Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα - Εργασία 2

Δημητρακόπουλος Κωνσταντίνος	ΑΜ: 1115201500034
Κακαβάς Αντώνης	ΑΜ: 1115201500052

Δομή καταλόγων

Το σύνολο των αρχείων που απαρτίζουν την εργασία είναι κατηγοριοποιημένο στους εξής καταλόγους:

- bin:

  Περιέχει τα παραγόμενα εκτελέσιμα αρχεία.

- obj:

  Περιέχει τα παραγόμενα αντικείμενα αρχεία.

- include:

  Περιέχει τα πηγαία αρχεία επικεφαλίδων.

- src:

  Περιέχει τα πηγαία αρχεία κώδικα.

- test:

  Περιέχει τα πηγαία αρχεία κώδικα και επικεφαλίδων που είναι υπεύθυνα για την εκτέλεση των unit tests.

Μεταγλώττιση και Εκτέλεση

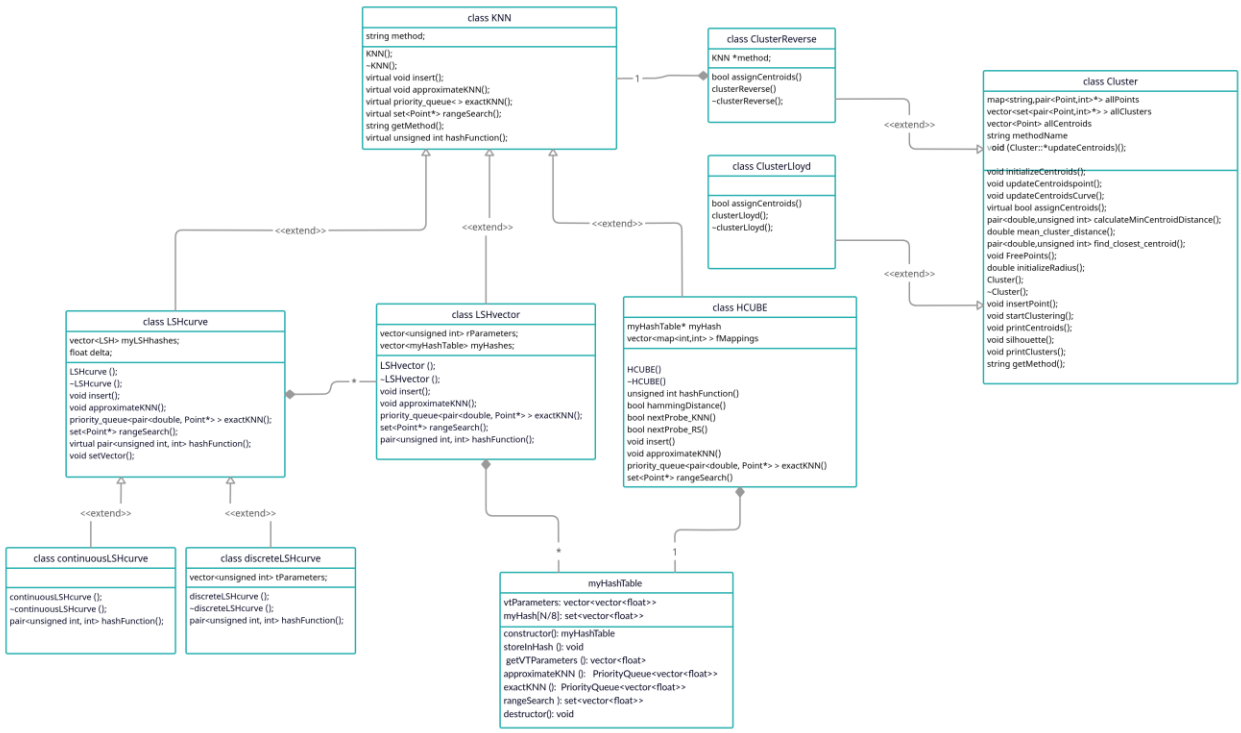
Με την εντολή make πραγματοποιείται μεταγλώττιση και σύνδεση των αρχείων πηγαίου κώδικα. Τελικά παράγονται τρία εκτελέσιμα αρχεία search, cluster, test εντός του καταλόγου bin.

Τα παραπάνω εκτελούνται σύμφωνα με τις προδιαγραφές της εκφώνησης:

- $./bin/search –i <input file> –q <query file> –k <int> -L <int> -M <int> -probes <int> -o <output file> -algorithm <LSH or Hypercube or Frechet> -metric <discrete or continuous | only for –algorithm Frechet> -delta <double>
- $./bin/cluster –i <input file> –c <configuration file> -o <output file> -update <Frechet or Vector> –assignment <Classic or LSH or Hypercube or LSH_Frechet> -complete <optional> -silhouette <optional>
- $./bin/test

Σχεδίαση του Λογισμικού

Το παρακάτω UML Class Diagram παρουσιάζει τη σχεδίαση της εργασίας:



Παρατηρήσεις σχετικά με την σχεδίαση:

- 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.

