**LPPARSER**

Η παρούσα βιβλιοθήκη, χρησιμοποιείται για την μετατροπή ενός γραμμικού προβλήματος από την γενική μορφή του, σε συμπτυγμένη μορφή με την χρήση μητρών.

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/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 lpparser.js από το github ή από κάποια άλλη τοποθεσία.
3. Δημιουργούμε ένα νέο αρχείο στον ίδιο φάκελο, με όποιο όνομα θέλουμε, έστω test.js
4. Προσθέτουμε στην πρώτη γραμμή του test.js την εξής γραμμή κώδικα:

const lpparser = require(“./lpparser.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:

lpparser(<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 lpparser = require(‘./lpparser.js’);

lpparser(‘./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) θα πρέπει να γράφονται αναλυτικά για κάθε j, δηλ. x1 >=0, x2 >=0

με κάθε έναν περιορισμό σε νέα γραμμή.

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

1. Ελέγχεται ένα υπάρχει η λέξη min ή max μπροστά από την αντικειμενική συνάρτηση. Εάν όχι, εμφανίζεται στον χρήση το μήνυμα :

“The problem type identifier was not found. Please ensure that the first three characters of your document make up either the word min or max .”

1. Ελέγχεται αν υπάρχει η λέξη st ή s.t. ή subject to πριν τους τεχνολογικούς περιορισμούς. Εάν όχι, εμφανίζεται στον χρήστη το μήνυμα:

«The restrictions initializer is misplaced or not included at all. Please ensure that it is placed right before the first restriction.”

1. Ελέγχεται εάν όλες οι μεταβλητές χρησιμοποιούν το ίδιο σύμβολο. Εάν όχι το παρακάτω μήνυμα εμφανίζεται στον χρήστη:

“The variable symbols are not the same. Please use the same symbol (x is suggested) for all the variables, followed by it’s index.”

1. Ελέγχεται αν λείπει κάποιο πρόσημο, ή αν υπάρχουν περιττοί χαρακτήρες μεταξύ των μεταβλητών της αντικειμενικής συνάρτησης και των περιορισμών. Εάν ναι, τότε

εμφανίζεται στον χρήστη το εξής μήνυμα:

«There was an error with an operand. Please ensure that all the operands are placed in the right side of the variables’ factors.”

1. Ελέγχεται εάν όλες οι μεταβλητές έχουν δείκτη. Εάν έστω μια δεν έχει εμφανίζεται το παρακάτω μήνυμα:

“A variable without index was found. Please ensure that all the variables have their indexes assigned by their right side.”

1. Ελέγχεται εάν όλοι οι περιορισμοί έχουν ένα εκ των συμβόλων =, >=, <=. Εάν κάποιος δεν έχει, εμφανίζεται το παρακάτω μήνυμα:  
   “No restriction type operator was found for the restriction number #. Please include all the operators, choosing from =, <=, >=”
2. Ελέγχεται εάν όλοι οι περιορισμοί έχουν δεξιό μέλος. Εάν κάποιος δεν έχει, εμφανίζεται το εξής μήνυμα:

“The restriction number # has no right side.”

DOCUMENTATION

**-parseLinearProblem(filename)**

Δέχεται ως όρισμα την διαδρομή του αρχείου εισόδου, και παράγει το αρχείου εξόδου στην ίδια τοποθεσία, με το όνομα output.txt

**-normalizeData(data)**

Η συνάρτηση αυτή, δέχεται το περιεχόμενο του αρχείο κειμένου εισόδου ως string και επιστρέφει ένα string, από το οποίο έχουν αφαιρεθεί όλα τα κενά, έχουν γίνει όλοι οι χαρακτήρες πεζοί και όλο το κείμενο έχει συμπτυχτεί σε μία γραμμή. Πριν από κάθε σημείο όπου άρχιζε νέα γραμμή στο αρχικό κείμενο, τοποθετείται ο δείκτης «\_nl\_”.

**-getProblemTypeIdentifier(data)**

Λαμβάνει το ομαλοποιημένο κείμενο και εξετάζει τους τρεις πρώτους χαρακτήρες του. Εάν αυτοί σχηματίζουν την λέξη max επιστρέφει [1], αν σχηματίζουν την λέξη min [-1] και σε κάθε άλλη περίπτωση εμφανίζει μήνυμα σφάλματος και τερματίζει το πρόγραμμα.

