

ΦΥΣ 140 – Εισαγωγή στην Επιστημονική Χρήση Υπολογιστών

4^η Εργασία

Επιστροφή: Τετάρτη 12/10/2022

Υπενθύμιση: Οι εργασίες πρέπει να επιστρέφονται με e-mail στο fotis@ucy.ac.cy που θα στέλνεται από το πανεπιστημιακό σας λογαριασμό το αργότερο μέχρι την ημερομηνία που αναγράφεται.

Ως subject του e-mail θα πρέπει να αναγράφεται την εργασία (username_phy140_hmX όπου X ο αριθμός της εργασίας)

Κάθε αρχείο που επισυνάπτετε (attach) στο e-mail σας θα πρέπει να έχει το όνομα στη μορφή username_hmX.tgz όπου username είναι το username του e-mail σας και X ο αριθμός της εργασίας. Επίσης σαν πρώτο σχόλιο μέσα σε κάθε file που περιέχει το πρόγραμμά σας θα πρέπει να αναφέρεται το ονοματεπώνυμό σας. Οι εργασίες είναι ατομικές και πανομοιότυπες εργασίες δε θα βαθμολογούνται. Για να κάνετε ένα tgz file (ουσιαστικά tar zipped file) θα πρέπει να δώσετε στο terminal την εντολή `tar -czvf username_hmX.tgz *.py` όπου py είναι όλα τα py files των προγραμμάτων σας.

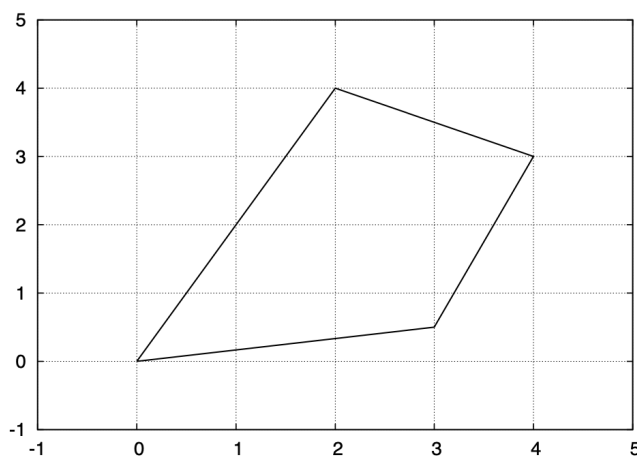
1. Μπορούμε να περιγράψουμε ένα τυχαίο τρίγωνο χρησιμοποιώντας τις συντεταγμένες των τριών κορυφών του: $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Το εμβαδό του τριγώνου δίνεται από την εξίσωση:

$$A = \frac{1}{2} [x_2 y_3 - x_3 y_2 - x_1 y_3 + x_3 y_1 + x_1 y_2 - x_2 y_1]$$

ενώ το μήκος της περιμέτρου του από την εξίσωση:

$$C = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} + \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} + \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2}$$

Γράψτε μία συνάρτηση **triangle(corners)** η οποία να επιστρέφει το εμβαδό A και την περίμετρο, C , ενός τριγώνου οι κορυφές του οποίου ορίζονται από το όρισμα **corners**, που



αποτελεί μια φωλιασμένη λίστα των συντεταγμένων των κορυφών του. Για παράδειγμα, **corners** μπορεί να είναι `[[0,0], [1,0], [0,2]]` αν οι τρεις κορυφές του τριγώνου έχουν συντεταγμένες (0,0), (1,0) και (0,2). Δείξτε πως θα καλέσετε τη συνάρτηση **triangle** για να υπολογίσετε το εμβαδό και περίμετρο του τριγώνου με κορυφές (-3,0), (3,0) και (0,4). Χρησιμοποιείστε την συνάρτησή σας για να υπολογίσετε το εμβαδό του πολυγώνου του παρακάτω σχήματος.

2. Το αποτέλεσμα ενός υπολογισμού είναι ένα σύνολο σημείων που σχηματίζουν μια καμπύλη. Κάθε σημείο αντιπροσωπεύεται από ένα ζεύγος συντεταγμένων (x, y) . Το σύνολο αυτό των σημείων είναι αποθηκευμένο σε ένα αρχείο με την x συντεταγμένη του κάθε σημείου ως την 1^η στήλη και την y συντεταγμένη ως την 2^η στήλη. Συγκεκριμένα, το περιεχόμενο του αρχείου μοιάζει με το ακόλουθο:

```
# File with (x,y) data
#
x = 0.102871   y = 8.12134
x = 0.113526   y = 7.98211
x = 0.132912   y = 2.67152
```

