

Σχεδιαστικές Επιλογές

Graph

Επιλέξαμε να φτιάξουμε δική μας κλάση γράφου με template για να δέχεται δεδομένα κάθε μορφής η οποία περιέχει:

- Ένα `std::set` με τα vertices

Τα duplicates αγνοούνται.

- Ένα vector απο vectors για adjacency list

Το πρώτο στοιχείο σε κάθε vector είναι το vertex το ίδιο και όχι γείτονας του.

- Ένα map απο int σε vertices για indexing (ξεκινώντας απο το 0)

Τα index αντιστοιχίζονται στο αντίστοιχο vertex στην adjacency list.

- Είναι Directed by default αλλά αυτό μπορεί να αλλάξει μέσω του ορίσματος `isDirected`

Για undirected κάθε undirected ακμή μετράει ως 2 στο count.

Utilities

- Header file με βοηθητικές συναρτήσεις για την υλοποίηση και λειτουργία του project καθώς και για την δημιουργία unit test.

- Φτιάξαμε δική μας Euclidean distance με template για να λειτουργεί για int και float.

- Δημιουργήσαμε μια κλάση TestGraph μέσω της οποίας ο χρήστης μπορεί να έχει πρόσβαση στο adjacency list του γράφου για να να χρησιμοποιηθούν στα unit tests.

- Η εισαγωγή των δεδομένων στο σύστημα γίνεται μέσω μιας συνάρτησης η οποία δέχεται ένα .fvec αρχείο και έναν άδειο γράφο και γεμίζει τον γράφο με τα δεδομένα του αρχείου.

- Στην πορεία δημιουργήθηκαν και αντίστοιχες συναρτήσεις για sets και vectors οι οποίες υποστηρίζουν και ivec αρχεία.

KNN

- Αντί για dataset P υλοποιούμε κατευθείαν έναν γράφο με P στοιχεία και μετά προσθέτουμε κόμβους

- Το σύνολο L στην GreedySearch περιέχει τον κόμβο Xq αν αυτός προστεθεί σε αυτό αλλά όταν καλείται η `retain_closest_neighbors` αυτή το αγνοεί.

- Όταν 2 κόμβοι απέχουν ίδια απόσταση από κάποιον κόμβο, ως πιο κοντινός επιλέγεται αυτός που εμφανίζεται πρώτος στο σύνολο.
- Ο βαρμάνι επιστρέφει το index του medoid του γράφου ετ

Main

- Το αρχείο εισόδου είναι by default το shiftsmall όπως και τα αντιστοιχα query και groundtruth αρχεία.
- Οι παράμετροι k,L,R και προαιρετικά το a δίνονται ως ορίσματα στην γραμμή εντολών.
 - Το a μπορούμε να το παραβλέψουμε και by default θα είναι 1.2