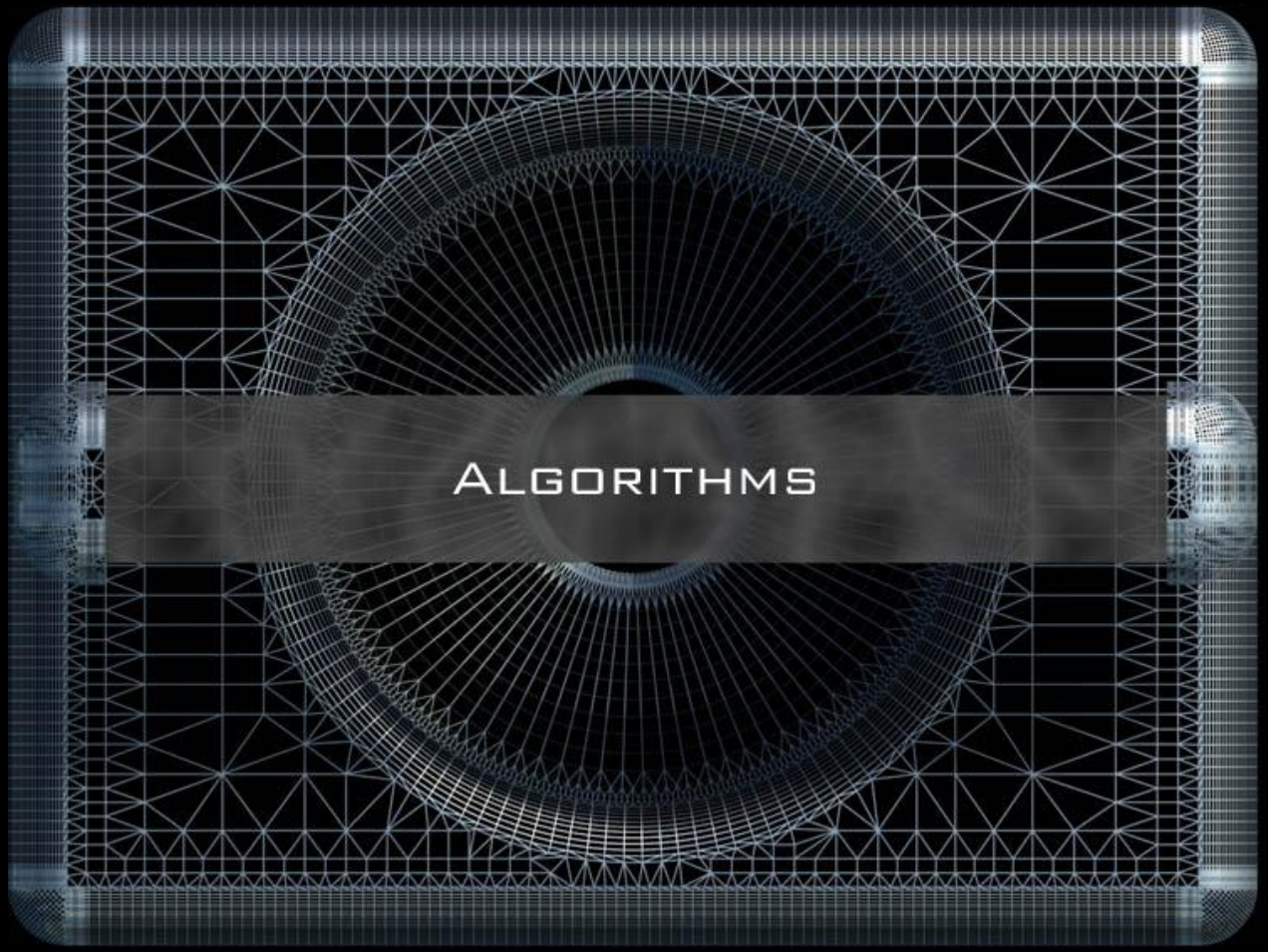
1η Προγραμματιστική Άσκηση



ΜΑΓΚΟΣ ΡΑΦΑΗΛ-ΓΕΩΡΓΙΟΣ 3100098

ΜΠΟΓΔΑΝΟΣ ΜΙΧΑΗΛ 3100123

Περιεχόμενα

Κλάση Regex

Στην κλάση αυτή έχουμε τον ορισμό της κανονικής έκφρασης που χρησιμοποιείται για να ελέγχουμε εάν το input που δίνεται από τον χρήστη είναι σε 2-CNF (συλλογή από clauses-όρους καθένας από τους οποίους αποτελείται από την διάζευξη ακριβώς δύο στοιχείων-literals τα οποία στοιχεία είναι λογικές μεταβλητές ή η αρνησή τους).Η κανονική έκφραση αποθηκεύεται στο final String REGEX και σε φυσική γλώσσα σημαίνει «η συμβολοσειρά που αποτελείται από μια ή περισσότερες επαναλήψεις των υποσυμβολοσειρών (¬xV ή xV ακολουθούμενη από το ¬(μία ή καμία φορά) και το y).Τα x και y μπορούν να είναι δύο οποιαδήποτε γράμματα του λατινικού αλφαβήτου.

Επιπλέον ορίζεται η μέθοδος static boolean matches με όρισμα ένα String str που επιστρέφει true ή false εάν ταιριάζει ή οχι το όρισμα str με την κανονική έκφραση που είναι αποθηκευμένη στην μεταβλητή REGEX και περιγράφηκε παραπάνω.

