

Κ23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα Χειμερινό εξάμηνο 2016-17

3^η Προγραμματιστική Εργασία

Ομάδα:

- Φελιμέγκα Αγγελική 1115201300192 – sdi1300192@di.uoa.gr
- Βούρου Παγώνα 1115201300254 – sdi1300254@di.uoa.gr

Μέρος 2^ο - ConformationA – ConformationB

Τίτλος και περιγραφή του προγράμματος :

Το πρόγραμμα ονομάζεται “proj-phase3-part2” και έχει υλοποιηθεί σε γλώσσα c++.

ConformationA: Το πρόγραμμα διαβάζει διαμορφώσεις που αποτελούνται από N γραμμές, σε μορφή ευκλείδιων διανυσμάτων, από ένα αρχείο και τα τοποθετεί σε clusters με τη βοήθεια της cRMSD , η οποία κατασκευάζεται με την βοήθεια των συναρτήσεων της βιβλιοθήκης Eigen(βιβλιοθήκη γραμμικής άλγεβρας).

ConformationB: Το πρόγραμμα αποθηκεύει τις διαμορφώσεις όπως στο ConformationA και συμπεριφέρεται στα διανύσματα που τις αποτελούν με βάση την ευκλείδια απόσταση τους.

Κατάλογος αρχείων – Περιγραφή Αρχείων :

- main.cpp : Στη main ,στο ConformationA, το πρόγραμμα διαβάζει ένα αρχείο που αποτελείται από numConform διαμορφώσεις, όπου η κάθε μια αποτελείται από N γραμμές. Αποθηκεύουμε τις διαμορφώσεις σε έναν πίνακα, και κλείνουμε το αρχείο. Δημιουργούμε τα cluster των οποίων τα κέντρα τα αρχικοποιούμε με τον αλγόριθμο k-means. Και τα διαμορφώνουμε με βάση τη συνάρτηση cRMSD . Εκτυπώνουμε τα cluster σε ένα αρχείο. Στο ConformationB αποθηκεύουμε τις διαμορφώσεις σε κλάση από vectors. Έπειτα τα αποθηκεύουμε σε clusters ανάλογα την ευκλείδια απόσταση τους. Τα ελέγχουμε με τη συνάρτηση Silhouette.

- Cluster.cpp : Υλοποίηση κλάσης Cluster για την αποτύπωση και αποθήκευση των κέντρων των Cluster αλλά και των ίδιων των Cluster. Επίσης υλοποίηση συναρτήσεων εισαγωγής σημείων, ανανέωσης απόστασης σημείου από κέντρο. Σημείου από σημείο, διαγραφής και εκτύπωσης σημείων (ConformationA + ConformationB)
- Cluster.h : Δήλωση των συναρτήσεων που υλοποιούνται στο αρχείο Cluster.cpp. Ορισμός και υλοποίηση συναρτήσεων καταστροφής, καθώς και της συνάρτησης επιστροφής κέντρου από κάθε Cluster. (ConformationA + ConformationB).
- k_means.h: Δήλωση της συνάρτησης “k_means” για το ConformationA και της “k_meansb” για το ConformationB.
- k_means_.cpp: Υλοποίηση του αλγορίθμου k_means. Δημιουργία και υλοποίηση της συνάρτησης “k_means” για το ConformationA και της “k_meansb” για το ConformationB.
- PAM.h: Δήλωση των συναρτήσεων PAM και PAM_Update για το ConformationA και PAMB και PAM_UpdateB για το ConformationB.
- PAM.cpp: Υλοποίηση του αλγορίθμου PAM για τον διαμερισμό των σημείων στα κέντρα των Clusters. Μέσα στο αρχείο PAM.cpp υπάρχουν 4 συναρτήσεις. Η συνάρτηση PAM και PAMB για τον διαμερισμό των σημείων σε άδεια Clusters ανάλογα με το είδος των σημείων και την απόσταση τους από το κέντρο του κάθε Cluster, και η συνάρτηση PAM_Update και PAM_UpdateB , η οποία μοιράζει τα σημεία σε γεμάτα Clusters τα οποία είτε άλλαξαν κέντρο, είτε όχι. Γίνεται έλεγχος για τα σημεία και ανάλογα γίνεται ο διαμερισμός.

