Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών  
Προγραμματιστικές Γλώσσες Ι – 3η σειρά ασκήσεων

**ΑΝΑΛΥΣΗ ΤΟΥ ΤΡΟΠΟΥ ΕΠΙΛΥΣΗΣ**

**Βασική Ιδέα:** Πρέπει να βρούμε την πόλη εκείνη, από την οποία τα αυτοκίνητα θα απέχουν συνολικά (=άθροισμα αποστάσεων) τη μικρότερη απόσταση. Ταυτόχρονα πρέπει να ισχύει ο περιορισμός ότι το αμάξι που θα απέχει τη μεγαλύτερη απόσταση από την πόλη, θα απέχει το πολύ ένα βήμα παραπάνω από τη συνολική απόσταση όλων των άλλων πόλεων. Ο περιορισμός αυτός εξασφαλίζει ότι πάντοτε τα αυτοκίνητα θα κινούνται εναλλάξ, δηλαδή κανένα δε θα κάνει δύο διαδοχικές κινήσεις.

**Standard ML**

Για να βρούμε τη συγκεκριμένη πόλη εργαστήκαμε ως εξής στην ML:

1. Δημιουργήσαμε μια λίστα από λίστες με όλες τις δυνατές τελικές καταστάσεις των αυτοκινήτων. Για παράδειγμα αν και τότε οι τελικές δυνατές καταστάσεις είναι οι εξής:
2. Στη συνέχεια βρίσκουμε την διαφορά **κάθε** δυνατής τελικής κατάστασης με την αρχική μας. Εδώ προσέξτε ότι αν η τελική πόλη είναι μεγαλύτερη από την αρχική, πχ. είμαστε στη 2 και θέλουμε να πάμε στην 4, τότε απλά αφαιρούμε . Αν όμως είμαστε στην 2 και θέλουμε να πάμε στην 0 τότε πρέπει να κάνουμε την πράξεις (όπου T το σύνολο των πόλεων). Αν η αρχική μας κατάσταση είναι η τότε για την τελική κατάσταση [0, 0, 0, 0] η απόσταση είναι , ενώ για την είναι η
3. Τώρα βρίσκουμε το άθροισμα και το μέγιστο κάθε διαφοράς που βρήκαμε στο προηγούμενο βήμα. Για παράδειγμα για τη διαφορά το μέγιστο είναι το 3 και το άθροισμα το 9.
4. Ελέγχουμε αν ο συνδυασμός μέγιστου-αθροίσματος ικανοποιούν τον περιορισμό που εξηγήσαμε στην **βασική ιδέα**. Πχ. για max=3, sum=5 ο περιορισμός ικανοποιείται, όχι όμως για max=3, sum=4.
5. Αν λοιπόν ικανοποιείται η παραπάνω συνθήκη τότε το άθροισμα αποστάσεων για την που έχουμε βρει για τη συγκεκριμένη τελική κατάσταση είναι αποδεκτή λύση. Έτσι συγκρίνουμε το συγκεκριμένο άθροισμα για να διαπιστώσουμε αν είναι το ελάχιστο που έχουμε βρει προς το παρόν και συνεχίζουμε την αναζήτηση για την επόμενη πιθανή τελική κατάσταση. Προφανώς μαζί με την ελάχιστη απόσταση κρατάμε και το αντίστοιχο index.

Η

-- Τώρα υπάρχουν γενικά πολλές λεπτομέρειες στην υλοποίηση, και το να βρεις απευθείας την τελική συνάρτηση μου φαίνεται αρκετά δύσκολο. Θα πρότεινα να σπάσετε το πρόβλημα σε πολλές μικρές συναρτήσεις που κάθε μία κάνει μία απλή δουλειά (πχ. μία συγκρίνει τη λίστα [2, 0, 2, 2] με τη [3,3,3,3] και βρίσκει την απόστασή τους [1,3,1,1], ενώ μετά άλλη μία βρίσκει το max και το sum της λίστας αυτής. Μετά προσπαθήστε να συγχωνεύσετε όλες αυτές τις συναρτήσεις για να αυξήσετε την απόδοση. -- Τα δικά μου βήματα ήταν αρχικά τα εξής: 1. Διάβασμα αρχείου 2. Βρίσκω όλες τις δυνατές τελικές καταστάσεις (λίστα από λίστες) 3. Βρίσκω τις αποστάσεις μιας τελικής κατάστασης από την αρχική (λίστα) και βρίσκω τις αποστάσεις για κάθε δυνατή τελική κατάσταση (λίστα από λίστες) 4. Βρίσκω για κάθε απόσταση το max και το sum για κάθε ζεύγος (πιθανή τελική κατάσταση - αρχική κατάσταση) 5. Ελέγχω ποια ζεύγη max και sum είναι αποδεκτά. 6. Βρίσκω το min από τα αποδεκτά sum και το index του (τελική απάντηση).