**-getFunctionLimit(data)**

Η συνάρτηση αυτή, χρησιμοποιείται για να βρούμε που τελειώνει η δοθείσα συνάρτηση του προβλήματος. Για να επιτευχθεί αυτό, βρίσκουμε την θέση του δείκτη έναρξης των περιορισμών. Παράλληλα ελέγχεται και η σωστή θέση του. Συγκεκριμένα, εξετάζουμε την πρώτη θέση του δείκτη «\_nl\_” στο string του προβλήματος. Αν αυτή η θέση αυξημένη κατά 4 (το μήκος του \_nl\_) δεν αντιστοιχεί στην θέση ενός εκ των st, s.t. ή subjectto στο string, τότε εμφανίζεται μήνυμα σφάλματος. Ειδάλλως, επιστρέφεται η θέση του δείκτη, που σηματοδοτεί και το τέλος της αντικειμενικής συνάρτησης, καθώς και ο ίδιος ο δείκτης έναρξης των περιορισμών.

**-getFunctionArray(data, functionLimit)**

Η συνάρτηση αυτή δέχεται το αλφαριθμητικό του προβλήματος και επιστρέφει έναν υπό πίνακα αυτού, από την θέση 3, μέχρι την θέση functionLimit, η οποία σηματοδοτεί το τέλος της συνάρτησης. Το αποτέλεσμα είναι να πάρουμε ένα αλφαριθμητικό που περιέχει μόνο την αντικειμενική συνάρτηση (χωρίς το min/max στην αρχή), ακολουθούμενη από τον δείκτη \_nl\_.

**-getVariables(functionArray)**

Δέχεται ως όρισμα το αλφαριθμητικό της συνάρτησης και αναζητά όλους τους χαρακτήρες που αποτελούν γράμματα του αγγλικού αλφαβήτου και αποτελούν υποψήφια σύμβολα μεταβλητών του προβλήματα, χρησιμοποιώντας Regular Expressions, επιστρέφοντας έναν πίνακα που περιέχει αυτούς και τις θέσεις τους.

**-normalizeAndCheckVariables(variables)**

Δέχεται ως όρισμα τον πίνακα όλων των γραμμάτων που υπάρχουν μέσα στην αντικειμενική συνάρτηση. Μέσα στα γράμματα που περιέχονται σε αυτόν τον πίνακα συμπεριλαμβάνονται και τα ‘n’ και ‘l’ καθώς η συνάρτηση στο τέλος της έχει τον δείκτη \_nl\_. Αυτά αφαιρούνται, καθώς δεν είναι υποψήφια σύμβολα μεταβλητών. Στην συνέχεια ελέγχεται αν όλα τα στοιχεία αυτού του πίνακα είναι ίδια, για να διασφαλιστεί ότι χρησιμοποιείται το ίδιο σύμβολο για όλες τις μεταβλητές. Εάν βρεθεί έστω και ένα διαφορετικό από τα υπόλοιπα, εμφανίζεται μήνυμα σφάλματος.

**-getVariableSymbols(variables)**

Δεδομένου του ότι έχει ελεγχθεί και διορθωθεί ο πίνακας των συμβόλων μεταβλητών ώστε όλα του τα στοιχεία να είναι τα ίδια, λαμβάνεται το πρώτο στοιχείο του (θα μπορούσε να είναι και οποιοδήποτε άλλο) και επιστρέφεται, έτσι ώστε να πάρουμε το ενιαίο σύμβολο των μεταβλητών του προβλήματος.

**-getVariablesIndexArray(functionArray, variableSymbol)**

Δέχεται το αλφαριθμητικό της συνάρτησης και το σύμβολο μεταβλητών ως ορίσματα και επιστρέφει έναν πίνακα με τον δείκτη κάθε μεταβλητής και την θέση του στο αλφαριθμητικό. Αυτό επιτυγχάνεται αν για κάθε εμφάνιση του συμβόλου μεταβλητών στην αντικειμενική συνάρτηση, πάρουμε όσους χαρακτήρες μετά από αυτό είναι αριθμοί, χρησιμοποιώντας Regular Expressions. Αν βρεθεί χαρακτήρας που δεν είναι αριθμός επιστρέφεται μήνυμα σφάλματος.

