Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

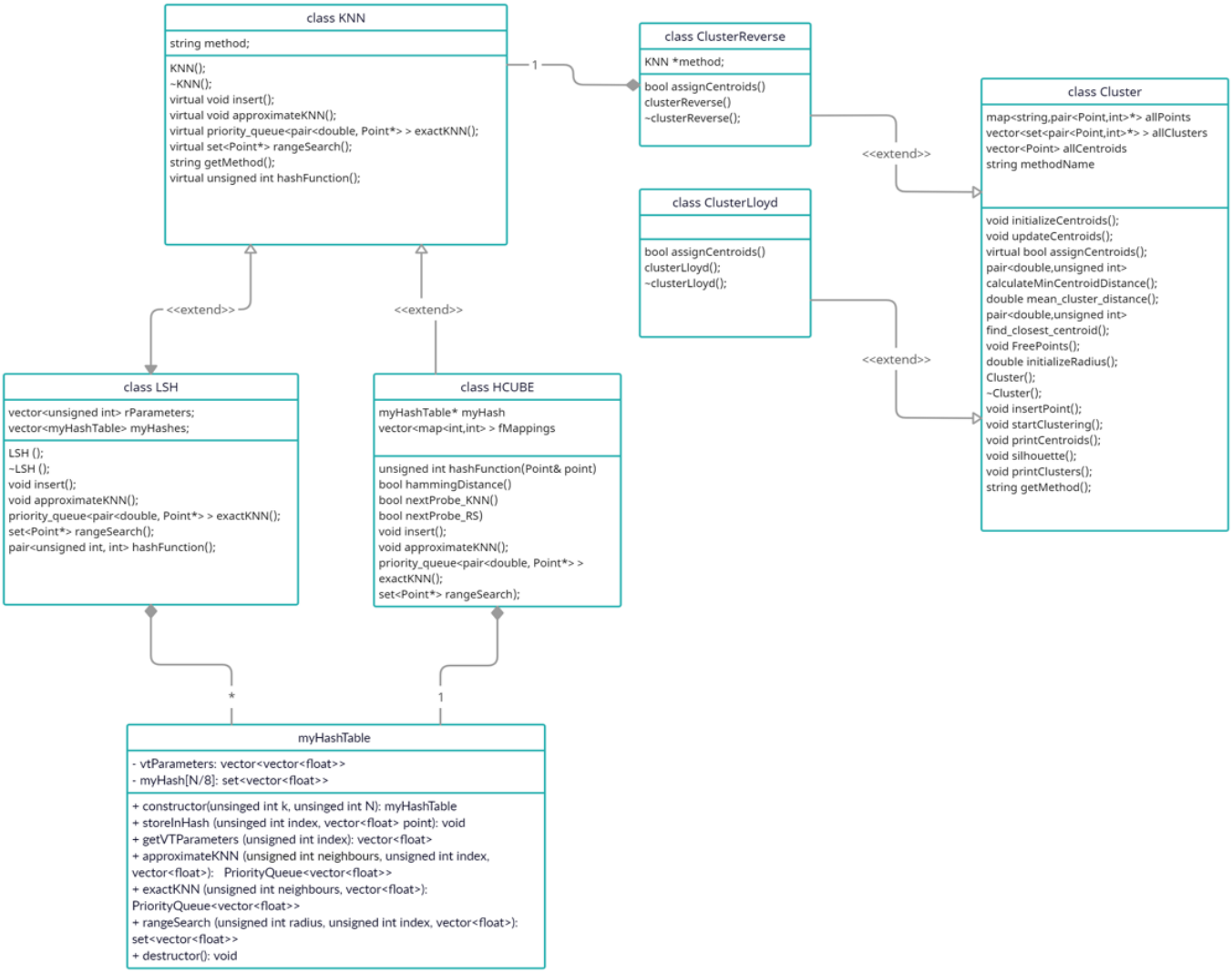
Εργασία 1

Δημητρακόπουλος Κωνσταντίνος (sdi1500034) & Κακαβάς Αντώνιος (sdi1500052)

Contents

Δημητρακόπουλος Κωνσταντίνος (sdi1500034) & Κακαβάς Αντώνιος (sdi1500052)

Διαμόρφωση προδιαγραφών – UML



ΣΗΜΕΙΩΣΗ: Το πρόγραμμα μας χωρίζεται σε 3 μεγάλες ενότητες(modules). Το module *"myHashTable"* αποτελείται από συναρτήσεις που βοηθούν στο στήσιμο των πινάκων κατακερματισμού που ύστερα θα χρησιμοποιηθούν από τους δύο βασικούς μας αλγόριθμους. Δεδομένου αυτού, με τη χρήση του συγκεκριμένου module αποφεύγονται τυχών επαναλήψεις κώδικα.

