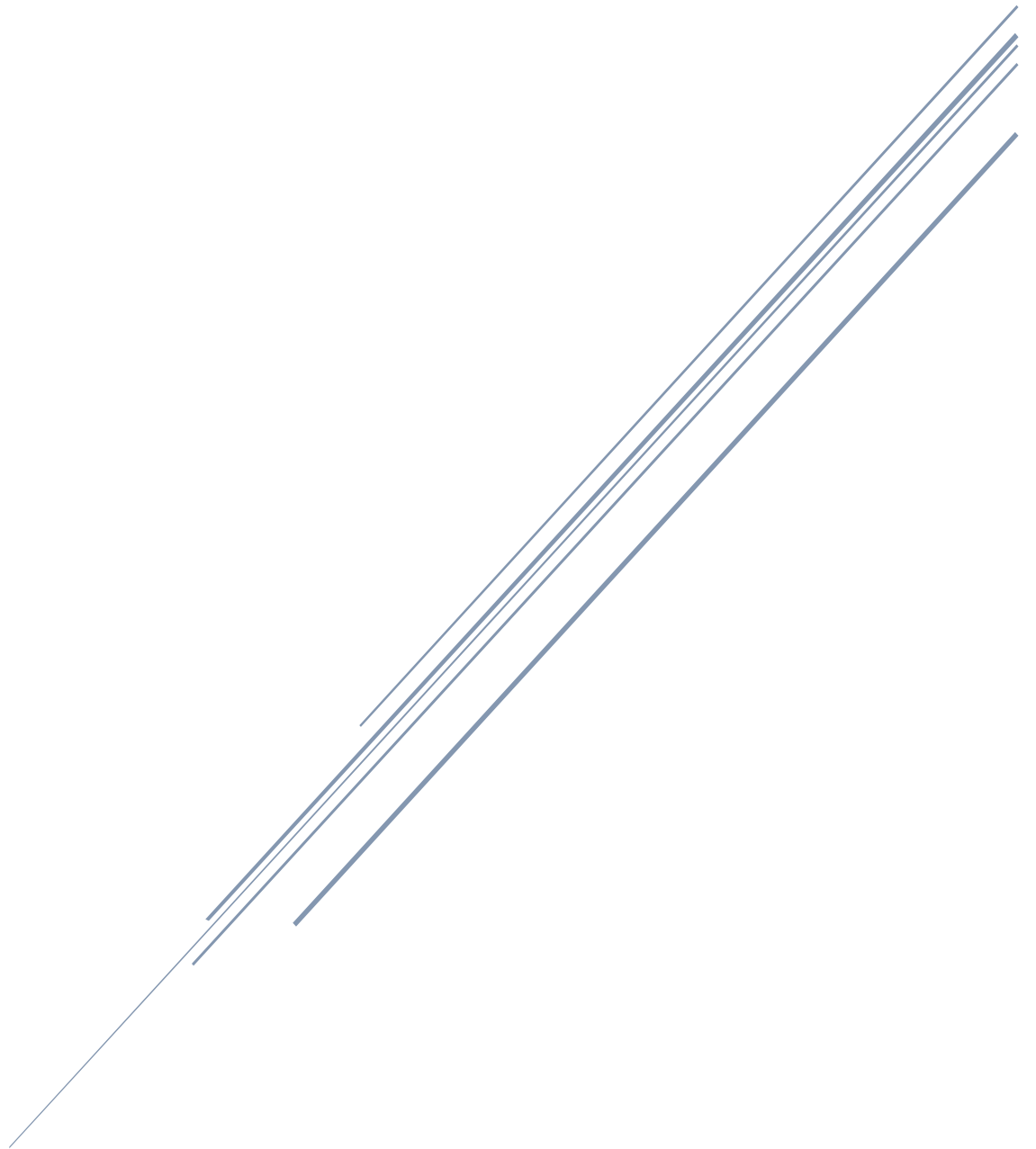


ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΩΝ

2η Εργαστηριακή Άσκηση



Μπουραΐμης Φώτης 2013030067 LAB21143241

01/12/2019

Πρόλογος:

Στη δεύτερη εργαστηριακή άσκηση του μαθήματος είχαμε να διαχειριστούμε ένα αρχείο από αποδείξεις και να το αποθηκεύσουμε σε κατάλληλες δομές δεδομένων έτσι ώστε να μπορούμε να το διαθέσουμε στο χρήστη γρήγορα και αποτελεσματικά.

