Τεχνητή νοημοσύνη

Ακαδημαϊκό έτος: 2019–20

Κουρλός Κωνσταντίνος 3170077

Λαζελλάρι Αλέξης 3170088

Μαρκάκη Αικατερίνη Κωνσταντίνα 3170102

1η Εργασία: Πρόγραμμα μαθημάτων γυμνασίου

Για την επίλυση της εύρεσης προγράμματος μαθημάτων ενός γυμνασίου ακολουθήσαμε την παρακάτω διαδικασία σύμφωνα με την εκφώνηση που μας δόθηκε:

* Δημιουργήσαμε 2 αρχεία απλού κειμένου (.txt). Ένα με όνομα Lessons, για τα μαθήματα που διδάσκονται, και ένα με όνομα Teachers, για τους καθηγητές που διδάσκουν στο Γυμνάσιο, σύμφωνα με τις προδιαγραφές της εκφώνησης.
* Θεωρήσαμε ότι έχουμε 5 διδακτικές μέρες με 7 διδακτικές ώρες η κάθε μια. Η κάθε τάξη είναι χωρισμένη σε 3 τμήματα και τα μαθήματά τους είναι βασισμένα σε πραγματικό πρόγραμμα γυμνασίου και στο σύνολο τους για κάθε τμήμα είναι 32.
* Έπειτα με αντίστοιχες συναρτήσεις, της κλάσης fileRW που φτιάξαμε, διαβάζουμε τα δύο αρχεία .txt και δημιουργούμε αντικείμενα αντίστοιχων κλάσεων(με αντίστοιχες μεταβλητές) και τα περνάμε σε μία κλάση Data, η οποία κρατάει αποθηκευμένα όλα τα δεδομένα του προγράμματος σε πίνακες.
* Στην κλάση Data, εκτός από τους πίνακες με τα δεδομένα έχουμε υλοποιήσει
  + μια λίστα whoMakes, η οποία περιέχει για κάθε μάθημα λίστα με τους καθηγητές που μπορούν να το διδάξουν
  + μια συνάρτηση initializerTotalHours(), η οποία βρίσκει για κάθε τμήμα τις συνολικές του ώρες διδασκαλίας ανά εβδομάδα
* Στη συνέχεια, δημιουργούμε μία κατάσταση State με τα δεδομένα των μαθημάτων και των καθηγητών. Συγκεκριμένα κάθε κατάσταση State έχει ένα πίνακα δύο διαστάσεων με το αριθμό των max ωρών διδασκαλίας της εβδομάδας για στήλες και τον αριθμό των καθηγητών για γραμμές. Ο πίνακας αυτός σε κάθε κελί του περιέχει ένα τμήμα (πχ Α3) και ένα κωδικό του μαθήματος (πχ Α110). Γίνεται τυχαία αρχικοποίηση για κάθε μάθημα σε οποιαδήποτε ώρα διδασκαλίας και επιλέγεται εξίσου τυχαία ποιος καθηγητής από τους αρμόδιους θα το διδάξει.
* Στο τέλος της τυχαίας αρχικοποίησης ενός προγράμματος δημιουργούμε ένα νήμα στο οποίο αναθέτουμε την αξιολόγηση του προγράμματος αυτού με βάση τους περιορισμούς που πρέπει να ικανοποιεί. Για την χρήση του νήματος δημιουργήσαμε μια κλάση Restrictions, η οποία υλοποιεί μια μέθοδο run με τους περιορισμούς. Όταν κάποιος περιορισμός παραβιάζεται αυξάνουμε έναν μετρητή errorCounter με την ανάλογη βαρύτητα. Πιο συγκεκριμένα, έχουμε υλοποιήσει τους εξής περιορισμούς:
