1η Σειρά Προγραμματισμού Τεχνητής Νοημοσύνης

Ξυπόλιτος Γιώργος 03113629
Πετρίδης Στέφανος 03113418

21-11-2016

1 Στόχος

Στο πλαίσιο αυτής της εργασίας καλούμαστε να υλοποιήσουμε το βασικό κορμό μιας ευφυούς υπηρεσίας εξυπηρέτησης πελατών ταξί. Συγκεκριμένα, θεωρούμε ότι υπάρχει ένας πελάτης που βρίσκεται σε μια ορισμένη τοποθεσία, ο οποίος διαθέτει κινητό τηλέφωνο με GPS και επιθυμεί να καλέσει ένα ταξί. Η υπηρεσία διαθέτει μια βάση δεδομένων με όλα τα διαθέσιμα ταξί και τη γεωγραφική θέση στην οποία βρίσκονται κάθε χρονική στιγμή, η οποία ανανεώνεται συνεχώς. Η υπηρεσία θα πρέπει να εντοπίζει και να ειδοποιεί το ταξί που μπορεί να μεταβεί πιο γρήγορα στη θέση του πελάτη ώστε να τον εξυπηρετήσει.

2 Υλοποίηση

Η υλοποίηση της άσκησης έγινε μέσω της χρήσης τριών αρχείων CSV(nodes, taxis, client), τα οποία περιείχαν τους κόμβους όλων των δρόμων, τις θέσεις των ταξί, οι οποίες δεν ήταν απαραίτητα και θέση σε δρόμο και την θέση του πελάτη για τον οποίο ισχύει ό,τι και για τα ταξί.
Οι θέσεις των ταξί και του πελάτη, για την υλοποίηση, θεωρήθηκαν ως οι κοντινότερες θέσεις προς αυτά.
Το πρόγραμμα αρχικά διαβάζει όλα τα nodes μέσω της κλάσης CSV και τα αποθηκεύει σε μια δομή δεδομένων hashMap με key ένα String και Data μια δικιά μας δομή δεδομένων, την Coords, η οποία είναι μια "τούπλα" δύο float variables.
Οι θέσεις των client και taxis, διαβάζονται επίσης από την κλάσση CSV και το πρώτο επιστρέφεται απευθείας ως Coords, ενώ το δεύτερο ως HashMap με key το Coords του εκάστοτε ταξί και Data το id του.
Παράλληλα σε αυτή την διαδικασία δημιουργεί τον γράφο, με τους κόμβους και τις ακμές, πάνω στον οποίο θα εφαρμοστεί ο αλγόριθμος A* παίρνοντας κατάληλλες συνθήκες για να βρίσκει τις διασταυρώσεις και αποφεύγοντας loops.
Ο A* εφαρμόστηκε για κάθε ταξί, με αρχικό κόμβο τον κόμβο του ταξί και κόμβοστόχο τον πελάτη και από αυτούς επιλέχθηκε αυτός με την μικρότερη απόσταση και αναπαραστάθηκε στο αρχείο KML με πράσινο χρώμα.
Ως ευριστική χρησιμοποιήθηκε η Ευκλείδια απόσταση.
Για την υλοποίηση του A* χρησιμοποιήθηκαν δομές δεδομένων τύπου HashSet όπου κρατούσε τους κόμβους οι οποίοι έχουν ήδη επεκταθεί, Priority Queue για να υλοποιηθεί η priority ώστε να παίρνουμε εύκολα και γρήγορα τον κόμβο προς επέκταση και HashMap ώστε να κρατάμε τους predecessor των στοιχείων και τα τρέχοντα κόστη τους.
Μόλις ο αλγόριθμός μας συναντήσει τον τελικό κόμβο και πάει να τον επεκτείνει, κατλαβαίνει ότι βρίσκεται σε τελική κατάσταση και πηγαίνοντας προς τα πίσω με την βοήθεια της δομής cameFrom βρίσκει την λίστα των κόμβων την οποία προσπέλασε.