Τέλος ορίζεται η μέθοδος static String randomSat() η οποία παράγει μια τυχαία συμβολοσειρά που ανήκει όμως στην κανονική έκφραση που έχουμε ορίσει(δηλαδή είναι στιγμιότυπο 2-CNF).Κάθε literal έχει 50% πιθανότητες να είναι άρνηση.Το μέγιστο πλήθος από clauses που μπορεί να παραχθεί είναι 8(το max του r.nextInt(6)+3 είναι το 5+3=8) και το ελάχιστο 3(το min του r.nextInt(6)+3 είναι το 0+3=3).Επίσης, το μέγιστο πλήθος από literals που μπορεί να παραχθεί είναι 8(το max του r.nextInt(4)+4 είναι το 3+4=7) και το ελάχιστο 4(το min του r.nextInt(4)+4 είναι το 0+4=3).Σε περίπτωση που το literal που επιλεγεί αριστερά του V είναι ίδιο με το literal που επιλεγεί δεξιά του τότε επιλέγουμε κάποιο άλλο.

Κλάση NodeComparator

Στην κλάση αυτή έχουμε τον ορισμό ενός Comparator<Node> που θα χρησιμοποιήσουμε για sort δομών δεδομένων που θα περιέχουν Node αντικείμενα.Συνεπώς,υπάρχει η υπερφόρτωση της συνάρτησης compare που παίρνει ως όρισμα 2 αντικείμενα τύπου Node και επιστρέφει ως αποτέλεσμα ένα int που προκύπτει από κλήση της compareTo στα ονόματα των αντικειμένων Node (τα οποία επιστρέφονται με την μέθοδο getNode())

Κλάση Node

Η κλάση Node έχει τις εξής μεταβλητές:

