23/11/19

Εναλλακτικές που είχαμε:

* Αντικειμενική συνάρτηση που θα υπολογίζει τον χρόνο. Στόχος είναι να μειωθεί ο μεγαλύτερος χρόνος για να επισκεφτούμε ένα κόμβο ακόμη και αν αργήσουμε να πάμε σε κάποιον που θα μπορούσαμε να πάμε νωρίτερα.

! Δεν γυρνάμε πίσω

! Κίνδυνος να μην χωράει τελικά κάτι (?)

30/11/19

Συζητήσεις για τους αλγορίθμους και πώς θα χειριστούμε το θέμα. Ουσιαστικά έλεγχος ότι όλοι θα έχουν την Κυριακή να παρουσιάσουν την ιδέα τους.

1/12/2019

1. Κοιτάμε όλοι τους κώδικες και συγκεντρώνουμε απορίες + προτείνουμε ποιον να ακολουθήσουμε

2. Τρίτη -> Κλήση για απορίες

3. Δουλειά μας: Να συνεχίσουμε να βελτιώνουμε τον αλγόριθμο μέχρι το κάθε φορτηγό να τρέχει σε λιγότερο από 4 ώρες

2/12/2019

Q&A

Q: Τι είναι το nodeCounter?

A: Όλα διατηρούν το ίδιο αντίγραφο της μεταβλητής που έχει δηλωθεί στο VRP και αυξάνει το ID.

Q: 1000F σε αριθμούς

A: Το F και το D δηλώνουν ότι ο αριθμός δεν είναι ακέραιος.

Q: method findCurrentQuickestRoute()

A: .stream() σαν for περνάει λίστες

Filter φιλτράρει μόνο αυτές που δεν έχουν τερματίσει και πετάει τις υπόλοιπες.

Από τις available επιστρέφει αυτή με τον ελάχιστο χρόνο.

Q: Δεν βρίσκαμε πώς σπας την αλυσίδα;

Α: Clarck Wright vs. τωρινός αλγόριθμος.

Q: Τι κάνει το totalTrucks στην εντολή routes = new ArrayList<Route>(totalTrucks)

Α: Είναι μια arraylist που έχει μέγεθος 25 και κάθε λίστα επιστρέφει διαδρομή

Αν βρούμε null κάνουμε finalize μεταβλητή στην findMinUnservice. Εγώ δεν μπορώ να συνεχίσω με null και κάνω επανάληψη μέχρι να μην το βρίσκω.

Κάθε μεταβλητή τύπου εκτός γνωστών δείχνει σε θέση πίνακα. Πρέπει πριν χρησιμοποιηθεί να υπάρχει ήδη κάποια θέση αντικειμένου που να δείχνει σε αυτό.

27/12/2019

‘Η θα δημιουργούμε γειτονιά στο local search με το μεγαλύτερο route ή με όλα για καλύτερο