

Εργασία 1

Το πρόγραμμα αναπτύχθηκε στα πλαίσια του μαθήματος Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα 2019-2020 για το Τμήμα Πληροφορικής και Τηλεπικοινωνιών του ΕΚΠΑ.

Αρχεία

Γενικά

- **G.hpp, G.cpp**: Ορισμός και υλοποίηση της κλάσης G, η οποία αντιστοιχεί στις συναρτήσεις $g_i()$ των διαφανειών.
- **H.hpp, H.cpp**: Ορισμός και υλοποίηση της κλάσης H, η οποία αντιστοιχεί στις συναρτήσεις $h_i()$ των διαφανειών.
- **Point.hpp, Point.cpp**: Ορισμός και υλοποίηση της κλάσης Point, για την αναπαράσταση των σημείων.
- **dist.hpp, dist.cpp**: Συναρτήσεις υπολογισμού απόστασης μεταξύ σημείων και μεταξύ καμπυλών.
- **util.hpp, util.cpp**: Βοηθητικές συναρτήσεις.
- **Makefile**

Ζητούμενο A (i) + bonus

- **LSH/main.cpp**: Περιέχει την main για το ζητούμενο A (i).
- **LSH/LSH.hpp**: Ορισμός κλάσης LSH.
- **LSH/LSH.cpp**: Υλοποίηση των συναρτήσεων της LSH.

Ζητούμενο A (ii) + bonus

- **HyperCube/main.cpp**: Περιέχει την main για το ζητούμενο A (ii).
- **HyperCube/HC.hpp**: Ορισμός κλάσης HC.
- **HyperCube/HC.cpp**: Υλοποίηση των συναρτήσεων της HC.

Ζητούμενο B (i)

- **Curves/main.cpp**: Περιέχει την main για το ζητούμενο B (i).

- **Curves/Curve.hpp, Curves/Curve.cpp:** Ορισμός και υλοποίηση της κλάσης Curve, για την αναπαράσταση των καμπυλών.
- **Curves/Grid.hpp, Curves/Grid.cpp:** Ορισμός και υλοποίηση της κλάσης Grid, για τα grids.
- **Curves/CurveHashing.hpp, Curves/CurveHashing.cpp:** Ορισμός και υλοποίηση των κλάσεων CurveHashing, LSHC, HCC.

Compilation και Εκτέλεση

Η μεταγλώττιση του προγράμματος μπορεί να πραγματοποιηθεί με την εντολή

`make,`

καθώς υπάρχει ένα Makefile. Να σημειωθεί ότι χρησιμοποιείται το flag `-O3` παντού. Τα εκτελέσιμα που δημιουργούνται ονομάζονται `lshDemo`, `hypercubeDemo` και `curvesDemo`. Τα ορίσματα τους είναι τα εξής (τα **bold** είναι υποχρεωτικά):

```
/lshDemo -d [inputFile] -q [queryFile] -k [k] -L [L] -w [w] -r [r] -o [outputFile]
```

```
/hypercubeDemo -d [inputFile] -q [queryFile] -k [k] -dd [dd] -L [L] -w [w] -r [r] -M [M] -  
probes [probes] -o [outputFile]
```

Προσοχή: Σε αντίθεση με την εκφώνηση, εδώ το k είναι το πλήθος των h_i , και dd είναι το d' των διαφανιών. Αυτό ώστε να μπορούν να ρυθμιστούν το k του `lsh` από την γραμμή εντολών.

```
/lshDemo -d [inputFile] -q [queryFile] -hf [LSH ή HC] -k [k] -dd [dd] -L [L] -w [w] -M [M] -  
probes [probes] -o [outputFile]
```

Παράδειγμα:

```
/lshDemo -d Data/input_small_id -q Data/query_small_id -hf LSH -o output.txt
```

Περιγραφή Κλάσεων

- **Point**
Αντιπροσωπεύει ένα σημείο.
- **Curve**
Αντιπροσωπεύει μία καμπύλη. Αποτελείται κυρίως από ένα vector από Points.
- **H, G**
Για τις συναρτήσεις $h_i()$ και $g_i()$ των διαφανειών αντίστοιχα.
- **LSH**

Ο σκοπός αυτής της κλάσης είναι να αποθηκεύει τα απαραίτητα δεδομένα και συναρτήσεις για την εκτέλεση του αλγορίθμου LSH. Αποτελείται κυρίως από: το vector dataset στο οποίο αποθηκεύονται τα points που διαβάζονται από το αρχείο εισόδου, το vector g που περιέχει τις g_i , το vector hashTables για τα L hashTables (ένα για κάθε g_i). Η συνάρτηση **insert()** εισάγει ένα νέο σημείο στο dataset και στα hashTables. Αφού γίνει insert όλο το dataset, μπορούν να κληθεί η συνάρτηση **answerQuery**. Σε αυτήν καλούνται έμμεσα οι συναρτήσεις κοντινότερου γείτονα (**nearestNeighbour()**, **nearestNeighbours()**).

- **HC**

Η αντίστοιχη κλάση για την αλγόριθμο HyperCube. Όμοια με την *LSH*, αποτελείται κυρίως από: το vector dataset στο οποίο αποθηκεύονται τα points που διαβάζονται από το αρχείο εισόδου, το vector g που περιέχει τις g_i , το vector f που περιέχει τις f_i (μία για κάθε g_i) και το hashtable cube. Οι συναρτήσεις δουλεύουν με όμοιο τρόπο με την *LSH*.

- **Grid**

Η κλάση αντιστοιχεί στα G πλέγματα των διαφανειών.

- **CurveHashing**

Είναι μία abstract κλάση την οποία κληρονομούν οι *LSHC* και *HCC*. Στο vector allCurves αποθηκεύονται οι καμπύλες του αρχείου εισόδου. .

- **LSHC**

LSH για καμπύλες.

- **HCC**

HCC για καμπύλες.

Περιγραφή τρόπου προσέγγισης

Υπάρχουν επεξηγηματικά σχόλια σε όλη την έκταση του κώδικα, για να είναι εύκολη η ανάγνωση του. Ως αποτέλεσμα οι λεπτομέρειες την υλοποίησης είναι προφανείς από αυτά. Ωστόσο στη συνέχεια θα περιγράψουμε την γενική ιδέα.

LSH, HyperCube

Και οι δύο διαδικασίες ακολουθούν παρόμοια βήματα. Η κύρια λειτουργία των αλγορίθμων είναι υλοποιημένη στις κλάσεις LSH και HyperCube. Αρχικά δημιουργείται ένα στιγμιότυπο μιας από τις δύο κλάσεις. Στη συνέχεια, καθώς γίνεται η ανάγνωση του input dataset από αρχείο, εισάγονται points στα hashtable μέσω της συνάρτησης insert(). Η εισαγωγή γίνεται με βάση τα hashkey που παράγει ο κάθε αλγόριθμος. Τέλος για κάθε query καλείται η συνάρτηση answerQuery(), κατά την οποία βρίσκεται ο κοντινότερος ή οι κοντινότεροι (εντός ακτίνας r) γείτονες.

LSH, HyperCube σε καμπύλες

Τα βήματα που ακολουθεί η main είναι όμοια με τους αλγορίθμους που απευθύνονται σε σημεία.