- Static fields:
  - ✓ int newNodeX=250 : Μεταβλητή που θα καθορίζει τη συντεταγμένη Χ κάθε κόμβου που θα σχεδιαστεί.Ο πρώτος κόμβος θα σχεδιαστεί με Χ=250.Η μεταβλητή αυτή θα ανανεώνεται κατά την σχεδίαση με νέες τιμές.
  - ✓ int newNodeY=20 : Ακριβώς το ίδιο με την παραπάνω μεταβλητή μόνο που αφορά την συντεταγμένη Υ.
  - ✓ boolean leftNode=true : Κατα τον σχεδιάσμό των κόμβων θα δημιουργηθούν 2 στήλες κόμβων,μία αριστερά που θα βρίσκονται τα μη αρνητικά literal και μια δεξιά που θα βρίσκονται οι αρνήσεις.Συνεπώς η μεταβλητή αποφασίζει εάν ο κόμβος που θα δημιουργηθεί θα βρίσκεται αριστερά ή δεξιά.Η μεταβλητή μόλις αναθέσει την τιμή της στη μεταβλητή isLeft του κόμβου,λαμβάνει την αντίθετη τιμή από αυτή που έχει για μελλοντική χρήση.
- Instance fields:
  - ✓ Point p : Αντικείμενο τύπου Point που περιέχει τις συντεταγμένες του κέντρου του κύκλου που αναπαριστά τον κόμβο.
  - ✓ String name : Μεταβλητή που περιέχει το όνομα του κόμβου (δηλαδή το literal)
  - ✓ boolean isLeft : Μεταβλητή που θα είναι true αν ο κόμβος πρέπει να σχεδιαστεί αριστερά και false για δεξιά.
  - ✓ boolean visited : Μεταβλητή που θα είναι true εάν ο αλγόριθμος DFS(υλοποίηση στην κλάση GraphCanvas) έχει επισκεφτεί τον κόμβο και false αλλιώς.
  - ✓ ArrayList<Node> connectedTo : Δομή ArrayList<Node> που θα περιλαμβάνει τους κόμβους στους οποίους έχει πρόσβαση(παρέχει σύνδεση) το συγκεκριμένο αντικείμενο-κόμβος.

- Constructor : Λαμβάνει ως όρισμα το όνομα του κόμβου και το αναθέτει στην κατάλληλη μεταβλητή.Ακολούθως,αναθέτει την τιμή της στατικής μεταβλητής leftNode στο πεδίο isLeft του αντικειμένου και θέτει την αντίθετη τιμή από αυτή που έχει τώρα στην μεταβλητή leftNode,ώστε ο επόμενος κόμβος που θα κατασκευαστεί να είναι στην δεξιά στήλη.

- Methods :
  - ✓ void drawNode(int x, int y, String name, Graphics g) : Σχεδιάζει τον κόμβο με όνομα name στις συντεταγμένες x,y που δίνονται ως όρισμα.Αρχικά θέτει το χρώμα με το οποίο θα σχεδιαστεί ο κόμβος (το οποίο επιλέγεται από το property του αρχείου Conf.properties ,η ανάλυση των properties θα γίνει στην κλάση AppletProperties και Configs).Οι κόμβοι θα αναπαριστώνται ως κύκλοι οι οποίοι χρωματίζονται με το προηγούμενο χρώμα και σχεδιάζονται με κλήση της fillOval η οποία θα σχεδιάσει κύκλο στις συντεταγμένες x,y διότι ως όρισμα για μήκος και πλάτος θα δωθεί η διάμετρος που επιθυμούμε να έχει ο κόμβος(μεταβλητή Configs.nodeDiameter).Στη συνέχεια σχεδιάζουμε μάυρο(αλλάζουμε το χρώμα των γραφικών σε μαύρο) το περίγραμμα του κόμβου με κλήση της drawOval και με τα ίδια ορίσματα με την fillOval.Ακολούθως,ευθυγραμμίζουμε το όνομα που θα έχει ο κόμβος. ανάλογα με το αν είναι άρνηση(δηλ. 2 χαρακτήρες θα σχεδιαστούν οπότε πρέπει να γίνει κατάλληλη ευθυγράμμιση) ή όχι (άρα 1 χαρακτήρας και ευθυγραμμίζουμε κατάλληλα).Τέλος, αυξάνουμε το newNodeX κατά 120 εάν ο κόμβος ήταν αριστερά (ώστε ο επόμενος που θα είναι δεξιά να ζωγραφιστεί σε απόσταση x+120 από τον τρέχοντα κόμβο) ή εάν ήταν δεξιά τότε αυξάνουμε τα newNodeY κατά 120 και το newNodeeX μειώνεται κατά 120 (ώστε ο επόμενος κόμβος να σχεδιαστεί στην αριστερή στήλη στη θέση y+120 , x-120 δηλαδή στο x που έχουν όλοι οι αριστεροί κόμβοι και στο y που δείχνει ότι πήγαμε σε επόμενη «σειρά» από κόμβους).
  - ✓ Point Point getCirclePoint(Node n,double angle) : Επιστρέφει ένα σημείο πάνω στον κύκλο που αντιστοιχεί στην γωνία angle.Οι συντεταγμένες δίνονται από τις γνωστές τριγωνομετρικές σχέσεις x=getX()+r*cos(angle) και y=getY()+r*sin(angle) όπου x,y αντιστοιχεί στο κέντρο του κύκλου και το r το μισό της διαμέτρου.
  - ✓ Point getRandomCirclePoint(Node n,double startAng,double endAng) : Αρχικά η μέθοδος υπολογίζει μια τυχαία γωνία μεταξύ startAng και endAng και χρησιμοποιεί την getCirclePoint με όρισμα αυτλη την τυχαία γωνία και επιστρέφει ένα τυχαίο σημείο ως αποτέλεσμα.

