσεων από hash tables και έναν πίνακα από g functions.

CUBE

Αρχικά, δημιουργούμε ένα hash table στο οποίο αποθηκεύονται όλα τα vectors. Όσον αφορά την f δεν φτιάχνουμε ξεχωριστές function, αλλά φτιάχνουμε μία η οποία έχει $k * h$ hash functions.

Κατάλογος των αρχείων κώδικα / επικεφαλίδων και περιγραφή τους

ΑΡΧΕΙΑ ΚΩΔΙΚΑ	ΑΡΧΕΙΑ ΕΠΙΚΕΦΑΛΙΔΑΣ
vector_list.c	vector_list.h
vector.c	vector.h
random.c	random.h
mod.c	mod.h
loading.c	loading.h
hash.c	hash.h
data.c	data.h
HT.c	HT.h
cluster.c	cluster.h
cluster_main.c	

vector_list.c / vector_list.h

Αρχείο για την ομαδοποίηση των vectors (διανύσματα στο χώρο) σε μια δομή λίστας. Περιέχει επίσης συναρτήσεις για την αναζήτηση κοντινότερων γειτόνων και για αναζήτηση ακτίνας.

vector.c / vector.h

Αρχεία με δομές για τα vectors που χρησιμοποιούνται στο παρόν project. Περιέχει και συναρτήσεις σύγκρισης απόστασης με μετρική L2 (dist_L2).

random.c / random.h

Αρχεία που περιέχουν συναρτήσεις για τυχαία νούμερα δοθέντος εύρους μέσω της τυχαίας γεννήτριας αριθμών. Υπάρχει και υλοποίηση μιας γεννήτριας τυχαίων αριθμών από την κανονική κατανομή, βασισμένη από την παρακάτω πηγή :

<https://mathworld.wolfram.com/Box-MullerTransformation.html>

mod.c / mod.h

Περιέχει την συνάρτηση mod, η οποία επιστρέφει το θετικό υπόλοιπο μιας διαίρεσης. Π.χ $\text{mod}(-10,4) = 2$, ενώ $-10\%2 = -2$

loading.c / loading.h

Περιέχουν συναρτήσεις για την επίδειξη της προόδου ενός loop για κάποιο μέγεθος. Δίνει αποτελέσματα σε %.

hash.c / hash.h

Περιέχουν υλοποιήσεις για τις hash functions που χρησιμοποιούνται στο παρόν project, όπως την h, f (για το hypercube) και g (για το lsh).

data.c / data.h

Συναρτήσεις και δομές για την ανάγνωση και αποθήκευση δεδομένων από αρχεία ή την γραμμή εντολών σε πίνακες από strings (char*) .

HT.c / HT.h

Δομές και συναρτήσεις για την δημιουργία και διαχείριση hash tables, των οποίων τα buckets είναι λίστες από τα vector_list.c vector_list.h .

cluster.c / cluster.h

Συναρτήσεις για την δημιουργία cluster. Περιέχονται υλοποιήσεις των αλγορίθμων kmeans++, Lloyd κλπ.

Οι υλοποιήσεις των kmeans++ και Lloyd είναι βασισμένες από την παρακάτω πηγή :

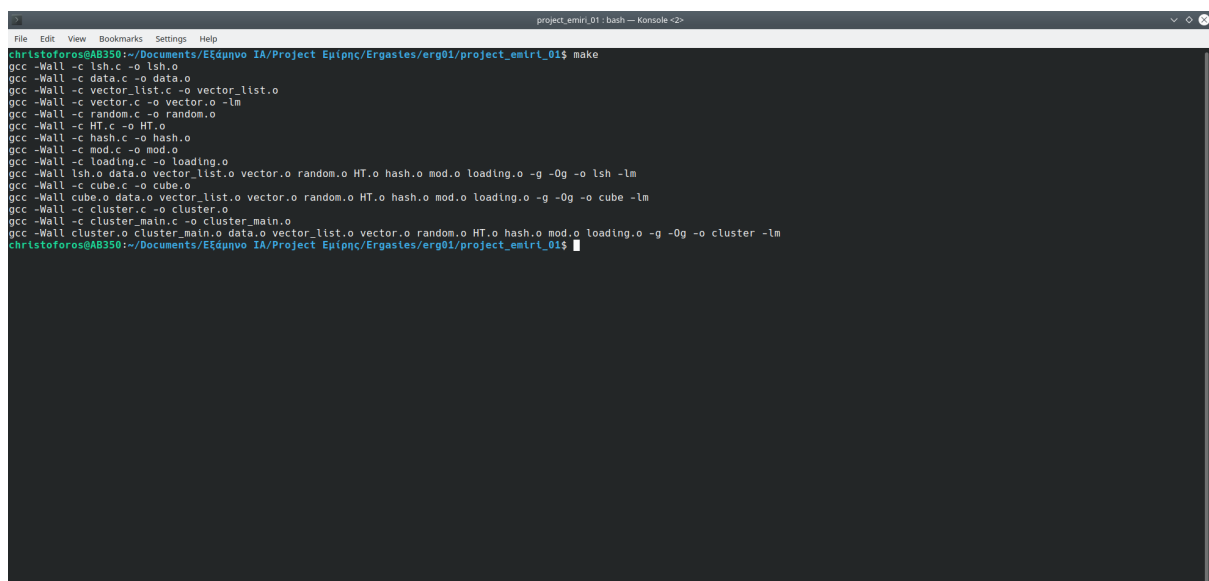
https://rosettacode.org/wiki/K-means%2B%2B_clustering

