Εργαλεία Ανάπτυξης Λογισμικού και Προγραμματισμός Συστημάτων

Αναφορά Άσκησης 1

Ομάδα Εργασίας: LAB21142642
Μουδήρη Κωνσταντίνα 2012030128
Μπαρτς Μάνουελ Ραφαελ 2013030098
Υλοποιήσαμε απο κοινού όλη την άσκηση

Θέμα 1

Στο πρώτο κομμάτι της άσκησης μας ζητήθηκε να γράψουμε ένα πρόγραμμα regr σε Bash shell που να δέχεται ως παραμέτρους προγράμματος μία λίστα από αρχεία. Κάθε αρχείο έχει γραμμές της μορφής num1:num2.

Πρέπει να θεωρήσουμε τις τιμές της πρώτης στήλης ως ένα διάνυσμα X, και τις τιμές της δεύτερης στήλης ως ένα διάνυσμα Y. Για να το πετύχουμε αυτό, αρχικά θεωρήσαμε κάθε γραμμή του αρχείου ως ένα στοιχείο του πίνακα array και στην συνέχεια χωρίσαμε την κάθε γραμμή σε δύο στοιχεία num1 και num2 (my_array[0] & my_array[1]) και τα τοποθετήσαμε στο διάνυσμα X και διάνυσμα Y αντίστοιχα. Ένα Loop while read line μας βοήθησε ώστε να γνωρίζουμε πόσες γραμμές έχει το κάθε αρχείο. Και η εντολή
IFS=':' read -ra my_array <<< "$my_string" μας χώρισε το my_string (στοιχείο πίνακα array) σε my_array[0] και my_array[1] ανάλογα με το που έβλεπε ":".

Τελειώνοντας με τα παραπάνω έχουμε δύο διανύσματα X και Y τα οποία έχουν n στοιχεία (n=γραμμές του αρχείου). Με βάση λοιπόν αυτό μας ζητήθηκε να υπολογίσουμε τις παραμέτρους γραμμικής παλινδρόμησης a, b, c που ελαχιστοποιούν το συνολικό τετραγωνικό σφάλμα err της προσέγγισης cY=aX+b.

Αρχικά υπολογίσαμε το sum_x, sum_y, sum_xy και sum_x2 με βάση τους τύπους που μας δόθηκαν στην εκφώνηση.

$$sum\_x = \sum_{i=0}^{length-1} X[i] \qquad sum\_y = \sum_{i=0}^{length-1} Y[i] \qquad sum\_xy = \sum_{i=0}^{length-1} X[i] \cdot Y[i]$$

$$sum\_x2 = \sum_{i=0}^{length-1} (X[i])^2$$

Εν συνεχεία, πήραμε τις παρακάτω δύο περιπτώσεις:

- Το διάνυσμα X δεν είναι σταθερό. Επομένως χρησιμοποιούμε τους τύπους που μας δόθηκαν για να υπολογίσουμε το a, b και err και θεωρήσαμε c=1 σε όλες τις περιπτώσεις.

$a = \frac{length \times sum\_xy - sum\_x \times sum\_y}{length \times sum\_x2 - sum\_x \times sum\_x}$	$b = \frac{sum\_y - a \times sum\_x}{length}$
$err = \sum_{i=0}^{length-1} (Y[i] - (aX[i] + b))^2$	

- Το διάνυσμα X είναι σταθερό. Τώρα δεν μπορούμε να χρησιμοποιήσουμε τους παραπάνω τύπους καθώς ο παρονομαστής του α γίνεται 0 αλλά με σταθερό X γνωρίζουμε ότι η ευθεία που θα πάρουμε θα είναι της μορφής X=-b επομένως με βάση τον τύπο cY=aX+b προκύπτει c=0, a=1 και b=-X[0]

Ενδεικτικά βλέπουμε στην συνέχεια κάποια αρχεία του τρέξαμε με τα αποτελέσματα που πήραμε για να ελέγξουμε την αποτελεσματικότητα του κώδικά μας.

File1:

```
1:0
1.3:2
2:5.39
4:19
```

file2:

```
1:0
1:2
1:5.39
1:19
```

```
konstantina@konstantina:~/Documents$ bash regr file1 file2
FILE: file1, a=6.33 b=-6.53 c=1 err=.71
FILE: file2, a=1 b=-1 c=0 err=0
```

Θέμα 2

```
while read line; do
        #echo "Line No. $n : $line"
        array[$n]="$line"
        my_string=${array[$n]}
        IFS=':' read -ra my_array <<< "$my_string"
        teams[$n]=${my_array[0]}
        score[$n]=${my_array[1]}
        IFS='-' read -ra team_array <<< "${teams[$n]}"
        IFS='-' read -ra score_array <<< "${score[$n]}"
        team1[$n]=${team_array[0]}
        team2[$n]=${team_array[1]}
        score1[$n]=${score_array[0]}
        score2[$n]=${score_array[1]}
        let "n=n+1"
done < $1
```

Το αρχείο που δίνεται έχει γραμμές της μορφής Ομάδα1-Ομάδα2:Σκορ1-Σκορ2.Όπως φαίνεται στον παραπάνω κώδικα αρχικά διαβάζουμε το αρχείο γραμμή γραμμή και για κάθε γραμμή κάνουμε το εξής: Διαβάζουμε τη γραμμή και την αποθηκεύουμε σε ένα string και μετά σπάμε το string ως προς το delimiter ":" και παίρνουμε Ομάδα1-Ομάδα2 στο στοιχείο my_array[0] και Σκορ1-Σκορ2 στο στοιχείο my_array[1].Έπειτα σπάμε αυτά τα δύο ως προς το delimiter "-" για να πάρουμε team1[n]=Ομάδα1, team2[n]=Ομάδα2, score1[n]=Σκορ1,score2[n]=Σκορ2 ,όπου n είναι ο αριθμός της γραμμής.