- ✓ void drawConnection(Node dest) : Προσθέτει τον κόμβο dest στην ArrayList connectedTo και τακτοποιεί λεξικογραφικά τους κόμβους (η τακτοποίηση γίνεται με τον NodeComparator που περιγράφηκε στη σελ.2)
- ✓ void drawConnection(Graphics2D g,Node dest) : Σκοπός της μεθόδου είναι να ζωγραφίσει την γραμμή που ενώνει τον κόμβο this με τον κόμβο dest.Αρχικά επιλέγουμε ως χρώμα αυτό που έχει επιλεγεί στην κλάση Configs.Ακολούθως,επιλέγουμε το πάχος της γραμμής (εδώ 2) και καλούμε την drawConnection που περιγράφηκε παραπάνω για να προσθέσουμε τον κόμβο στη λίστα με τους συνδεδεμένους.Στη συνέχεια ποιά είναι η θέση του κόμβου this σε σχέση με την θέση του κόμβου dest.Συγκεκριμένα,εάν βρίσκονται στην ίδια στήλη αλλά ανάμεσα τους υπάρχει άλλος κόμβος τότε η γραμμή που θα ενώνει τους κόμβους θα είναι καμπύλη ξεκινάει από συγκεκριμένο σημείο του κόμβου this (ανάλογα εάν είναι αριστερά ή δεξιά ο κόμβος,έχει επιλεγεί κατάλληλη γωνία που επιστρέφει το επιθυμητό σημείο) και καταλήγει σε τυχαίο σημείο(που όμως βρίσκεται σε συγκεκριμένο τμήμα του κόμβου,ανάλογα εάν είναι δεξιά ή αριστερά).Το σημείο ελέγχου της καμπύλης κατασκευάζεται έτσι ώστε σταδιακά για κάθε καμπύλη που κατασκευάζεται , αυτή να είναι ολοένα και πιο απομακρυσμένη από τους κόμβους(αυτό συμβαίνει για να μην υπάρχει μεγάλη πιθανότητα οι γραμμές να επικαλύπτονται εάν υπάρχουν πολλές που ξεκινηνούν από το ίδιο σημείο και το φροντίζει η στατική μεταβλητή curveStep η οποία κάθε φορά που σχεδιάζεται καμπύλη αυξάνει κατά 20).Εάν οι κόμβοι είναι στην ίδια στήλη αλλά δεν μεσολαβεί άλλος κόμβος ανάμεσά τους,τότε σχεδιάζουμε απλή ευθεία γραμμή που ξεκινάει από το κάτω μέρος του this κόμβου και καταλήγει στο πάνω μέρος του dest κόμβου.Εάν οι κόμβοι έχουν το ίδιο y , δηλαδή είναι στην ίδια γραμμή τότε πάλι σχεδιάζουμε ευθεία γραμμή από σημείο που προκύπτει από angle τέτοιο ώστε να μην υπάρχει επικάλυψη των γραμμών.Εάν δεν έχουν ίδιο x ούτε ίδιο y τότε η σύνδεση θα είναι διαγώνια και πάλι με ευθεία γραμμή.Τέλος,σχεδιάζεται το βέλος στο καταληκτικό σημείο της σύνδεσης(ευθείας ή καμπύλης) με κλήση της drawArrowHead που περιγράφεται ακολούθως.
- ✓ void drawArrowHead(Graphics2D g,Point src,Point dest) : Για να σχεδιάσουμε το βέλος στην άκρη της σύνδεσης,χρειαζόμαστε 3 σημεία.Το ένα σημείο το γνωρίζουμε ήδη και θα είναι το καταληκτικό σημείο της σύνδεσης.Για να βρούμε τα υπόλοιπα σημεία πρέπει να γνωρίζουμε την γωνία που πρέπει να στρέψουμε το βέλος ώστε να συμπίπτει με την γωνία που κατευθύνεται η γραμμή.Για να το κάνουμε αυτό βρίσκουμε το αντίστροφο της εφαπτομένης (συνάρτηση atan) με ορίσματα τα x,y του κόμβου προορισμού και κόμβου κατεύθυνσης.Με αυτά τα δεδομένα και αποφασίζοντας ότι το μήκος του βέλους θα είναι 9 βρίσκουμε τα άλλα 2 σημεία και γεμίζουμε το πολύγωνο που σχηματίζουν αυτά τα 3 σημεία.
- ✓ void updatePre(Graphics g,int pre) και void updatePre(Graphics g,int pre) : μέθοδοι που χρησιμεύουν στην καταγραφή του post και του pre κατά την εκτέλεση του αλγορίθμου DFS.