Κατάλογος Αρχείων

- **Αρχεία point.cpp, sequence.cpp, curve.cpp, point.h, sequence.h, curve.h**:
  - Αφορούν την αναπαράσταση του input.
  - Χρησιμοποιείται η κλάση **Sequence** η οποία κληρονομείται από τις κλάσεις **Curve** και **Point**. Μέσω πολυμορφισμού καλώντας τη συνάρτηση get_distance μεταξύ δύο Sequences καλείται κατάλληλα η l-norm απόσταση για points (vectors) και η discrete/continuous frechet για curves, ανάλογα με τη μορφή του προβλήματος.
  - Η κλάση **Curve** περιέχει κατάλληλες μεθόδους για τη διαχείρηση καμπυλών όπως ανάκτηση συνεχούς-διακριτής απόστασης frechet, της μέσης καμπύλης δύο καμπυλών, φιλτραρίσματος μιας καμπύλης με επαναλαμβανόμενα αυξανόμενο όριο λάθους μέχρι το complexity της καμπύλης να ικανοποιεί συγκεκριμένα κριτήρια.
- **Αρχεία lsh_curve.cpp, lsh_curve.h, lsh_discrete.cpp, lsh_discrete.h, lsh_continuous.cpp, lsh_continuous.h:**
  - Αφορούν τη λογική που εφαρμόζεται στους αλγορίθμους lsh discrete και lsh continuous για καμπύλες.
  - Χρήση πολυμορφισμού μεταξύ των κλάσεων discreteLSHcurve και continuousLSHcurve όπου κληρονομούν τη κλάση LSHcurve. Με κλήση της συνάρτησης hashfunction ενός LSHcurve (ή KNN) θα πραγματοποιηθεί κατάλληλα είτε lsh continuous hashing είτε discrete lsh hashing.

- **Αρχεία lsh_vector.cpp, lsh_vector.h:**
    - Αφορούν τη λογική που εφαρμόζεται στους αλγορίθμους lsh για διανύσματα.

- **Αρχεία hcube.cpp, hcube.h:**
    - Αφορούν τη λογική που εφαρμόζεται στους αλγορίθμους Hypercube για διανύσματα.

Η κλάση LSHvector, LSHcurve και HCUBE κληρονομούν τη κλάση KNN και κάνουν χρήση πολυμορφισμού τόσο κατά τη χρήση τους από την απευθείας κλήση τους, τόσο από την χρήση τους στο clusterReverse.

- **Αρχεία myhashTable.cpp, myHashTable.h:**
    - Χρησιμοποιείται για αποθήκευση των ακολουθιών απο τους αλγορίθμους **LSHvector** και **HCUBE**.

- **Αρχεία cluster.cpp, cluster.h, clusterLloyd.cpp, clusterLloyd.h, clusterReverse.cpp, clusterReverse.h:**
    - Αφορούν τη λογική του αλγορίθμου για clustering διανυσμάτων και καμπυλών.
    - Η κλάση clusterLloyd κληρονομεί κατάλληλα τη κλάση cluster και χρησιμοποιώντας πολυμορφισμό ορίζει την ανάθεση ακολουθιών Lloyd.
    - Χρησιμοποιώντας κλάσεις Sequence ως δεδομένα και πολυμορφισμό στην ίδια κλάση υπολογίζει τις αποστάσεις κατάλληλα (discrete/continuous frechet για καμπύλες και l-norm για διανύσματα), όπως αναφέρεται παραπάνω.
    - Η κλάση clusterReverse κληρονομεί επίσης τη κλάση cluster χρησιμοποιώντας κατάλληλα τη προκαθορισμένη από το χρήστη μέθοδο αναζήτησης (LSHvector – HCUBE - discreteLSHcurve).

- **Αρχεία utils.cpp, utils.h, PQUnique.h, PQUnique.t.hpp:**
    - Αφορούν αρχεία για τη διαχείριση διανυσμάτων, ανάγνωση καμπυλών και διανυσμάτων από την είσοδο, καθώς και μία υλοποίηση ουράς προτεραιότητας με μοναδικά στοιχεία και μεγίστου μεγέθους.
- **Αρχεία search.cpp, cluster.cpp:**
    - Αποτελούν τα αρχεία ανάγνωσης των δεδομένων και οργάνωσης και κλήσης των συναρτήσεων των κλάσεων της εργασίας.
- **Αρχεία main_test.cpp, curve_test.cpp, curve_test.h, cluster_test.cpp, cluster_test.h, lsh_discrete_test.cpp, lsh_discrete_test.cpp:**
    - Αποτελούν τα αρχεία πραγματοποίησης των unit tests. Η υλοποίηση βασίστηκε στη βιβλιοθήκη cppunit. Για την εκμάθηση και παραγωγή των τελικών tests αντλήσαμε γνώση και αξιοποιήσαμε κώδικα από τον ιστοχώρο SourceForge
    - Curve_test: Κατάλληλα εισαγωγή ενός διανύσματος από points για αναπαράσταση ως curve.
    - Lsh_discrete_test: Κατάλληλη εισαγωγή ενός curve στο σύστημα discreteLSHcurve.
    - Cluster_test: Κατάλληλη εισαγωγή ενός curve στο σύστημα Cluster.