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

3η Προγραμματιστική Εργασία

Ομάδα:

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

Μέρος 1ο - RecommendationA - RecommendationB

Τίτλος και περιγραφή του προγράμματος :

Το πρόγραμμα ονομάζεται “proj-phase3-part1” και έχει υλοποιηθεί σε γλώσσα c++.

RecommendationA : Το πρόγραμμα διαβάζει σημεία – δεδομένα από ένα αρχείο και τα βάζει σε hash table (τα οποία έχουν δημιουργηθεί στην αρχλη του προγράμματος) ανάλογα με τη μέθοδο που επιλέγει ο χρήστης, Hamming, Euclidean, Cosine Similarity. Τα δεδομένα που διαβάζει το πρόγραμμα είναι βαθμολογίες χρηστών, για αντικειμενα που έχει δει. Σκοπός του προγράμματος ειναι να συστήνει στον χρήστη «γειτονικά» αντικείμενα με αυτά που έχει δει, με καλή βαθμολογία.

RecommendationB : Το πρόγραμμα κάνει το ίδιο πράγμα με το RecommendationA. Με τη διαφορά του ότι αντί για hash table έχουμε clusters. Τα δεδομένα αποθηκεύονται σε clusters.

Κατάλογος αρχείων – Περιγραφή Αρχείων :

- main.cpp : Στη main το πρόγραμμα διαβάζει ένα αρχείο. Υπάρχει ένα μενού επιλογης από τον χρήστη, του προγράμματος που θέλει να υλοποιήσει., είτε το RecommendationA είτε το RecommendationB, και έπειτα ποια μέθοδο θέλει να χρησιμοποιηθεί για αυτά, Hamming, Euclidean, Cosine Similarity.Αφού γίνει αυτό γίνεται κανονικοποίηση των βαθμολογιών με τη βοήθεια της συνάρτησης Rate που βρίσκεται μέσα στη main. Έπειτα βάζουμε τα δεδομένα σε hash tables και με την Query Search βρίσκουμε τους κοντινότερους

χρήστες(γείτονες). Τέλος με την NNRMDA ελέγχουμε ποια αντικείμενα δεν έχουν βαθμολογηθεί απο κάθε χρήστη και τα βαθμολογούμε ανάλογα με την βαθμολογία που έχουν δώσει οι γείτονες του χρήστη. Η main χρειάζεται και για το RecommendationA, και για το RecommendationB.

- Hashtable.h : Αυτό το αρχείο επικεφαλίδας περιέχει την αρχικοποίηση και τον ορισμό του πίνακα κατακερματισμού. Ο πίνακας κατακερματισμού αποτελείται απο αντικείμενα της Συνδεδεμένης Λίστας. (RecommendationA)
- Hashtable.cpp : Σε αυτο το αρχείο εκτός από τους constructors, τον destructor τον πίνακα, και τη συνάρτηση εκτύπωσης του, printTable(), υλοποιούνται και οι δύο άλλες συναρτήσεις, η InsetIntoHashtable(),και η SearchBucket . Η InsetIntoHashtable() ανάλογα με τη μέθοδο και τη g, εισάγει τα στοιχεία στο κατάλληλο bucket. Έχει γίνει διαχωρισμός στον τρόπο εισαγωγής των στοιχείων απο Hamming, Cosine (και DistanceMatrix) απο την Euclidean επιστρέφει την συνάρτηση fi που αποτελείται απο τις g. Η SearchBucket() ανάλογα με τη μέθοδο, Hamming , Cosine και Euclidean καλεί τις κατάλληλες συναρτήσεις για τον εντοπισμό των κοντινότερων γειτόνων ακτίνας και τον κοντινότερο γείτονα με την μικρότερη απόσταση. (RecommendationA)

- LinkedListh.h : Αυτο το αρχείο επικεφαλίδας περιέχει την αρχικοποίηση και τον ορισμό της συνδεδεμένης λίστας που χρησιμοποιείται για την υλοποίηση του πίνακα κατακερματισμού. (RecommendationA)
- LinkedList.cpp : Σε αυτό το αρχείο εκτός από τους constructors, τον destructor της λίστας και την συνάρτηση εκτύπωσης της ,printList(), υλοποιούνται και δύο άλλες συναρτήσεις, η Search() και η NN_Search(). Η Search ανάλογα με τη μέθοδο που το γράφει το αρχείο, και που της έχει στείλει η SearchBucket του πίνακα κατακερματισμού, ψάχνει και εκτυπώνει τους κοντινότερους γείτονες ακτίνας R. Η κάθε μέθοδος, ανάλογα με τον ορισμό της, έχει και διαφορετικο υπολογισμό των κοντινότερων γειτόνων. Η NN_Search(), αντίστοιχα, ανάλογα με τη μέθοδο που το γράφει το αρχείο, και που της έχει στείλει η SearchBucket του πίνακα κατακερματισμού, ψάχνει και επιστρέφει τον (έναν) κοντινότερο