  + Να μην υπάρχουν κενά στο πρόγραμμα κάθε τμήματος και ο ημερήσιος αριθμός ωρών να είναι κατά το δυνατόν ομοιόμορφος όλες τις ημέρες, περιορισμοί 1 και 3 της εκφώνηση. Για κάθε τμήμα αρχικοποιούμε μια μεταβλητή max hours per day (mhpd) με τον μέσο όρο των ωρών του ανά μέρα (εβδομαδιαίες ώρες τμήματος δια αριθμό σχολικών ημερών) και ελέγχουμε αν δεν διαιρούνται ακριβώς με τις σχολικές ημέρες. Στη συνέχεια, διατρέχουμε τον πίνακα του προγράμματος για κάθε ημέρα και ώρα και ελέγχουμε αν το τμήμα αυτό έχει μάθημα κρατώντας ένα άθροισμα(statehpd). Αν το άθροισμα αυτό δεν είναι ίδιο με το mhpd τότε αυξάνουμε το errorCounter του προγράμματος αυτού κατά 2. Έπειτα για την συγκεκριμένη μέρα και για κάθε μια από τις ώρες που βρήκαμε ελέγχουμε από το τέλος προς την αρχή αν υπάρχει κάποιος καθηγητής που να του αντιστοιχεί το τμήμα αυτό. Αν όχι, προσθέτουμε στον το errorCounter του προγράμματος το 5.
  + Να μη διδάσκει κανένας καθηγητής περισσότερες από δύο συνεχόμενες ώρες. Κρατώντας ένα άθροισμα ανά γραμμή που αντιστοιχεί στους καθηγητές, διατρέχουμε τον πίνακα του προγράμματος και ελέγχουμε αν το πεδίο του τμήματος που μπορεί να διδάξει είναι κενό. Στην περίπτωση που δεν είναι ενημερώνουμε το άθροισμα, ενώ αν είναι κενό μηδενίζουμε το άθροισμα. Αν το άθροισμα φτάσει το 3 τότε ο έλεγχος διακόπτεται και το errorCounter του προγράμματος αυξάνεται κατά 1.
  + Οι ώρες διδασκαλίας κάθε μαθήματος σε ένα τμήμα να είναι κατά το δυνατόν ομοιόμορφα κατανεμημένες σε όλες τις ημέρες της εβδομάδας. Αρχικά ελέγχουμε αν το μάθημα έχει περισσότερες από 1 ώρες διδασκαλίας για να έχει νόημα ο περιορισμός. Έπειτα, χρησιμοποιούμε μία λογική μεταβλητή flag και ελέγχουμε αν ο αριθμός ωρών διδασκαλίας του μαθήματος δία τις διδακτικές μέρες (mhpd) είναι ακέραιο πολλαπλάσιο του αριθμού των ημερών διδασκαλίας (5). Διασχίζουμε τον πίνακα του προγράμματος για κάθε μέρα, αρχικοποιώντας κάθε φορά τη μεταβλητή mhpd, και για κάθε ώρα. Ελέγχουμε αν το κάθε στοιχείο του είναι το μάθημα του οποίου εξετάζουμε την εβδομαδιαία κατανομή. Αν το συναντήσουμε κατά την αναζήτηση αυξάνουμε αντίστοιχα αθροίσματα του μαθήματος για ημερήσιο (statehpd) και εβδομαδιαίο/συνολικό αριθμό ωρών (totalHours). Μετά από κάθε 7 ώρες (1 διδακτική ημέρα) ελέγχουμε αν αληθεύει η λογική μεταβλητή flag και συγκρίνουμε την ημερήσια τιμή των ωρών διδασκαλίας που βρήκαμε αθροίζοντας από την αναζήτηση και την τιμή που υπολογίζαμε αρχικά πως αναλογεί στην κάθε μέρα της εβδομάδας (mhpd). **Αν η flag αληθεύει και οι δύο άλλες μεταβλητές (statehpd και mhpd) δεν είναι μεταξύ τους ίσες, αυξάνουμε τον errorCounter κατά 1. Αν η flag δεν αληθεύει και πάλι οι τιμές των δύο μεταβλητών είναι άνισες αυξάνουμε ξανά τον errorCounter κατά 1.** Αν έχουμε φτάσει τον συνολικό αριθμό ωρών του μαθήματος ανά εβδομάδα τότε η διάσχιση του πίνακα σταματά.
