

# ΑΝΑΦΟΡΑ ΣΧΕΔΙΑΣΤΙΚΩΝ ΕΠΙΛΟΓΩΝ

## (Part2)

(ΜΙΧΟΠΟΥΛΟΣ ΣΟΦΟΚΛΗΣ - ΙΩΑΝΝΗΣ, ΦΡΑΝΤΖΗΣ ΑΝΔΡΕΑΣ-ΣΥΜΕΩΝ)

### Σχεδιαστικές επιλογές FilteredVamana:

**FilteredGreedySearch(map<float, float> S, vector<float>& Xq, int k, int L, vector<int>& Fq);**

Για να υλοποιηθεί η συνάρτηση FilteredGreedySearch όπως αναφέρεται και στην εκφώνηση χρειάστηκε να δωθούν στην συνάρτηση τα εξής ορίσματα:

- a) int S: Κόμβος S του γράφου
- b) vector
- c) int k: k γείτονες που θα βρούμε
- d) int L: Μέγεθος λίστας γειτόνων που ψάχνουμε
- e) vector

Η συνάρτηση διατηρεί την ίδια λογική με αυτή του πρώτου παραδοτέου μόνο που αυτήν την φορά έπρεπε να βρεθούν αυτοί οι κοντινότεροι γείτονες που έχουν το ίδιο φίλτρο.

\*\*Σύμφωνα με τα δεδομένα που είχαμε για να πειραματιστούμε σχεδιάσαμε τον αλγόριθμο FilteredGreedySearch έτσι ώστε να είναι αποτελεσματικός και σε αυτά τα queries που δεν είχαν κάποιο φίλτρο.

**void FilteredRobustPrune(int p, unordered\_set<int>& Visited, int a, int R);**

Για να υλοποιηθεί η συνάρτηση FilteredRobustPrune όπως αναφέρεται στην εκφώνηση χρειάστηκε να δωθούν στην συνάρτηση τα εξής ορίσματα:

- a) int p: Ο κόμβος ή σημείο του γράφου που θα τροποποιηθεί
- b) unordered\_set
- c) int a: Παράγοντας απόστασης
- d) int R: Ανώτατο όριο νέων γειτόνων

Ο ρόλος της συνάρτησης είναι να επιλέγει και να φιλτράρει την έρεση και την προσθήκη γειτόνων του κόμβου P, βασισμένο σε αποστάσεις και φίλτρα. Η συνάρτηση σταματάει την διαδικασία όταν φτάσει το όριο R γειτόνων ή όταν το visited αδειάσει.

Παράλληλα με τον προγραμματισμό αυτών των δύο βασικών συναρτήσεων, υλοποιήθηκαν και άλλες δύο βασικές συναρτήσεις στο αρχείο Functions.h:

- Sint findFilteredMin(unordered\_set<int>& DiffResult, map<int, vector<float>>& coordinates, vector<float>& Xq);  
Συνάρτηση που επιστρέφει το id του κόμβου που έχει μικρότερη απόσταση από το Xq
- Smap<float, float> findFiltersMedoid(vector<vector<float>>& P, int threshold, set<float>& filters);  
Συνάρτηση για τον υπολογισμό του FilteredMedoid.

Το αρχείο FilteredVamana.cpp περιέχει όλες τις υλοποιήσεις συναρτήσεων των αρχείων Graph.h και Functions.h και αποτελεί το τελικό αρχείο του δεύτερου παραδοτέου. Η εκτέλεση του αρχείου ξεκινάει με το διάβασμα των δεδομένων το οποίο έγινε με την βοήθεια των συναρτήσεων του αρχείου io.h. Στην αρχή της εκτέλεσης γίνεται η αποθήκευση και η επεξεργασία (cords, filters) των δεδομένων σε vectors που μας παρέχουν πρόσβαση στα δεδομένα της εργασίας. Για τις ανάγκες τις εργασίας διαμορφώσαμε τα δεδομένα έτσι ώστε στην πρώτη θέση να εμπεριέχεται το φίλτρο και στις δύο θέσεις που υπήρχαν timestamps βάλαμε τον αριθμό -1. Στην συνέχεια αρχικοποιείται ένας τυχαίος γράφος του οποίου ο κάθε κόμβος θα έχει R ακμές και ένα vector N θέσεων το οποίο θα περιέχει τιμές από το 0 μέχρι το n-1 αντίστοιχα με τους κόμβους. Έπειτα υπολογίζεται το medoidMap του αρχείου για κάθε φίλτρο που υπάρχει και ξεκινάει η διαδικασία του αλγορίθμου FilteredVamana όπως ακριβώς φαίνεται και στην εκφώνηση της εργασίας. Μετά την ολοκλήρωση της επαναληπτικής διαδικασίας που σημαίνει το τέλος του αλγορίθμου FilteredVamana καλείται η FilteredGreedySearch για κάθε query του αρχείου query και ελέγχεται το αποτέλεσμα με βάση το αρχείο ground truth υπολογίζοντας ένα ποσοστό επιτυχίας.

## Σχεδιαστικές επιλογές StitchedVamana:

Στην υλοποίηση του StitchedVamana δεν χρειάστηκε να υλοποιήσουμε κάποια συνάρτηση, γιατί όπως φαίνεται και στην εκφώνηση, ο StitchedVamana χρησιμοποιεί συναρτήσεις που έχουμε ήδη υλοποιήσει. Κατά την εκτέλεση του, για κάθε φίλτρο δημιουργείται ένας γράφος μέσω του αλγόριθμου Vamana και στην συνέχεια υπολογίζεται το ποσοστό επιτυχίας κάθε γράφου και το συνολικό ποσοστό επιτυχίας όλων των γράφων.

## Προσθήκες στο αρχείο Graph.h

Στο αρχείο Graph.h υλοποιήθηκαν επίσης οι συναρτήσεις SaveGraphBinary(const string& filename) και Graph LoadGraphBinary(const string& filename) ώστε να μπορούμε να αποθηκεύουμε τους γράφους σε binary μορφή και να μπορούμε να τους φορτώνουμε στη μνήμη.

- **SaveGraphBinary(const string& filename)** →  
Συνάρτηση που αποθηκεύει τον γράφο σε binary μορφή void
- **Graph LoadGraphBinary(const string& filename)**  
Συνάρτηση που φορτώνει τον γράφο από το binary αρχείο και τον επιστρέφει.