

# ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΑΛΓΟΡΙΘΜΙΚΑ ΠΡΟΒΛΗΜΑΤΑ

## 1 η Προγραμματιστική Εργασία

### Υλοποίηση δομής για την εύρεση κοντινών γειτόνων στη γλώσσα C/C++

**ΟΝΟΜΑΤΕΠΩΝΥΜΟ:** Παπαγεωργίου Βασίλειος-Νικηφόρος  
**ΑΜ**=1115201400140

#### Γενικές Πληροφορίες:

Η εργασία μου υλοποιεί το ζητούμενο και είναι χωρισμένο σε 2 προγράμματα: το Lsh.cpp(φάκελος LSH) και το Cube.cpp(φάκελος Hypercube).Επίσης, σε κάθε φάκελο υπάρχουν το αρχείο επικεφαλίδας με τον ορισμό των κλάσεων, τα datasets που μας έχετε δώσει και το αρχείο: structures.cpp ,που περιέχει τις υλοποιήσεις των διαφόρων δευτερεύουσων και κύριων **ΔΙΚΩΝ ΜΟΥ** δομών(hashtable,lists καθώς επέλεξα να μην κάνω χρήση STL,διότι ήθελα να έχω πλήρη επίγνωση της κάθε δομής κι όχι να χρησιμοποιήσω κάτι έτοιμο) και αντίστοιχων συναρτήσεων της εργασίας.Επίσης στους φακέλους έχω φτιάξει και το αντίστοιχο Makefile και συνεπώς με την εντολή make μεταγλωττίζεται το αντίστοιχο πρόγραμμα και παράγονται τα εκτελέσιμα lsh,cube!Συνεπώς, τα πρόγραμμα μου εκτελούνται με την εντολή ./lsh ή ./cube, ακολουθούμενη από τις ζητούμενες παραμέτρους.Στην υλοποίηση μου, έχω δώσει περισσότερη βάση στο να δημιουργήσω έναν απόλυτα ολοκληρωμένο αλγόριθμο LSH, ενώ για τον υπερκύβο έχω κάνει απλά τις αρχικές διαδικασίες.

Για τον LSH, καταρχάς χρησιμοποιώ στις δοκιμές μου σταθερό  $w=400$  και  $ri$  στο διάστημα  $[0,800)$ . Αρχικά για  $k=4, L=5$  με το `input_small` και το `query_small`, το μέσο κλάσμα προσέγγισης βγαίνει κοντά στο 2,4 και ο μέσος χρόνος `tLsh` περίπου 0,0007 , που είναι περίπου 100 φορές μικρότερος απ τον μέσο `tReal` της εξαντλητικής αναζήτησης.Επίσης το μέσο κλάσμα είναι κοντά στο 1,1 οπότε έχουμε ικανοποιητική ακρίβεια.

Κρατώντας σταθερό το  $k=4$  ,πειραματιζόμαστε με το  $L$ :

Αυξάνοντας το  $L$  παρατηρούμε ότι το μέγιστο και το μέσο κλάσμα μειώνεται, η ακρίβεια αυξάνεται και ο χρόνος επίσης αυξάνεται(λογικό λόγω μεγαλύτερης σπατάλης χώρου)

Μειώνοντας το  $L$  παρατηρούμε ότι το μέγιστο και το μέσο κλάσμα μεγαλώνουν, η ακρίβεια μειώνεται(χειροτερεύει) και ο χρόνος επίσης

μειώνεται(λογικό λόγω μικρης σπατάλης χώρου)

Κρατώντας σταθερό το  $L=5$ ,πειραματιζόμαστε με το  $k$   
Αυξάνοντας το  $k$  παρατηρούμε ότι το μέγιστο και μέσο κλάσμα  
μεγαλώνουν,οπότε η ακρίβεια μειώνεται και ο χρόνος επίσης μειώνεται.  
Μειώνοντας το  $k$  παρατηρούμε ότι το μέγιστο και μέσο κλάσμα  
μειώνονται,οπότε η ακρίβεια μεγαλώνει και ο χρόνος επίσης αυξάνεται.

ΣΥΜΠΕΡΑΣΜΑ:Τα  $L,k$  έχουν αντίστροφη επιρροή στα αποτελέσματα του  
LSH

\*Οι παραπάνω μετρήσεις έγιναν με ευκλείδια μετρική.Η γενική παρατήρηση  
για την cosine μετρική είναι ότι για μικρά  $k$  και μεγάλο tablesize πλησιάζει την  
εξαντλητική αναζήτηση σε χρόνο και ακρίβεια,παραμένοντας φυσικά ταχύτερη  
από αυτήν.

Για την μορφή των αρχείων έχω κάνει την παραδοχή ,οτι η πρώτη γραμμή των  
input **πάντα** θα εμπεριέχει την μετρική στη μορφή :@{metric} metric,ενώ τα  
query sets **πάντα** την ακτίνα Radius: number.Τα paths των αρχείων,δίνονται  
μέσω της γραμμής εντολών,από τις αντίστοιχες παραμέτρους.Τα προγράμματα  
έχουν ελεγχθεί και με valgrind ,το οποίο πιστοποιεί το ότι δεν υπάρχουν  
memory leaks,καθώς **τα allocations έχουν ακριβώς ίδιο πλήθος με τα frees!**

Περαιτέρω λεπτομέρειες παρέχονται σε σχόλια στα αρχεία και οτιδήποτε  
προκύψει είμαι διαθέσιμος να το αιτιολογήσω στην προφορική εξέταση!