γείτονα, αυτόν με τη μικρότερη απόσταση. Η κάθε μέθοδος, ανάλογα με τον ορισμό της, έχει και διαφορετικο υπολογισμό του κοντινότερου γείτονα.(RecommendationA)

- 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 για κάθε είδος σημείου ξεχωριστά, ανανεώσης απόστασης σημείου απο κέντρο. Σημείου απο σημείο, διαγραφής σημείου και εκτύπωση σημείων για κάθε είδος σημείων ξεχωριστά. (RecommendationB)
- Cluster.h : Δήλωση των συναρτήσεων που υλοποιούνται στο αρχείο Cluster.cpp. Ορισμός και υλοποίηση συναρτήσεων καταστροφής, καθώς και της συνάρτησης επιστροφής κέντρου απο κάθε Cluster ανάλογα με το είδος των σημείων. (RecommendationB)

- DataPoints.h : Δήλωση των συναρτήσεων CreateHammingPoints, CreateEuvlideanPoints, CreateCosinePoints και CreateMatrixPoints.(RecommendationA + RecommendationB)

- DataPoints.cpp: Το αρχείο DataPoints.cpp περιέχει τις υλοποιήσεις των συναρτήσεων : - CreateHammingPoints που πέρνει τα σημεία απο το αρχείο Hamming(αν έχουμε) και τα βάζει σε πίνακες hamming, - CreateEuclideanPoints που πέρνει τα σημεία απο το αρχείο Euclidean (αν έχουμε) και τα βάζει σε πίνακες euclidean, - CreateCosinePoints που πέρνει τα σημεία απο το αρχείο Cosine (αν έχουμε) και τα βάζει σε πίνακες cosine, - CreateMatrixPoints που πέρνει τα σημεία απο το αρχείο DistanceMatrix (αν έχουμε) και τα βάζει σε πίνακες diastancematrix.
  (RecommendationA+RecommendationB)

- K_medoids.h: Δήλωση της συνάρτησης “k_medoidspp”.
  (RecommendationB)

- K_medoids.cpp: Υλοποίηση του αλγορίθμου k-medoids++. Δημιουργία και υλοποίηση της συνάρτησης "k_medoidspp". Ανάλογα με το είδος των σημείων, αρχικοποιείται τυχαία το πρώτο κέντρο και έπειτα σύμφωνα με τον αλγόριθμο και τις αποστάσεις, γίνεται δημιουργία και των υπόλοιπων κέντρων. (RecommendationB)

- LSH.h: Δήλωση της συνάρτησης LSH. (RecommendationB)
- LSH.cpp: Υλοποίηση του αλγορίθμου LSH για διαμερισμό των σημείων στα κέντρα των Cluster. Μέσα στο αρχείο LSH.cpp γίνται η αρχικοποίηση των Hashtables, που τα έχουμε εισάγει στα κέντρα των Clusters και η υλοποίση της συνάρτησης QuerySearch που βρίσκει τα κοντινότερα σημεία με βάση την ακτίνα –απόσταση τους από τα κέντρα. Στη συνάρτηση LSH περνάμε μια μεταβλητή choice. (RecommendationB)

- NN_recommendation.h: : Δήλωση της συνάρτησης NNRMD. (RecommendationA+RecommendationB)

- NN_recommendation.cpp: Υλοποίηση της συνάρτησης NNRMD. Ανάλογα με την μέθοδο που έχει επιλεχθεί να χρησιμοποιηθεί για την κανονικοποίηση και την αποθήκευση στα Hashtables, περνάμε τα σημεία απο τη main στη συνάρτηση NNRMD. Μέσα στη συνάρτηση γίνεται εύρεση των αντικειμένων που δεν εχουν βαθμολογηθέι για κάθε χρήστη και αν οι γειτονές του τα έχουν βαθμολογήσει βρίσκουμε μια καινούρια βαθμολόγηση για αυτά με βάση την βαθμολογία που τους έχει δωθεί απο τους άλλους. Επίσης γίνεται ταξινόμηση και αποθήκευση σε πίνακα των 5 καλύτερων – με βάση την βαθμολογία- αντικειμένων. (RecommendationA+RecommendationB)

