**PTOD**

Η παρούσα βιβλιοθήκη, χρησιμοποιείται για την μετατροπή ενός γραμμικού προβλήματος από την primal στην dual μορφή του.

Η ανάγνωση του προβλήματος γίνεται από αρχείο κειμένου .txt και η έξοδος είναι επίσης .txt αρχείο.

To GitHub repository του λογισμικού είναι το:

<https://github.com/Dimtop/PTOD>

Η βιβλιοθήκη lpparser μπορεί να βρεθεί στο:

<https://github.com/Dimtop/Linear-Problem-Parser>

ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ

Για την ανάπτυξη της παρούσας βιβλιοθήκης, χρησιμοποιήθηκε η γλώσσα προγραμματισμού javascript, και η τελευταία έκδοση του nodejs, η οποία είναι και απαραίτητη για την χρήση του λογισμικού. Η μόνη επιπλέον βιβλιοθήκη που χρησιμοποιείται, είναι η fs (διεπαφή με το σύστημα αρχείων του υπολογιστή) που συμπεριλαμβάνεται στο βασικό πακέτο του nodejs.

ΤΡΟΠΟΣ ΧΡΗΣΗΣ

1. Εγκαθιστούμε την τελευταία έκδοση το nodejs
2. Κατεβάζουμε το αρχείο ptod.js από το github ή κάποια άλλη τοποθεσία.
3. Κατεβάζουμε το αρχείο lpparser.js από το github ή από κάποια άλλη τοποθεσία, και το αποθηκεύουμε στον ίδιο φάκελο με το ptod.js
4. Δημιουργούμε ένα νέο αρχείο στον ίδιο φάκελο, με όποιο όνομα θέλουμε, έστω test.js
5. Προσθέτουμε στην πρώτη γραμμή του test.js την εξής γραμμή κώδικα:

const ptod= require(“./ptod.js”);

ΠΡΟΣΟΧΗ: Εάν τα δύο αρχεία δεν είναι στον ίδιο φάκελο, δεν θα μπορεί να τρέξει η βιβλιοθήκη.

1. Δημιουργούμε ένα αρχείο txt που περιέχει το γραμμικό πρόβλημα, κατά προτίμηση στον ίδιο φάκελο, έστω lp1.txt. Προσθέτουμε την εξής γραμμή κώδικα:

ptod(<filename>);

και αντικαθιστούμε το <filename> με την απόλυτη διαδρομή του αρχείου εισόδου μας. Εάν είναι στον ίδιο φάκελο, γράφουμε ‘**./**lp1.txt’

Ολοκληρωμένο το αρχείο του κώδικά μας, θα πρέπει να είναι:

const ptod= require(‘./ptod.js’);

ptod(‘./lp1.txt’);

1. Μέσω ενός terminal της επιλογής μας, μεταβαίνουμε στον φάκελο που βρίσκεται το αρχείο μας, και τρέχουμε την εντολή node test.js

Σε περίπτωση που έχουμε δώσει άλλο όνομα στο αρχείο javascript, αντικαθιστούμε το test.js με το αντίστοιχο όνομα.

1. Εάν όλα πάνε καλά, θα μας εμφανιστεί μήνυμα επιβεβαίωσης στο τερματικό, και το αρχείο εξόδου θα έχει δημιουργηθεί στον ίδιο φάκελο, με το όνομα output.txt

ΠΑΡΑΔΟΧΕΣ

1. Η ορθή εκτέλεση του προγράμματος, προϋποθέτει πώς η αντικειμενική συνάρτηση μαζί με τον δείκτη μορφής του προβλήματος βρίσκεται στην πρώτη γραμμή, και κάθε ένας περιορισμός βρίσκεται σε ξεχωριστή γραμμή επίσης. Ο δείκτης των περιορισμών θα πρέπει να είναι ένας από τους εξής: st ή s.t. ή subject to και θα πρέπει να βρίσκεται στην αρχή της γραμμής του πρώτου περιορισμού.
2. Τα σύμβολα των περιορισμών είναι = για το ίσον, >= για το μεγαλύτερο ή ίσο και <= για το μικρότερο ή ίσο.
3. Επιτρέπονται μόνο αριθμοί ως δείκτες μεταβλητών. Περιορισμοί της μορφής

xj >= 0 (j = 1,…,n) δεν θα πρέπει να συμπεριήφθούν.