- ✓ Node getAdjUnvisitedVertex() : Επιστρέφει τον πρώτο γειτονικό κόμβο που δεν έχει επισκεφτεί από τον αλγόριθμο DFS.
- ✓ void resetConnections() : Αδειάζει την ArrayList που περιέχει τους κόμβους με τους οποίους συνδέεται ο συγκεκριμένος κόμβος.
- ✓ static void reset() : Επαναφέρει τις στατικές μεταβλητές στις αρχικές τους τιμές.
- ✓ repaint(Graphics2D g) : Αναθέτει στην μεταβλητή p το κέντρο του κόμβου και σχεδιάζει τον κόμβο με κλήση της drawNode.
- ✓ Λοιπές get και set μέθοδοι που επιστρέφουν και θέτουν μεταβλητές του αντικειμένου.

Κλάση InvalidInputListener

- Σκοπός της κλάσης αυτής είναι να διαχειρίζεται με την μέθοδο keyTyped τα KeyEvent ώστε να αποτρέπει εισαγωγή ψηφιων,κενών και λοιπών ειδικών χαρακτήρων στο Input text field του applet.Εάν ο χρήστης πληκτρολογήσει έναν από τους παραπάνω χαρακτήρες,ο χαρακτήρας απορρίπτεται και παράγεται ένα beep.

Κλάση ValidationInputListener

- Σκοπός της κλάσης αυτής είναι να ελέγχει το input που έχει δώσει ο χρήστης 3 δευτερόλεπτα από την τελευταία φορά που το τροποποίησε.Το event που ελέγχουμε για να πετύχουμε το προηγούμενο είναι ένα DocumentEvent.Το DocumentEvent το διαχειριζόμαστε στις μεθόδους insertUpdate(προσθήκη χαρακτήρων) και removeUpdate(αφαίρεση χαρακτήρων).Η μέθοδος changedUpdate αφορά αλλαγή στη μορφή του κειμένου(γραμματοσειρά,μέγεθος) και γι'αυτό δεν χρησιμοποιείται. Όταν μια από της παραπάνω μεθόδους λάβει ένα DocumentEvent τότε καλεί την scheduleTask η οποία δημιουργεί (εάν έχει ήδη δημιουργηθεί το ακυρώνει) ένα νέο TimerTask που θα αλλάξει την εικόνα του label validation σε 3 δευτερόλεπτα ανάλογα με το αν η είσοδος είναι έγκυρη ή άκυρη.Επίσης υπάρχει και η μέθοδος cancelTask που ακυρώνει το τρέχον TimerTask(χρησιμοποιείται κατά την διαδικασία που προκαλεί το reset button).