-**initializeFactorsArray(variablesIndexArray, factorsArray)**

Δέχεται ως ορίσματα τον πίνακα με τις θέσεις των δεικτών των μεταβλητών και έναν άδειο πίνακα τον οποίον θα γεμίσει με τους παράγοντες κάθε μεταβλητής. Αρχικά βρίσκει τον μεγαλύτερο δείκτη μεταβλητής και μετά, αρχικοποιεί όλες τις θέσεις του πίνακα των παραγόντων με ‘’ εάν ο δείκτης υπάρχει μέσα στον πίνακα δεικτών και με ‘0’ εάν ο δείκτης δεν υπάρχει. Έτσι διασφαλίζουμε ότι παίρνουμε τον παράγοντα 0 για κάθε μεταβλητή που παραλείπεται. Για παράδειγμα, εάν έχουμε την συνάρτηση 2x1 + 3x3 , ο πίνακας παραγόντων θα αρχικοποιηθεί ως [‘0’,‘’,’0’,’’]. Θα πρέπει να σημειωθεί πως είναι αποδεκτές μεταβλητές με δείκτη 0.

**-getFactors(functionArray,variablesIndexArray,factorsArray)**

Δέχεται ως ορίσματα το αλφαριθμητικό της συνάρτησης, τον πίνακα δεικτών των μεταβλητών και τον αρχικοποιημένο πίνακα παραγόντων. Η λειτουργία της είναι η εξής. Για κάθε δείκτη μεταβλητής, μεταβαίνει δύο θέσεις πίσω στο αλφαριθμητικό της συνάρτησης. Έτσι λοιπόν, ο έλεγχος θα βρίσκεται σε αυτό που υπάρχει αριστερά κάθε μεταβλητής (όπου και είναι η θέση των παραγόντων). Στην συνέχεια, εξετάζουμε προς τα αριστερά, όσο βρισκόμαστε μέσα στα όρια του πίνακα, εάν ο τρέχων χαρακτήρας είναι αριθμός ( ή τελεία αν πρόκειται για δεκαδικό παράγοντα). Κάθε αριθμός που βρίσκεται λοιπόν, προστίθεται στην θέση του πίνακα παραγόντων που αντιστοιχεί στην αριθμητική τιμή του δείκτη της μεταβλητής που εξετάζεται. Αυτή λοιπόν η διαδικασία μας δίνει τον πίνακα παραγόντων με όλους τους παράγοντες στις αντίστοιχες θέσεις με τους δείκτες. Το μόνο πρόβλημα είναι ότι εφόσον διαβάσαμε τους παράγοντες από τα δεξιά προς τα αριστερά, αυτοί θα έχουν αποθηκευτεί ανάποδα. Γι’ αυτόν τον λόγο, κάθε στοιχείο του πίνακα παραγόντων αντιστρέφεται.

**-getAndCheckOperands(functionArray,variablesIndexArray,factorsArray,operandsArray)**

Δέχεται το αλφαριθμητικό της συνάρτησης, τον πίνακα δεικτών των μεταβλητών, τον πίνακα παραγόντων και έναν άδειο πίνακα για να αποθηκευτούν οι τελεστές. Σκοπός είναι να βρεθούν οι τελεστές, να ελεγχθούν για την ορθότητά τους και να επιστραφούν. Η λογική που ακολουθείται, λαμβάνει τους τελεστές ως πρόσημα των παραγόντων κάθε μεταβλητής, που θα έπρεπε να βρίσκονται αριστερά κάθε παράγοντα. Ο μόνος τελεστής που μπορεί να παραληφθεί, είναι ο τελεστής της πρώτης μεταβλητής, που σε περίπτωση που είναι ο +, μπορεί να παραλείπεται. Γι’ αυτόν τον λόγο, εξετάζεται ξεχωριστά με τον εξής τρόπο: Ελέγχεται τον πρώτο στοιχείο του αλφαριθμητικού της δοθείσας συνάρτησης. Σε περίπτωση που αυτό δεν είναι ούτε γράμμα, ούτε αριθμός, ελέγχεται αν είναι ένας από τους επιτρεπόμενους τελεστές (+ και -). Αν είναι ένα εκ των δύο τότε αποθηκεύεται στον πίνακα τελεστών μαζί με τον δείκτη της μεταβλητής στην οποία αναφέρεται. Αν δεν βρεθεί κάποιος τελεστής , αποθηκεύεται αυτόματα ο +. Για τις υπόλοιπες μεταβλητές, οι θέσεις των οποίων λαμβάνονται από τον πίνακα δεικτών μεταβλητών, ακολουθείται η ίδια διαδικασία. Στην περίπτωση όμως που δεν βρεθεί τελεστής, εμφανίζεται μήνυμα σφάλματος.