1. Όλες οι μεταβλητές θα πρέπει να χρησιμοποιούν το ίδιο σύμβολο π.χ. μόνο x ή μόνο y.

ΧΡΗΣΤΙΚΟΤΗΤΑ

1. Επιτρέπονται δείκτες μεταβλητών οποιουδήποτε αριθμού ψηφίων (π.χ. x121) και επίσης λαμβάνονται υπόψη μεταβλητές με δείκτη 0, δηλ. x0
2. Oποιαδήποτε κενά ή κενές γραμμές, spaces και tabs δεν δημιουργούν πρόβλημα.
3. Κεφαλαία και πεζά γράμματα επιτρέπονται, είτε εξ’ ολοκλήρου, είτε σε μίξη.
4. Λαμβάνονται υπόψη και δεκαδικοί αριθμοί ως παράγοντες.

ΕΛΕΓΧΟΙ

Οι έλεγχοι που γίνονται κατά το parsing του προβλήματος, βρίσκονται αναλυτικά στο documentation της βιβλιοθήκης lpparser.

DOCUMENTATION

**-convertProblemType(filename)**

Δέχεται ως όρισμα έναν πίνακα MinMax, ο οποίος περιέχει ένα στοιχείο που περιγράφει αριθμητικά το είδος του γραμμικού προβλήματος (Min=-1, Max=1). Η λειτουργία της, είναι να μετατρέψει την αντίθετη τιμή από αυτήν που δίνει ο πίνακας, να αντιστρέψει δηλ. το είδος του προβλήματος.

**-convertRestrictionsToVariables(Α)**

Δέχεται τον πίνακα των παραγόντων των μεταβλητών της αντικειμενικής συνάρτησης της primal μορφής, και εκτελεί την πράξη transpose πάνω στον πίνακα, προσθέτοντας 0 για κάθε μεταβλητή που παραλείπεται (επειδή θεωρητικά έχει πολλαπλασιαστεί με το 0). Να σημειωθεί εδώ πως λαμβάνονται υπόψη και μεταβλητές x0.

**-normalizeConvertedProblem(b,c,A)**

Η λειτουργία αυτής της συνάρτησης είναι να ομαλοποιήσει τα δεδομένα, κάνοντας μικρές αλλαγές, ώστε να τυπωθούν σωστά.

**-printConvertedProblem(ATranspose,c,b,Eqin,MinMaxInverted)**

Είναι η συνάρτηση που τυπώνει το πρόβλημα μετά την μετατροπή. Γνωρίζοντας πως μετά το αρχικό parsing έχουμε τους πίνακες c(παράγοντες αντικειμενικής συνάρτησης), A(παράγοντες περιορισμών), b(σταθερές περιορισμών), Eqin(τελεστές περιορισμών), MinMax(είδος προβλήματος), μπορούμε να συνθέσουμε την dual μορφή χρησιμοποιώντας τον αντεστραμένο MinMax, τον b ως πίνακα παραγόντων της αντικειμενικής συνάρτησης, τον ATranspose ως αντικαταστάτη του Α, τον Eqin ως έχει και τον c ως πίνακα σταθερών για τους περιορισμούς.

**-ptod(fileNameInput)**

Είναι η κύρια συνάρτηση. Καλεί την lpparser για να κάνει parse το πρόβλημα που βρίσκεται στο αρχείο της διαδρομής *fileNameInput.* Στην συνέχεια καλεί διδοχικά τις *convertProblemType, convertRestrictionsToVariables, normalizeConvertedProblem , printConvertedProblem.* Με το αποτέλεσμα της τελευταίας τυπώνει το αρχείο output.txt