  + Ο αριθμός ωρών διδασκαλίας ανά εβδομάδα να είναι κατά το δυνατόν ομοιόμορφος για όλους τους καθηγητές. Για κάθε καθηγητή βρίσκουμε τον συνολικό αριθμό ωρών διδασκαλίας του για κάθε μάθημα, αρχικοποιώντας μία δομή List η οποία περιέχει List με Strings. Διατρέχουμε ολόκληρο τον πίνακα του προγράμματος και για κάθε γραμμή, δηλαδή για κάθε καθηγητή αρχικοποιούμε σε μεταβλητές τις μαξ διδακτικές ώρες του ανά ημέρα (max\_hpd) και ανά εβδομάδα (max\_hpw), και σε μία δομή List τα μαθήματα που διδάσκει. Για κάθε μία ώρα του εβδομαδιαίου προγράμματος, αν διδάσκει μάθημα, προσθέτουμε το μάθημα στην λίστα με τα διδασκόμενα μαθήματά του και μειώνουμε τις μεταβλητές max\_hpd και max\_hpw αντίστοιχα. Αν ξεκινάει καινούρια διδακτική μέρα θέτουμε ξανά στην max\_hpd την αρχική της τιμή και αν η max\_hpd λάβει αρνητικές τιμές αυξάνουμε τον errorCounter κατά 1. Ο τρόπος με τον οποίο ελέγχεται εάν υπάρχουν ανισότητες μεταξύ του αθροίσματος ωρών διδασκαλίας ανά καθηγητή είναι με max/min τιμές. Κάθε καθηγητής έχει ένα σύνολο ωρών που διδάσκει .Εάν υπάρχει μεγάλη διαφορά ανάμεσα στον καθηγητή που διδάσκει τις περισσότερες ώρες με αυτόν που διδάσκει τις λιγότερες, τότε αυξάνουμε τον errorCounter.
  + Να μην έχουν δύο ή περισσότεροι καθηγητές την ίδια ώρα μάθημα στο ίδιο τμήμα. Αρχικοποιήσαμε μια δομή ArrayList (samehours) που θα δέχεται τα τμήματα. Για κάθε ώρα του προγράμματος διατρέχουμε όλες τις γραμμές των καθηγητών και αν το κελί του πίνακα δεν είναι άδειο το προσθέτουμε στο ArrayList. Αν το τμήμα αυτό υπάρχει ήδη στη λίστα, δηλαδή το έχουμε ξανασυναντήσει στην ίδια στήλη που αντιστοιχεί σε μία σχολική ώρα, ο κανόνας παραβιάζεται και αυξάνουμε τον errorCounter κατά 30.
  + Να διδάσκονται όλες οι διδακτικές ώρες ενός μαθήματος από έναν μόνο καθηγητή. Δημιουργήσαμε μία δομή ArrayList (whoMakes) στην οποία αποθηκεύουμε για κάθε μάθημα**, μία επιμέρους λίστα με τους καθηγητές που μπορούν να το διδάξουν,** καλώντας την αντίστοιχη μέθοδο που υλοποιήσαμε στην κλάση των δεδομένων (Data.whoMakes()). Έπειτα για κάθε ένα μάθημα ελέγχουμε αν η δομή whoMakes περιέχει περισσότερους από έναν καθηγητές ώστε να έχει νόημα ο έλεγχος του περιορισμού. Αν ναι, για κάθε έναν καθηγητή που μπορεί να διδάξει το συγκεκριμένο μάθημα και για κάθε διδακτική ώρα της εβδομάδας εξετάζουμε αν το μάθημα διδάσκεται από αυτόν, αυξάνοντας αντίστοιχο άθροισμα. Αφού εξεταστούν όλες οι ώρες, ελέγχουμε αν το άθροισμα που βρήκαμε είναι διάφορο του μηδενός και μικρότερο από τις υποχρεωτικές ώρες διδασκαλίας του μαθήματος ανά εβδομάδα και αυξάνουμε τον errorCounter κατά 1.