**-applyOperandsToFactors(factorsArray, operandsArray)**

Δέχεται ως ορίσματα τον πίνακα παραγόντων και τον πίνακα τελεστών. Η λειτουργία που επιτελεί είναι να εφαρμόσει τα πρόσημα στους παράγοντες. Κάθε παράγοντας που είναι ‘’ μετατρέπεται αρχικά σε 1, έτσι ώστε να έχει αριθμητική μορφή για να μπορεί να εφαρμοστεί το πρόσημο. Για κάθε στοιχείο λοιπόν του πίνακα τελεστών, το αντίστοιχο στοιχεί του πίνακα παραγόντων μετατρέπεται σε αριθμό, θετικό αν ο τελεστής είναι + και αρνητικό αν ο τελεστής είναι -. Στο τέλος κάθε στοιχεία του πίνακα παραγόντων που είναι ‘0’ μετατρέπεται από χαρακτήρα στον αριθμό 0.

**-getRestrictionArrays(data,functionLimit,restrictionsIdentifier)**

Δέχεται ως ορίσματα το αλφαριθμητικό ολόκληρου του προβλήματος, το όριο της αντικειμενικής συνάρτησης και τον δείκτη έναρξης των περιορισμών. Προσθέτει στο όριο της συνάρτησης το όριο του δείκτη και επιστρέφει έναν πίνακα, κάθε στοιχείο του οποίου είναι η συνάρτηση ενός περιορισμού ακολουθουμένη από τον δείκτη \_nl\_.

**-getRestrictiontTypes(restrictionsArrays)**

Δέχεται ως όρισμα τον πίνακα των περιορισμών, τον διατρέχει και για κάθε έναν αναζητά και παίρνει την θέση του =, <= ή του >= στο αντίστοιχο αλφαριθμητικό, επιστρέφοντας έναν πίνακα με τις κατάλληλες τιμές ακεραίων (0,-1,1 αντίστοιχα) για κάθε είδος περιορισμού. Αν για έναν περιορισμό δεν βρεθεί ένας από τους παραπάνω τελεστές, επιστρέφεται μήνυμα σφάλματος.

**-getRestricitonRightSides(restrictionArrays,restrictionTypes,restrictionTypeIndexes)**

Δέχεται τον πίνακα των περιορισμών, τον πίνακα του είδους των περιορισμών και έναν πίνακα με τις θέσεις των τελεστών ισότητας/ανισότητας για κάθε περιορισμό. Για κάθε έναν περιορισμό, παίρνει το μέρος του αλφαριθμητικού, από την αμέσως επόμενη θέση του τελεστή, μέχρι το τέλος ήτοι το δεξιό μέλος της μαθηματικής παράστασης. Αν κάποιος περιορισμός δεν έχει δεξιό μέλος, επιστρέφεται μήνυμα λάθους.

**-parseRestrictionsLeftSides(restrictionsArrays,restrictionTypesIndexes,variableSymbol)**

Δέχεται ως ορίσματα τον πίνακα των περιορισμών, τον πίνακα θέσεων των τελεστών ισότητας/ανισότητας και το σύμβολο μεταβλητής. Επιστρέφει στο τέλος τον πίνακα μεταβλητών των περιορισμών, τον πίνακα συμβόλων και δεικτών μεταβλητών, τον πίνακα παραγόντων των μεταβλητών και τον πίνακα τελεστών. Η διαδικασία που ακολουθείται για κάθε περιορισμό είναι η ίδια που ακολουθείται για την αντικειμενική συνάρτηση. Συγκεκριμένα καλούνται οι εξής συναρτήσεις, για τις οποίες υπάρχει τεκμηρίωση πιο πάνω στο έγγραφο:

1. getVariables()
2. normalizeAndCheckVariables()
3. getVariableSymbol()
4. getVariableIndexesArray()
5. initializeFactorsArray()
6. getFactors()
7. getAndCheckOperands()
8. applyOperandsToFactors()

**-printLinearProblem(A,b,c,Eqin,MinMax)**

Η συνάρτηση αυτή, δημιουργεί το αρχείο εξόδου, με το συμπτυγμένο γραμμικό πρόβλημα.