Κατάλογος & Ομαδοποίηση των αρχείων κώδικα

Δημητρακόπουλος Κωνσταντίνος (sdi1500034) & Κακαβάς Αντώνιος (sdi1500052)

Γενικά, όλα τα πηγαία αρχεία εμπεριέχουν σχόλια σχετικά με τον σχεδιασμό που έχει ακολουθηθεί.

Επίσης, επειδή οι αλγόριθμοι είναι τυχαιοκρατικοί μπορεί να χρειαστεί να γίνουν πολλαπλές εκτελέσεις για να εμφανίσουν τα αναμενόμενα αποτελέσματα.

1. **Όσο αναφορά τον LSH** , έχουν υλοποιηθεί τα παρακάτω αρχεία κώδικα:

   I) **lsh_main.cpp** → Το πηγαίο αρχείο που εμπεριέχει την main. Στο αρχείο αυτό διαχειριζόμαστε τα arguments του command line και καλούμε όλες τις επιμέρους συναρτήσεις που σχετίζονται με τον συγκεκριμένο αλγόριθμο.

   II) **lsh.h** ➔ Στο συγκεκριμένο header file υπάρχουν όλα τα πρωτότυπα των συναρτήσεων(declarations) που έχουν χρησιμοποιηθεί για την υλοποίηση του συγκεκριμένου αλγορίθμου.

   III) **lsh.cpp** → Στο συγκεκριμένο αρχείο έχει γίνει το definition όλων των συναρτήσεων που αφορούν τον lsh.

   Για την υλοποίηση του lsh (αλλά και του hypercube σε επόμενο βήμα, έχουν υλοποιηθεί και **κάποια έξτρα αρχεία κώδικα** τα οποία βοηθούν κυρίως στο καλύτερο μερισμό της εργασίας μας αλλά και στην αποφυγή διπλότυπων.) Αυτά είναι :

   I) **point.cpp + point.h** → Περιλαμβάνει το declaration αλλά και το definition μιας μικρής κλάσης που φτιάξαμε προκειμένου να μπορούμε λίγο πιο εύκολα να διαχειριζόμαστε το structure ενός σημείου , καθώς και να ξεχωρίζουμε το id του σημείου από την υπόλοιπη πληροφορία η οποία αντιπροσωπεύει τις διαστάσεις του.

   II) **utils.cpp + utils.h** → Περιλαμβάνει το declaration αλλά και το

definition κάποιων έξτρα συναρτήσεων που έχουν χρησιμοποιηθεί κατά την υλοποίηση των αλγορίθμων, κυρίως για επαναληπτικούς υπολογισμούς. Ο λόγος ύπαρξης των συγκεκριμένων συναρτήσεων είναι καθαρά για την καλύτερη οργάνωση του κώδικα και για την διατήρηση μιας αρμονικής επαναληπτικότητας.

III) **myHashTable.cpp + myHashTable.h** →Περιλαμβάνει το declaration αλλά και το definition όλων των συναρτήσεων που έχουν χρησιμοποιηθεί και συνδέονται με το συγκεκριμένο class. Οι συναρτήσεις που ανήκουν στο συγκεκριμένο class , έχουν προσαρμοστεί έτσι , ώστε να μπορούν να χρησιμοποιηθούν και από τον lsh αλλά και από τον randomized projection. Έτσι , καταφέραμε να έχουμε ένα βασικό core στο οποίο φτιάχνονται όλα τα βασικά κομμάτια γύρω από τους πίνακες κατακερματισμού και μετά απλά , κάθε αλγόριθμος ξεχωριστά καλεί και βασίζεται στις συναρτήσεις του myHashTable. Έτσι , γλυτώνουμε σε μεγάλο βαθμό τυχών διπλότυπα και περιττές επαναλήψεις

IV) **PQUnique.h + PQUnique.t.hpp** → Περιλαμβάνει το declaration αλλά και το definition του συγκεκριμένου template μιας ουράς προτεραιότητας , χωρίς διπλότυπα στοιχεία και με δυνατότητα επιλογής μεγίστου μεγέθους , η οποία χρησιμοποιείται για να κάνουμε store τα αποτελέσματα από τους αλγόριθμούς αναζήτησης. Χρειαζόμασταν μια τέτοια priority queue για την επιστροφή μοναδικών γειτόνων μεγέθους N, ταξινομημένοι με βάση την απόσταση των γειτόνων.