* Αφού η Restrictions αξιολογήσει το πρόγραμμα που της δώθηκε επιστρέφει τον errorCounter. **Αν δεν έχει προλάβει να δημιουργηθεί άλλο thread ενώ το τελευταίο έχει ήδη αξιολογηθεί περιμένει.**
* Ακόμα, χρησιμοποιούμε μία μέθοδο, την getChildren, η οποία αλλάζει τυχαία κάποια στοιχεία του προγράμματος State. Πιο συγκεκριμένα, η μέθοδος αποτελείται από δύο σκέλη, εκ των οποίων επιλέγεται κάθε φορά πιο θα εκτελεστεί βάσει πιθανοτήτων. Με πιθανότητα 20% και αν ο τελευταίος κανόνας που περιγράψαμε παραβιάζεται, **εκτελείται το πρώτο κομμάτι της μεθόδου**. Σε αυτό, επιλέγεται τυχαία ένας καθηγητής και μία ώρα από το ωρολόγιο πρόγραμμα στην οποία να διδάσκει κάποιο μάθημα σε κάποιο τμήμα. ‘Έπειτα, ελέγχουμε αν στο πρόγραμμα διδάσκονται όλες οι υποχρεωτικές ώρες του μαθήματος που έχουμε επιλέξει. Αν αυτό ισχύει, επαναλαμβάνουμε την διαδικασία εκ νέου, αλλιώς, σταματάμε. Στην συνέχεια, ελέγχουμε από ολόκληρο το ωρολόγιο πρόγραμμα ποιος καθηγητής διδάσκει τις υπόλοιπες υποχρεωτικές ώρες ,όταν ο αλγόριθμος εντοπίσει το συγκεκριμένο μάθημα που διεξάγεται σε συγκεκριμένο τμήμα ,οι ώρες μαθήματος αντικαθίστανται ,ως αποτέλεσμα μόνο ένα καθηγητής να διδάσκει το συγκεκριμένο μάθημα στο τμήμα. Με πιθανότητα 80% εκτελείται το **δεύτερο κομμάτι της μεθόδου.** Επιλέγουμε τυχαία δύο καθηγητές και δύο ώρες διδασκαλίας**, στις οποίες να διδάσκεται κάποιο μάθημα.** Αν στην αντίστοιχη διδακτική ώρα του πρώτου καθηγητή, διδάσκει κάποιο μάθημα τότε επιλέγουμε τυχαία κάποιον δεύτερο καθηγητή που να μπορεί, βάσει ειδικότητας, να διδάξει το μάθημα αυτό. αλλάζουμε αυτή την συγκεκριμένη ώρα του πρώτου καθηγητή με μία ώρα του δεύτερου καθηγητή.
* Όσον αφορά τις μεθόδου τεχνητής νοημοσύνης, χρησιμοποιήσαμε την μέθοδο της **αναρρίχησης λόφων (hill climbing)** την οποία υλοποιήσαμε στην αντίστοιχη κλάση. Ακολουθήσαμε τα βήματα των διαφανειών της 4ης διάλεξης (20η διαφάνεια).
* Το πρόγραμμά μας δέχεται τα μονοπάτια για τα προς εξέταση txt αρχεία των μαθημάτων και των καθηγητών ως όρισμα από τη γραμμή εντολών.
* Το πρόγραμμά μας παράγει ένα αρχείο απλού κειμένου schedule (txt) που παριστάνει το εβδομαδιαίο ωρολόγιο πρόγραμμα όλου του γυμνασίου με τις αντίστοιχες συναρτήσεις, της κλάσης fileRW. Πιο αναλυτικά σε αυτό αναγράφονται, οι μέρες και οι διδακτικές ώρες της εβδομάδας ως ονόματα στηλών και τα ονόματα τον καθηγητών ως ονόματα γραμμών. Ο πίνακας χωρίζεται σε κουτιά, κάθε ένα εκ των οποίων αντιστοιχεί σε έναν καθηγητή και μια ώρα διδασκαλίας. Μέσα στα κουτιά αυτά υπάρχει ένα τμήμα και ο κωδικός ενός μαθήματος ή “---” που δηλώνει ότι η εκείνη η ώρα διδασκαλίας είναι κενή και δεν διδάσκεται κανένα μάθημα από τον συγκεκριμένο καθηγητή.
* Πειραματικό αποτέλεσμα του προγράμματός μας