Κλάση AppletProperties

- Σκοπός της κλάσης αυτής είναι να φορτώνει τις ρυθμίσεις του Applet από το αρχείο Conf.properties.

Κλάση Configs

- Σκοπός της κλάσης αυτής είναι να αποθηκεύσει τις ρυθμίσεις του αρχείου Conf.properties σε στατικές δημόσιες μεταβλητές για πρόσβαση από οποιαδήποτε κλάση τις χρειάζεται.Η μεταβλητή codeBase αφορά το base URL δηλαδή την τοποθεσία που βρίσκεται ο εκτελέσιμος κώδικας του applet και χρησιμεύει στην εύρεση εικόνων και αρχείων που χρειάζεται το πρόγραμμα για να εκτελεστεί σωστά.Η μεταβλητή αυτή αρχικοποιείται στην κλάση Applet.

Κλάση Applet

- Σκοπός της κλάσης αυτής είναι να δημιουργήσει τα επιμέρους τμήματα που αποτελούν το applet (1 canvas και 3 panel) καθώς και να τα τοποθετήσει σωστά.Για την τοποθέτησή τους χρησιμοποιήθηκε ο BorderLayout manager χωρίς οριζόντια/κάθετα κενά(όρισμα 0,0 κατά την κατασκευή του manager).Τα επιμέρους τμήματα τοποθετήθηκαν πάνω στο applet με κλήση της add και προσδιορισμό του σημείου που πρέπει να τοποθετηθούν(Νότια,Δυτικά κτλπ).Επιπλέον έχουμε την αρχικοποίηση της μεταβλητής Configs.codeBase με κλήση της getCodeBase().Όλα αυτά γίνονται μέσα στην μέθοδο init που εκτελείται μόνο μια φορά όταν ξεκινάει το applet για να τοποθετήσει τα components του.

Κλάση Console

- Η κλάση αυτή είναι ένα JPanel που περιέχει μόνο ένα JTextField για να εμφανίζονται μηνύματα στον χρήστη κατά την εκτέλεση του αλγορίθμου.Για λόγους αισθητικής δεν έχει προστεθεί scollbar όμως υπάρχει η μέθοδος validateLines που ελέγχει έαν έχουμε υπερβεί το όριο των 5 γραμμών.

Εάν ναι τότε διαγράφεται η παλίοτερη γραμμή ώστε να προστεθεί νέα.Η μέθοδος write γράφει το όρισμα της στο textfield(αφού πρώτα καλέσει την validateLines) και μόνο εάν το όρισμα addNewLine είναι true προσθέτει χαρακτήρα '\n' στο τέλος.Το όρισμα text μετατρέπεται σε πίνακα χαρακτήρων και τυπώνεται κάθε χαρακτήρας με καθυστέρηση 70ms ώστε να δίνει την δυνατότητα στον χρήστη να προλαβαίνει να διαβάζει πριν ο αλγόριθμος προχωρήσει σε επόμενα βήματα.

Κλάση InputToolbar

Η κλάση αυτή διαχειρίζεται το input που δίνει ο χρήστης καθώς και βοηθάει τον χρήστη να γράψει αυτό το input ευκολότερα.Υπάρχει ένα JTextField και 3 Jbuttons.Τα Jbuttons όταν πατηθούν γράφουν το σύμβολο που έχουν στο textfield.Επίσης υπάρχει ένα Jlabel που αλλάζει εικόνα σύμφωνα με το εάν το input είναι σωστό ή όχι(το ποιά εικόνα θα είναι σε κάθε περίπτωση το διαχειρίζεται η κλάση ValidationInput Listener που έχει ήδη αναφερθεί η λειτουργία της) ή εάν το textfield είναι «κλειδωμένο»(δεν μπορεί να επεξεργαστεί).Δεν χρησιμοποιείται κάποιος Layout Manager και τα components λαμβάνουν τις θέσεις που ορίζονται στην add με την σειρά που προστίθενται.