Cluster_main.c

Αρχείο με την main συνάρτηση για το εκτελέσιμο cluster. Ελέγχει τις παραμέτρους που έδωσε ο χρήστης και με βάση αυτές καλεί τις συναρτήσεις για την δημιουργία cluster.

Οδηγίες μεταγλώττισης του προγράμματος

Η μεταγλώττιση γίνεται με την εντολή make.



```
project_emiri_01: bash — Konsole <2>
File Edit View Bookmarks Settings Help
christoforos@AB350:~/Documents/Εξάμηνο ΙΑ/Project Ειρήνης/Ergasties/erg01/project_emiri_01$ make
gcc -Wall -c lsh.c -o lsh.o
gcc -Wall -c data.c -o data.o
gcc -Wall -c vector_list.c -o vector_list.o
gcc -Wall -c vector.c -o vector.o -lm
gcc -Wall -c random.c -o random.o
gcc -Wall -c HT.c -o HT.o
gcc -Wall -c hash.c -o hash.o
gcc -Wall -c mod.c -o mod.o
gcc -Wall -c loading.c -o loading.o
gcc -Wall lsh.o data.o vector_list.o vector.o random.o HT.o hash.o mod.o loading.o -g -Og -o lsh -lm
gcc -Wall -c cube.c -o cube.o
gcc -Wall cube.o data.o vector_list.o vector.o random.o HT.o hash.o mod.o loading.o -g -Og -o cube -lm
gcc -Wall -c cluster.c -o cluster.o
gcc -Wall -c cluster_main.c -o cluster_main.o
gcc -Wall cluster.o cluster_main.o data.o vector_list.o vector.o random.o HT.o hash.o mod.o loading.o -g -Og -o cluster -lm
christoforos@AB350:~/Documents/Εξάμηνο ΙΑ/Project Ειρήνης/Ergasties/erg01/project_emiri_01$
```

Οδηγίες χρήσης του προγράμματος

- Για τα εκτελέσιμα lsh, cube , ο χρήστης πρέπει να δίνει κατ ελάχιστον τα ονόματα των αρχείων που θα διαχειριστεί , δηλαδή των input_file, query_file και output_file. Για το cluster πρέπει να δίνει ένα αρχείο <configuration file> για την αρχικοποίηση του προγράμματος. Το αρχείο <configuration file> πρέπει να περιέχει τουλάχιστον την ετικέτα “number_of_clusters.” με την αντίστοιχη τιμή

- Για τα εκτελέσιμα lsh και cube , ο χρήστης επιλέγει , αφού τελειώσει η εξέταση του query_file, αν θα κάνει έξοδο (εντολή exit) ή αν θα δώσει καινούργιο query_file. Τα αποτελέσματα όλων των query_files που θα δοθούν αποθηκεύονται στο ίδιο output_file.
- Σε κάθε εκτελέσιμο μπορεί να δοθεί στην γραμμή εντολών ως παράμετρος και η τιμή της μεταβλητής w (-w <int>) για περαιτέρω πειράματα πάνω στα lsh και cube. (Υπενθύμιση: η w είναι το window που χρησιμοποιούμε για την δημιουργία των συναρτήσεων h_i).
- Στο cluster μπορεί να δοθεί στην γραμμή εντολών ως παράμετρος και η τιμή της μεταβλητής t (-t <int>) για το μέχρι πόσες φορές μπορεί να εκτελεστεί η εκάστοτε επιλεγείσα μέθοδος δημιουργίας των clusters (lsh, cube, Lloyd).