- PAM.h: Δήλωση των συναρτήσεων PAM και PAM_Update. (RecommendationB)

- PAM.cpp: Υλοποίηση του αλγορίθμου PAM για τον διαμερισμό των σημείων στα κέντρα των Clusters. Μέσα στο αρχείο PAM.cpp υπάρχουν δύο συναρτήσεις. Η συνάρτηση PAM 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, η οποία μοιράζει τα σημεία σε γεμάτα Clusters τα οποία είτε άλλαξαν κέντρο, είτε όχι. Γίνεται έλεγχος για τα σημεία και ανάλογα γίνεται ο διαμέρισμος. (RecommendationB)

- alaLloyds.h: Δήλωση της συνάρτησης alaLloyds. (RecommendationB)
- 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. Αφου υπολογιστούν τα καινούρια κέντρα με τον αλγόριθμο a La Loyd's καλούμε την PAM_Update για να μοιραστούμε τα σημεία στα καινούρια κέντρα(εάν δεν υπάρχουν ήδη). (RecommendationB)
- 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. (RecommendationA+RecommendationB)
- Silhouette.h : Δήλωση της συνάρτησης Silhouette. (RecommendationA+RecommendationB)

- CosineSim.h : Δήλωση/αρχικοποίηση της κλάσης CosineSim και των μεθόδων της. (RecommendationA+RecommendationB)
- 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. (RecommendationA+RecommendationB)

- DistanceMatrix.h : Δήλωση/αρχικοποίηση της κλάσης DistanceMatrix και των μεθόδων της. (RecommendationA+RecommendationB)

- DistanceMatrix.cpp : Υλοποίηση της κλάση 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 εισάγωνται τα στοιχεία σε έναν πίνακα. Έπειτα στην συνάρτηση κατασκευής της g, με τις κατάλληλες μαθηματικές πράξεις δημιουργείται το αποτέλεσμα της g (παράθεση τιμών 0 και1) και επιστρέφεται. (RecommendationA+RecommendationB)

- Euclidean.h : Δήλωση/αρχικοποίηση της κλάσης Euclidean και των μεθόδων της. (RecommendationA+RecommendationB)
- Euclidean.cpp : Υλοποίηση της κλάσης Euclidean και των μεθόδων 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, αφου 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. (RecommendationA+RecommendationB)

- Hamming.h : Δήλωση/αρχικοποίηση της κλάσης Hamming και των μεθόδων της. (RecommendationA+RecommendationB)
- 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. (RecommendationA+RecommendationB)

- NeighbourSearch.h : Δήλωση/ αρχικοποίηση των συναρτήσεων αναζήτησης του κοντινότερου γείτονα, RangeNeighbourSearch και Nearest_Neighbor_Search. (RecommendationA+RecommendationB)
- NeighbourSearch.cpp : Υλοποίηση των συναρτήσεων RangeNeighbourSearch και Nearest_Neighbor_Search.
  RangeNeighbourSearch : Ανάλογα με τη μέθοδο, καλείται η κατασκευάστρια της g, της οποίας το αποτέλεσμα, μαζί με την ακτίνα που διαβάζεται απο το αρχείο, στέλνεται σαν όρισμα στη Searchbucket για την έυρεση των κοντινότερων γειτόνων ακτίνας R. (RecommendationA+RecommendationB)

- randomfunc.h : Δήλωση/ αρχικοποίηση των συναρτήσεων marsagliarandom(int j) και mod (int a, int b). (RecommendationA+RecommendationB)
- randomfunc.cpp : Υλοποίηση των συναρτήσεων marsagliarandom(int j) και mod (int a, int b).
  marsagliarandom(int j) : Γεννήτρια τυχαίων μεταβλητών με βάση την μέθοδο marsaglian.
  mod (int a, int b): Συνάρτηση υπολογισμού υπολοίπου.
  (RecommendationA+RecommendationB)

- makefile (RecommendationA+RecommendationB)

Οδηγίες Χρήσης του Προγράμματος :

Για τη μεταγλώττιση : **make all**

Για την εκτέλεση : *$./recommendation –d<input file> -o<output file>*