V) **core_utils.cpp + core_utils.h** → Περιλαμβάνει συναρτήσεις που αποτελούν τον κορμό του A ερωτήματος , καλούν KNN και Range Search.

2. **Όσο αναφορά τον HYPERCUBE** , έχουν υλοποιηθεί τα παρακάτω αρχεία κώδικα:

I) **hcube_main.cpp** → Το πηγαίο αρχείο που εμπεριέχει την main. Στο αρχείο αυτό διαχειριζόμαστε τα arguments του command line και καλούμε όλες τις επιμέρους συναρτήσεις που σχετίζονται με

τον συγκεκριμένο αλγόριθμο.

II) **hcube.h** ➔ Στο συγκεκριμένο header file υπάρχουν όλα τα πρωτότυπα των συναρτήσεων(declarations) που έχουν χρησιμοποιηθεί για την υλοποίηση του συγκεκριμένου αλγορίθμου.

III) **hcube.cpp** → Στο συγκεκριμένο αρχείο έχει γίνει το definition όλων των συναρτήσεων που αφορούν τον υπερκύβο.
Όπως και για την υλοποίηση του LSH , έτσι και στον υπερκύβο , χρησιμοποιούνται βοηθητικά τα αρχεία κώδικα που προαναφέρθηκαν (point.cpp , utils.cpp , myHashTable.cpp , PQUnique.cpp).

3. **Όσο αναφορά το κομμάτι του clustering** , έχουν υλοποιηθεί τα παρακάτω αρχεία κώδικα:

I) **confs.cpp + confs.h** → Διαχειρίζεται τις παραμέτρους του cluster.conf (όπως έχει οριστεί και από την εκφώνηση της εργασίας).

II) **cluster_main.cpp** → Το πηγαίο αρχείο που εμπεριέχει την main. Στο αρχείο αυτό διαχειριζόμαστε τα arguments του command line και καλούμε όλες τις επιμέρους συναρτήσεις που σχετίζονται με το Clustering.

III) **cluster.h** → Στο συγκεκριμένο header file υπάρχουν όλα τα πρωτότυπα των συναρτήσεων(declarations) που έχουν χρησιμοποιηθεί για την υλοποίηση της βασικής συσταδοποίησης των διανυσμάτων.

ΣΗΜΕΙΩΣΗ: Η παραπάνω κλάση, περιλαμβάνει την virtual συνάρτηση assignCentroids() η οποία γίνεται overload από τις παραγόμενες υποκλάσεις cluster Lloyd & cluster Reverse. Έτσι, επιτυγχάνεται η σωστή ανάθεση κεντροειδών στις συστάδες.

IV) **cluster.cpp** → Στο συγκεκριμένο αρχείο έχει γίνει το definition όλων των συναρτήσεων που αφορούν το clustering.

V) **clusterLloyd.cpp + clusterLloyd.h** → Εμπεριέχει την συνάρτηση

assignCentroids() η οποία κάνει overload την virtual συνάρτηση της κλάσης Cluster.

VI) **clusterReverse.cpp + clusterReverse.h** → Εμπεριέχει την συνάρτηση assignCentroids() η οποία κάνει overload την virtual συνάρτηση της κλάσης Cluster.Επίσης εμπεριέχει τον δείκτη KNN *method ο οποίος αποτυπώνει την μέθοδο για reverse clustering (LSH ή HCUBE).

Οδηγίες Μεταγλώττισης και Εκτέλεσης

Το πρόγραμμα μεταγλωττίζεται με την εντολή **make**.

Τα 3 εκτελέσιμα υπάρχουν εντός του φακέλου ***bin*** και εκτελούνται σύμφωνα με τις προδιαγραφές της εκφώνησης.

Εργαστήκαμε στα παρακάτω git repo(είναι private):

➔ https://github.com/kondim23/project_1_NN_Clustering/tree/point-abstraction