Στη συνέχεια κάνουμε τα εξής:

1. Φτιάχνουμε ένα πίνακα uteam που περιέχει όλες τις διαφορετικές ομάδες μια φορά. Αυτο το πετυχαίνουμε γεμίζοντας αρχικά τις πρώτες 2 θέσεις του πίνακα uteam με τις 2 ομάδες της πρώτης γραμμής του αρχείου και έπειτα για τις ομάδες απο τη δεύτερη μέχρι τη τελευταία γραμμή ελέγχουμε αν μια ομάδα υπάρχει στον πίνακα uteam. Αν δεν υπάρχει τη προσθέτουμε. Επειδή έχουμε απο πριν αποθηκεύσει τις ομάδες απο κάθε γραμμή στους πίνακες team1,team2 ,διατρέχουμε τους πίνακες αυτούς.
2. Φτιάχνουμε τους πίνακες points,goals1,goals2(ίδιου μεγέθους με τον πίνακα uteam) και τους αρχικοποιούμε με 0 σε κάθε θέση ,όπου ο points είναι για τους βαθμούς κάθε ομάδας ο goals1 για τα γκολ που έβαλε κάθε ομάδα και ο goals2 για

τα γκολ που δέχτηκε κάθε ομάδα.Για να τους γεμίσουμε σωστά διατρέχουμε τα δεδομένα που έχουμε απο κάθε γραμμή του αρχείου στους πίνακες team1,team2,score1,score2 . Συγκεκριμένα για κάθε καταχώρηση k(δεδομένα απο τη γραμμή k του αρχείου) βρίσκουμε τις 2 θέσεις των ομάδων team1[k-1],team2[k-1] στον πίνακα uteam και ανάλογα με τα score1[k-1],score2[k-1] ενημερώνουμε κατάλληλα τους πίνακες points,goals1,goals2 στις αντίστοιχες 2 θέσεις.Έτσι μετά απο αυτή τη διαδικασία για μια ομάδα uteam[j] οι συνολικοί της βαθμοί είναι points[j] ,τα γκολ που έβαλε είναι goals1[j] και αυτά που δέχτηκε goals2[j].

3. Χρησιμοποιώντας bubble sort ταξινομούμε τους πίνακες uteam,points,goals1,goals2 με βάση τους βαθμούς δηλαδή με βάση τα στοιχεία του πίνακα points. Συγκεκριμένα στο κώδικα που κάνει bubble sort στα στοιχεία του πίνακα points ώστε να είναι με φθίνουσα σειρά αυτό που προσθέσαμε είναι ότι κάθε φορά που γίνεται ανταλλαγή στοιχείων 2 θέσεων ανταλλάσουμε τα στοιχεία στις αντίστοιχες θέσεις στους πίνακες uteam,goals1,goals2 ώστε τα δεδομένα μιας ομαδας να παραμείνουν σε αντίστοιχες θέσεις όλων των πινάκων.
4. ‘Εχοντας κάνει τη παραπάνω ταξινόμηση ,ταξινομούμε τις ομάδες οι οποίες είναι ισόβαθμες αλφαβητικά σαν σε ένα λεξικό. Συγκεκριμένα ο κώδικας μας διατρέχει απο την αρχή τον ταξινομημένο πίνακα ως προς τους βαθμούς και μόλις βρεί μια ισόπαλη ομάδα με μια άλλη αποθηκεύει τη θέση της πρώτης ισόπαλης ομάδας(θ1). Έπειτα επειδή θεωρήσαμε οτι μπορεί να υπάρχει πολλαπλή ισοβαθμία(τριπλή,τετραπλή και τα λοιπά) μόλις δούμε οτι τελειώνει η ισοβαθμία αποθηκεύουμε τη θέση της τελευταίας ισόβαθμης ομάδας(θ2). Οπότε μετά κάνουμε ταξινόμηση αλφαβητικά σαν σε ένα λεξικό στα στοιχεία των πινάκων απο τη θέση θ1 ως τη θέση θ2. Στη συνέχεια αν έχει κι άλλα στοιχεία ο πίνακας συνεχίζουμε να τα διατρέχουμε και αν βρούμε κι άλλες ισόβαθμες ομάδες θα τις ταξινομήσουμε και αυτές με τον ίδιο τρόπο.Στο παρακάτω παράδειγμα εκτέλεσης του προγράμματος μας παρατηρούμε ότι στη δεύτερη περίπτωση το πρόγραμμα ταξινομεί αλφαβητικά σαν σε ένα λεξικό και τις ομάδες Germany ,Spain που είναι ισόπαλες καθώς και τις ομάδες Gra,Greece,Portugal που είναι επίσης ισόπαλες αλλά με διαφορετικούς βαθμούς.

```
manos@manos-VirtualBox:~/Documents$ bash results score1
1.      Greece  4       3-2
2.      Italy   4       3-2
3.      Spain   3       2-2
4.      U K     0       1-3
manos@manos-VirtualBox:~/Documents$ bash results score2
1.      Germany 4       4-1
2.      Spain   4       2-1
3.      Turkey  3       1-4
4.      Gra     2       3-4
5.      Greece  2       2-2
6.      Portugal        2       3-3
manos@manos-VirtualBox:~/Documents$
```

score1:

```
U K-Greece:1-2
Greece-Italy:1-1
Spain-U K:1-0
Spain-Italy:1-2
```

score2:

```
Portugal-Greece:1-1
Greece-Gra:1-1
Gra-Portugal:2-2
Spain-Turkey:1-0
Germany-Turkey:3-0
Spain-Germany:1-1
Turkey-Gra:1-0
```