A) Πώς διαβάσετε κάθε απόδειξη;

Αρχικά, σαν πρώτο πρόβλημα είχαμε να αντιμετωπίσουμε το διάβασμα του αρχείου το οποίο είναι άρρηκτα συνυφασμένο με τον τρόπο που ελέγχεται η απόδειξη ως προς την εγκυρότητά της. Για το διάβασμα του αρχείου χρησιμοποίησα μια while loop η οποία ελέγχει σε κάθε της επανάληψη την τιμή του κέρσορα (δλδ. τη θέση που βρισκόμαστε στο διάβασμα του αρχείου) και όταν η τιμή αυτή είναι ίση με το μέγεθος του αρχείου σημαίνει ότι έχουμε διαβάσει όλα τα δεδομένα. Ο λόγος που επιλέχτηκε η while και όχι η for είναι επειδή στο σώμα της επανάληψης η τιμή του κέρσορα αυξάνεται αυθαίρετα και όχι μόνο στην αρχή ή στο τέλος της επανάληψης. Για τον έλεγχο των αποδείξεων χρησιμοποίησα εμφωλευμένες if η οποίες ελέγχουν την κάθε γραμμή και όταν δεν βρίσκουν αυτό που “περιμένουν” στην επόμενη γραμμή σταματάνε και πηγαίνουν αμέσως στην πρώτη γραμμή που περιέχει παύλες (δλδ. στην αρχή της επόμενης απόδειξης). Αυτό σημαίνει ότι κατά τη διάρκεια της επανάληψης δεν γίνονται περιττοί έλεγχοι ή περιττές εκχωρήσεις στοιχείων που αργότερα θα διαγραφούν.

B) Πώς ελέγχετε αν η απόδειξη είναι σωστή;

Όσον αφορά την εγκυρότητα της κάθε γραμμής χρησιμοποίησα regular expressions τις οποίες τις χώρισα σύμφωνα με τα τέσσερα αποδεκτά είδη γραμμών που περιμένω, παύλες, γραμμή αφμ, γραμμή προϊόντων και γραμμή συνόλου. Με αυτόν τον τρόπο εάν έβρισκα στην πορεία κάποιο στοιχείο που δεν επιτρεπόταν άλλα το πρόγραμμα το δεχόταν ως έγκυρο, απλά άλλαζα την εκάστοτε regular expression.

Γ) Τι δομές δεδομένων χρησιμοποιήσατε για να απαντάτε γρήγορα τα ερωτήματα του χρήστη και γιατί τις επιλέξατε;

Προχωρώντας στο επόμενο ζητούμενο, για την αποθήκευση και τη διάθεση δεδομένων επέλεξα τα dictionaries. Κύρια πηγή γι' αυτή την επιλογή ήταν η σχέση μεταξύ των δεδομένων αφού όλα παρουσιάζονται ως μία ιεραρχία key-value. Πιο συγκεκριμένα οι τιμές των προϊόντων είναι τα values και τα ονόματα των προϊόντων είναι τα keys, ενώ το σύνολο αυτών είναι τα values για το κάθε αφμ. Η μορφή λοιπόν είναι nested dictionary για κάθε απόδειξη, φυσικά αποδείξεις με ίδια αφμ συνυπολογίζονται όπως και προϊόντα με ίδιο όνομα στην κάθε απόδειξη. Τα dictionaries υπερέχουν των άλλων δομών στην αναζήτηση όταν έχουμε key-values μορφή δεδομένων, εν προκειμένω με την επιλογή 2 θέλουμε τα αφμ ενός προϊόντος ενώ με την 3 θέλουμε όλα τα προϊόντα ενός αφμ. Ωστόσο, ένα μειονέκτημα που παρουσιάζεται είναι ότι δεν μπορούμε να αποθηκεύσουμε τα δεδομένα μας ταξινομημένα, παρά μόνο να τα εκτυπώνουμε ταξινομημένα όπως και γίνεται μέσα στο πρόγραμμα.

Δ) Πώς εξασφαλίζετε ότι μπορείτε να διαβάσετε πάρα πολύ μεγάλα αρχεία;

Τέλος εξασφάλισα ότι μπορώ να διαβάζω πολύ μεγάλα αρχεία αφού τα διαβάζω γραμμή-γραμμή και δεν δεσμεύομαι πρώτα στην αποθήκευση ολόκληρου του αρχείου.

Επίλογος:

Ένας επιπλέον λόγος που επέλεξα dictionary είναι επειδή η αποθήκευση των αποδείξεων με διπλά ονόματα προϊόντος ή αρχείο με διπλά αφμ γίνεται σε real time και δε χρειάζεται αφού τα αποθηκεύσω να χρησιμοποιώ συναρτήσεις που να κάνουν αυτή τη δουλειά. Αυτό έχει ως αποτέλεσμα να αργεί λίγο παραπάνω να διαβάσει το αρχείο αλλά όταν ολοκληρωθεί το διάβασμα δεν έχω περαιτέρω διεργασίες.