Πανεπιστήμιο Αιγαίου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Τεχνητή Νοημοσύνη

1η Εργασία

321/2018215 Σωτηρόπουλος Ανδρέας

321/2018232 Τσουκαλάς Δημήτριος

Απρίλιος 2021

Περιεχόμενα

1 Χρήση του προγράμματος

1.1 Κώδικας

Το πρόγραμμα βρίσκεται στο αρχείο main.py. Για την χρήση των αλγορίθμων αρκεί η εκτέλεση του αρχείου αυτού με του διερμηνέα της Python (δοκιμάστηκε σε cpython 3.8 και 3.9) και η εισαγωγή των απαραίτητων δεδομένων του προγράμματος όταν αυτά ζητηθούν.

1.2 Εκτελέσιμο

Στην περίπτωση που στον υπολογιστή του χρήστη δεν είναι εγκατεστημένη μια συμβατή έκδοση διερμηνέα Python, παρέχεται στο φάκελο dist ένα εκτελέσιμο αρχείο του προγράμματος μαζί με τα απαραίτητα αρχεία του cpython.Το αρχείο αυτό ονομάζεται main.exe

Για την χρήση του εκτελέσιμου πρέπει να χρησιμοποιηθεί cmd, καθώς αν το πρόγραμμα εκτελεστεί κατευθείαν θα κλείσει πριν ο χρήστης προλάβει να δει το αποτέλεσμα.

1.3 Δημιουργία του εκτελέσιμου

Για την δημιουργία του εκτελέσιμου πρέπει να εκτελεστεί το αρχείο setup.py. Απαιτείται η ύπαρξη συμβατού διερμηνέα Python και το πακέτο py2exe.

2 Προβλήματα υλοποίησης

2.1 Πώς θα αναπαραστήσουμε το πρόβλημα;

Το πρώτο πρόβλημα που προέκυψε σχετίζονταν με το πώς θα εφαρμόσουμε τους αλγορίθμους στο πρόβλημα της εργασίας, αλλά και πως θα αναπαραστήσουμε τα δεδομένα του προβλήματος.

Τα δεδομένα του προβλήματος είναι οι άνθρωποι και η ταχύτητά τους, ο φακός και το σημείο στο οποίο βρίσκεται, οι άνθρωποι που έχουν περάσει τη γέφυρα και οι άνθρωποι που έχουν μείνει πίσω. Για την αναπαράσταση των ανθρώπων και της ταχύτητάς τους δημιουργήσαμε μία κλάση (Person) η οποία διατηρεί το όνομα του ανθρώπου (A1) και την ταχύτητά του σε λεπτά. Ο φακός και το ποιος βρίσκεται σε ποια πλευρά της γέφυρας αναπαρίστανται από την κλάση State η οποία περιέχει μία πλειάδα (tuple) με τους ανθρώπους που πρέπει να περάσουν την γέφυρα, αλλά και αν έχουν τον φακό η όχι.

Το ίδιο το πρόβλημα αναπαρίσταται και αυτό με μία κλάση η οποία διατηρεί τον αρχικό αριθμό των ανθρώπων, το αρχικό State και το State στόχο. Επίσης παρέχει μεθόδους εύρεσης των ανθρώπων που έχουν περάσει την γέφυρα, και εύρεσης των επόμενων κόμβων.

Αναπαριστώντας το πρόβλημα με αυτές τις δομές η υλοποίηση των αλγορίθμων και η επίλυση του προβλήματος έγινε πλέον πολύ πιο εύκολη.

2.2 Εύρεση ευρετικών συναρτήσεων

Με από διάφορες δοκιμές καταλήξαμε σε δύο ευρετικές συναρτήσεις. Η πρώτη ευρετική συνάρτηση που χρησιμοποιήσαμε υποθέτει ότι ο φακός δεν χρειάζεται και ότι η γέφυρα δεν έχει κανένα όριο στον αριθμό των ανθρώπων που μπορούν να την διασχίσουν ταυτόχρονα. Έτσι όλοι οι άνθρωποι μπορούν να περάσουν την γέφυρα στον χρόνο του πιο αργού. Η δεύτερη συνάρτηση υποθέτει ότι ο φακός χρειάζεται, αλλά και πάλι δεν υπάρχουν όρια στην γέφυρα. Η πρώτη συνάρτηση κάνει τους αλγόριθμους να δημιουργούν και να επισκέπτονται λιγότερους κόμβους, όμως ο Greedy Best-first αλγόριθμος δεν είναι τόσο ακριβής όσο είναι με την δεύτερη ευρετική συνάρτηση. Έτσι χρησιμοποιήσαμε την πρώτη για τον A* και την δεύτερη για τον Greedy Best-first.

2.3 Σειρά των ανθρώπων

Εφόσον το αποτέλεσμα πολλών αλγορίθμων εξαρτάται από την σειρά εισαγωγής των ανθρώπων, είναι σημαντικό αυτή να διατηρείται όπως την εισήγαγε ο χρήστης. Για αυτό χρησιμοποιήσαμε ordered δομές δεδομένων, όπως λίστες και πλειάδες.

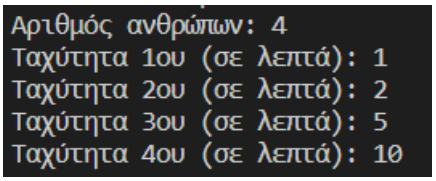
2.4 Επανάληψη κόμβων

Οι αλγόριθμοι επισκέπτονται τους ίδιους κόμβους πολλές φορές με αποτέλεσμα να καθυστερούν αρκετά στην εύρεση της λύσης για μεγάλο αριθμό ανθρώπων. Για την αποφυγή αυτού του προβλήματος η υλοποίησή μας διατηρεί τους κόμβους που έχει ήδη επισκεφτεί ο αλγόριθμος και έτσι αποφεύγονται οι επαναλήψεις. Εξαίρεση αποτελεί ο Iterative-deepening αλγόριθμος ο οποίος βασίζεται στην επανάληψη για την εύρεση του αποτελέσματος.

Σημείωση: λόγω αυτής της λειτουργίας ο Iterative-deepening αλγόριθμος είναι πιο αργός από τους υπόλοιπους και δημιουργεί και επισκέπτεται περισσότερους κόμβους.

3 Screenshots

Αρχικά το πρόγραμμα ζητάει από τον χρήστη το πλήθος των ατόμων που θέλει να προσθέσει στην μία άκρη της γέφυρας και στη συνέχεια μας ζητάει την ταχύτητα ξεχωριστά για κάθε άτομο. (Εισάγουμε τις ίδιες τιμές με το παράδειγμα της εργασίας)



Τέλος το πρόγραμμα μας εμφανίζει τις καλύτερες δυνατές λύσεις για την λύση του προβλήματος (για κάθε τύπο αναζήτησης) και το σύνολο των κόμβων που έχει διανύσει το πρόγραμμα μέχρι να φτάσει στη λύση.