- `alaLloyds.h`: Δήλωση της συνάρτησης `alaLloyds` για το `ConformationA` και `alaLloydsb` για το `ConformationB`.
 - `alaLloyds.cpp`: Υλοποίηση του αλγορίθμου `a La Loyd's` για την ανανέωση των κέντρων των `Clusters` και των σημείων που ανήκει σε κάθε κέντρο. Δημιουργία συνάρτησης `alaLloyds` και `alaLloydsb`. Αφου υπολογιστούν τα καινούρια κέντρα με τον αλγόριθμο `a La Loyd's` καλούμε την `PAM_Update` ή την `PAM_UpdateB`(για το δεύτερο μέρος) για να μοιραστούνε τα σημεία στα καινούρια κέντρα(εάν δεν υπάρχουν ήδη).
-
- `Silhouette.h` : Δήλωση της συνάρτησης `Silhouette`. (`ConformationB`)
 - `Silhoutte.cpp` : Υλοποίηση του αλγορίθμου `Silhouette`. Δημιουργία συνάρτησης `Silhouette`. Η συνάρτηση `Silhouette` δίνει αριθμούς απο το -1 μέχρι το 1, για αξιολόγηση των αλγορίθμων που προηγήθηκαν. (`ConformationB`)
-
- `Distances.h` : Δήλωση των συναρτήσεων `DistanceHamming`, `DistanceCosine`, `DistanceEuclidean`, `MinDistance`. (`ConformationA` + `ConformationB`).
 - `Distances.cpp`: Το αρχείο αυτό περιέχει την υλοποίηση των συναρτήσεων `DistanceHamming`, `DistanceCosine`, `DistanceEuclidean` για τον υπολογισμό του σημείου απο σημείο . είτε απο κέντρο και την υλοποίηση της συνάρτησης `MinDistance` για τον υπολογισμό της μικρότερης απόστασης. (`ConformationA` + `ConformationB`).
-
- `cRMSD.h` : Δήλωση της συνάρτησης `CRMSD`. (`ConformationA`).
 - `cRMSD.cpp`: Υλοποίηση της συνάρτησης `CRMSD` με βάση τον αλγόριθμο `CRMSD`. Η υλοποίηση έγινε με τη βοήθεια της βιβλιοθήκης γραμμικής άλγεβρας, `Eigen`. (`ConformationA`).

- Conformations.h : Δήλωση της κλάσης Conformations.
(ConformationB).
- Conformations.cpp : Υλοποίηση της κλάσης Conformations για τη διαχείριση των διαμορφώσεων στο δεύτερο μέρος.
(ConformationB).
- CosineSim.h : Δήλωση/αρχικοποίηση της κλάσης CosineSim και των μεθόδων της.
- CosineSim.cpp : Υλοποίηση της κλάσης CosineSim και των μεθόδων της. Η CosineSim λειτουργεί με βάση την ομοιότητα συνιμητόνου. Περιέχει τους constructors, και τον destructor της κλάσης καθώς και την μέθοδο ConstructGFunctionC(int k), για την δημιουργία της συνάρτησης g. Μέσα σε αυτή τη συνάρτηση υπολογίζεται το εσωτερικό γινόμενο ενός τυχαίου αριθμού με τις συντεταγμένες του αντικειμένου και ανάλογα ,αν το γινόμενο είναι θετικό ή αρνητικό δημιουργούμε μία συμβολοσειρά από παράθεση τιμών 0 και 1, η οποία και επιστρέφεται.
- DistanceMatrix.h : Υλοποίηση της κλάσης DistanceMatrix και των μεθόδων της. Περιέχει τους constructors, και τον destructor της κλάσης καθώς και την μέθοδο ConstructGFunctionC(int item,int k). Στον constructor της κλάσης αυτής διαβάζεται το αρχείο και εισάγονται τα στοιχεία σε έναν πίνακα. Έπειτα στην συνάρτηση κατασκευής της g, με τις κατάλληλες μαθηματικές πράξεις δημιουργείται το αποτέλεσμα της g (παράθεση τιμών 0 και 1) και επιστρέφεται.
- DistanceMatrix.cpp : Δήλωση/αρχικοποίηση της κλάσης DistanceMatrix και των μεθόδων της.
- Euclidean.h : Δήλωση/αρχικοποίηση της κλάσης Euclidean και των μεθόδων της.
- Euclidean.cpp : Υλοποίηση της κλάσης Euclidean και των μεθόδων της.

της. Περιέχει τους constructors, και τον destructor της κλάσης καθώς και την μέθοδο ConstructFiFunctionC(int item,int k), όπου κατασκευάζεται η fi. Μέσα στη συνάρτηση αυτή, αφού αποθηκευτούν οι διαστάσεις του σημείου σε έναν πίνακα, και μετά απο συγκεκριμένες μαθηματικές πράξεις επιστρέφεται το αποτέλεσμα της fi, το οποίο είναι ένας ακέραιος αριθμός.

- Hamming.h : Δήλωση/αρχικοποίηση της κλάσης Hamming και των μεθόδων της.
- Hamming.cpp : Υλοποίηση της κλάσης Hamming και των μεθόδων της. Περιέχει τους constructors, και τον destructor της κλάσης καθώς και την μέθοδο ConstructGFunctionC(int item,int k), όπου κατασκευάζεται η g (συμβολοσειρά από παράθεση τιμών 0 και 1) και επιστρέφεται.

- randomfunc.h : Δήλωση/ αρχικοποίηση των συναρτήσεων marsagliarandom(int j) και mod (int a, int b).
- randomfunc.cpp : Υλοποίηση των συναρτήσεων marsagliarandom(int j) και mod (int a, int b).

marsagliarandom(int j) : Γεννήτρια τυχαίων μεταβλητών με βάση τ την μέθοδο marsaglian.

mod (int a, int b): Συνάρτηση υπολογισμού υπολοίπου.

- makefile

Οδηγίες Χρήσης του Προγράμματος :

Για τη μεταγλώττιση : **make all**

Για την εκτέλεση : *./conformations -a<input file> -o<outputfile>*