Κλάση Options

Η κλάση αυτή είναι υπεύθυνη για την εκκίνηση των 3 βασικών λειτουργιών του applet (εκκίνηση αλγορίθμου με το input του χρήστη,reset για να εισαχθεί νέο input αφού ο αλγόριθμος έχει τελειώσει την λειτουργία του και εκτέλεση του αλγορίθμου με τυχαίο input).Το κουμπί start ξεκινάει τον αλγόριθμο(εάν είναι σωστή η είσοδος του χρήστη) σε νέο νήμα και καλεί την μέθοδο start της κλάσης GraphCanvas αφού πρώτα «κλειδώσει» όλα τα κουμπιά και το textfield του applet.Ο λόγος που χρησιμοποιείται νέο νήμα είναι για να μην επιβαρυνθεί το νήμα του γραφικού περιβάλλοντος με την αρχικοποίηση των δομών του αλγορίθμου αλλά και άλλων εργασιών που πραγματοποιούνται.Το κουμπί reset γίνεται ενεργό όταν ο αλγόριθμος τελειώσει την εργασία του και καθαρίζει όλα τα δεδομένα αλλά και τον GraphCanvas για επόμενη χρήση.Το κουμπί random εκτελεί ακριβώς την ίδια διαδικασία με το κουμπί start μόνο που εδώ το input ορίζεται τυχαία από την κλάση Regex.

Κλάση GraphCanvas

Η κλάση GraphCanvas έχει τις εξής μεταβλητές:

- Instance fields:

✓ ArrayList<Node> nodes : Δομή ArrayList<Node> που θα περιέχει όλους τους κόμβους που θα σχεδιαστούν.

✓ ArrayList<ArrayList<Node>> scc: Δομή ArrayList<ArrayList<Node>>που θα περιέχει σύνολα από κόμβους που ανήκουν στην ίδια ισχυρά συνεκτική συνιστώσα.

- Methods :

✓ void doRepaint() : Καλεί την μέθοδο paint για να σχεδιαστεί το κατάλληλο βήμα του αλγορίθμου.

✓ void start(String input) : Είναι η βασική μέθοδος της κλάσης που καλεί όλες τις υπόλοιπες ώστε να επιλύσει το πρόβλημα.Αρχικά καλεί την init.Έπειτα εάν η είσοδος περιλαμβάνει περισσότερους από 8 κόμβους τότε η λύση εμφανίζεται απευθείας αλλιώς εμφανίζεται παράλληλα με τα γραφικά.Τελικά καλείται η isSat που θα αποφανθεί εάν η πρόταση είναι ικανοποιήσιμη ή όχι.Εάν είναι θα γίνει τακτοποίηση λεξικογραφικά των κόμβων και θα κληθεί η satValue.

✓ sleep(int ms) : Καλεί την sleep του τρέχοντος νήματος.

✓ void paint(Graphics g) : Ανάλογα με τον αριθμό βήματος του αλγορίθμου,καλούνται οι αντίστοιχες συναρτήσεις.

✓ void init(String input) : Επεξεργάζεται την πρόταση της εισόδου και προσθέτει τα μοναδικά literals ως κόμβους στην δομή nodes.

✓ void repaintGraph() : Σχεδιάζει το γράφημα από τους κόμβους του γραφήματος ξανά(χρησιμεύει όταν αλλάζουν οι κατευθύνσεις των ακμών).

✓ void erase() : Καλεί την Nodes.reset() και σβήνει ό,τι έχει σχεδιαστεί.

✓ void clear() : Καλεί την erase() και αδειάζει όσες δομές χρησιμοποιήθηκαν στον αλγόριθμο ώστε να χρησιμοποιηθούν σε επόμενη κλήση.

- ✓ void addConections(String input,boolean reversed,boolean ignoreSleep) : Επεξεργάζεται το input και για κάθε ακμή x,y κατασκευάζει 2 ακμές(με κλήση της αντίστοιχης drawConnection της κλάσης Node,εάν θέλουμε γραφική επίλυση ή οχι) σύμφωνα με τον αλγόριθμο.Το όρισμα reversed είναι για να κατασκευάσει αυτές τις ακμές με την αντίστροφη φορά(χρήσιμο στην αντιστροφή του γραφήματος) και το όρισμα ignoreSleep είναι για να μην κληθεί η sleep στο τρέχον νήμα ώστε η σχεδίαση των συνδέσμων να γίνει αμέσως και όχι διαδοχικά.
- ✓ void graphSol() : Καλείται σε περίπτωση που το μέγεθος του input επιτρέπει σχεδιαστική επίλυση.Αυξάνει την μεταβλητή step και καλεί την doRepaint() έχοντας ενδιάμεσα τα κατάλληλα sleep ώστε να εμφανίζεται το αποτέλεσμα καλύτερα στον χρήστη και να προλαβαίνει να το δει.
- ✓ void instantSol() : Καλείται σε περίπτωση που το μέγεθος του input δεν επιτρέπει σχεδιαστική επίλυση και εκτελεί ακριβώς τα ίδια βήματα με την graphSol() μόνο που δεν σχεδιάζει το γράφημα.
- ✓ getIndexOf(String str,ArrayList<Node> al) : Επιστρέφει το index που έχει ο κόμβος με όνομα str στην ArrayList<Node> al.Εάν δεν υπάρχει τέτοιος κόμβος επιστρέφει το μέγεθος του al.
- ✓ int DFS(Node root,boolean ignoreOrder,Stack<Node> postOrder,int prevClock) : Εκτελεί τον αλγόριθμο DFS.Ο αλγόριθμος ξεκινάει από τον κόμβο root.Υπάρχει μεταβλητή clock που δείχνει σε ποιό βήμα βρισκόμαστε ώστε να κρατάμε τα post και pre των κόμβων.Υπάρχει όρισμα ignoreOrder που εάν είναι true τότε διατηρείται φθίνουσα σειρά των post στην στοίβα postOrder.Επιπλέον,υπάρχει το όρισμα prevClock που εάν δεν είναι 0 τότε η μεταβλητή clock ξεκινάει από το prevClock και όχι από το 0.
- ✓ getUnvisitedVertex(ArrayList<Node> list) : Επιστρέφει έναν κόμβο από την λίστα nodes που δεν έχει επισκεφτεί ο αλγόριθμος DFS.
- ✓ void SCC(Stack<Node> stack,ArrayList<ArrayList<Node>> SCC) : Εύρεση ισχυρά συνεκτικών συντιστωσών σύμφωνα με τον γνωστό από το μάθημα αλγόριθμο.
- ✓ String satValue(String input) : Αντικαθιστά το λογικό OR με κενό και τους όρους στο input με 0 και 1 (ανάλογα εάν είναι άρνηση ή όχι) και ελέγχει εάν αυτό που προκύπτει είναι της μορφής [(01) ή (10) ή (11)] μία ή περισσότερες φορές.Εάν ναι βρέθηκε αποτίμηση που ικανοποιεί την πρόταση.Αλλιώς συνεχίζει με τον επόμενο αριθμό που μετατρέπει σε δυαδικό.Αυτό επαναλαμβάνεται 2^(nodes.size()/2) και πρέπει σίγουρα να επιστρέψει αποτίμηση όταν η πρόταση είναι ικανοποιήσιμη.
- ✓ String increase(int n,int size) : Επιστρέφει την δυαδική αναπαράσταση μήκους size-bits του αριθμού n.
- ✓ boolean isSat() : Ελέγχει εάν υπάρχει ένα όρος και η άρνησή του στην ίδια ισχυρά συνεκτική συνιστώσα.Εάν συμβαίνει αυτό επιστρέφει false και η πρόταση δεν είναι ικανοποιήσιμη.Αλλιώς επιστρέφει true.