Το αρχείο μπορεί να περιέχει στην αρχή κάποιες γραμμές με σχόλια οι οποίες ξεκινούν με το σύμβολο `#` στην αρχή της γραμμής. Να γράψετε ένα πρόγραμμα το οποίο περιέχει μια συνάρτηση το όνομα της οποίας είναι **readme**, που δέχεται σαν όρισμα το όνομα του αρχείου το οποίο δίνει ο χρήστης στο κύριο πρόγραμμα από το πληκτρολόγιο και επιστρέφει τα δεδομένα για τις συντεταγμένες x και y ως στοιχεία για δύο *lists* x και y . Το κύριο πρόγραμμά σας θα πρέπει να τυπώνει στην οθόνη τις συντεταγμένες που διαβάσατε σε δύο στήλες με την μορφή $(x.xxx, y.yyy)$ όπου οι αριθμοί x και y έχουν 3 και 4 δεκαδικά ψηφία και μήκος 5 και 6 ψηφίων αντίστοιχα. Χρησιμοποιώντας αυτά που μάθατε στις ασκήσεις της 1^{ης} και 2^{ης} κατ' οίκον εργασιών, να κάνετε το γράφημα της καμπύλης με βάση τις συντεταγμένες των σημείων που διαβάσατε.

3. Υποθέστε ότι τα δεδομένα στην προηγούμενη άσκηση αντιπροσωπεύουν κάποια ποσότητα y που ταλαντώνεται συναρτήσει του x . Ενδιαφερόμαστε να προσδιορίσουμε όλα τα τοπικά μέγιστα της καμπύλης και τις αποστάσεις μεταξύ των x -θέσεων των μεγίστων που βρίσκουμε. Οι αποστάσεις αυτές αντιπροσωπεύουν την περίοδο των ταλαντώσεων, ενώ τα μέγιστα αντιστοιχούν στο πλάτος των ταλαντώσεων. Γράψτε ένα πρόγραμμα στην Python το οποίο χρησιμοποιεί μια συνάρτηση **findmax** που δέχεται τις x και y συντεταγμένες ως ορίσματα τύπου `array`, και επιστρέφει τα σημεία που αντιστοιχούν στα μέγιστα καθώς και τις αποστάσεις μεταξύ τους. Σημειώστε ότι ένα τοπικό μέγιστο συμβαίνει σε μια θέση $x[k]$ όταν $y[k-1] < y[k] > y[k+1]$.

Για σκοπούς ελέγχου του προγράμματός σας, θεωρήστε ότι οι συντεταγμένες των σημείων προέρχονται από την καμπύλη της συνάρτησης $f(x) = e^{-0.1x} \cos(2\pi x)$, όπου το x παίρνει τιμές στο διάστημα $[0,10]$ δευτερόλεπτα με βήμα 0.1. Βρείτε όλα τα τοπικά μέγιστα και τις αντίστοιχες αποστάσεις τους. Θα πρέπει να βρείτε επίσης την μεγαλύτερη και μικρότερη τιμή ανάμεσα σε αυτά τα τοπικά μέγιστα και να τις τυπώσετε στην οθόνη με ακρίβεια 4 δεκαδικών ψηφίων. Χρησιμοποιώντας αυτά που χρησιμοποιήσατε στην 1^η άσκηση της 1^{ης} κατ' οίκον εργασίας, να κάνετε το γράφημα της καμπύλης με βάση τις συντεταγμένες των σημείων που έχετε δημιουργήσει καθώς και την καμπύλη των μεγίστων συναρτήσει της θέσης x που βρέθηκε.

4. Πολλές φορές χρειάζεται να ψάξουμε για έναν συγκεκριμένο συνδυασμό αριθμό ή χαρακτήρων σε μια πολύ μακριά γραμματοσειρά (string). Για παράδειγμα μπορεί να έχετε ένα string της μορφής $gen = 'AGTCAATGGAATAGGCCAAGCGAATATTTGGGCTACCA'$.

(α) Γράψτε μια συνάρτηση $freq(letter, text)$ η οποία θα επιστρέφει τη συχνότητα του χαρακτήρα $letter$, στο string $text$, δηλαδή τον αριθμό των εμφανίσεων του χαρακτήρα διαιρεμένο με το μήκος του $string$. Καλέστε την συνάρτηση $freq$ για να βρείτε τη συχνότητα εμφάνισης του χαρακτήρα C και G στο string gen που δόθηκε παραπάνω. [2μ]

(β) Γράψτε μια συνάρτηση $pairs(letter, text)$ η οποία μετρά πόσες φορές ένα ζεύγος χαρακτήρων του χαρακτήρα $letter$ (π.χ. CC , ή GG) εμφανίζεται στο string $text$. Χρησιμοποιήστε την συνάρτηση που γράψατε για να βρείτε πόσες φορές εμφανίζεται το ζεύγος AA στην παραπάνω string gen . [2μ]

(γ) Γράψτε μια συνάρτηση $mystruct(text)$ η οποία μετρά το πλήθος συγκεκριμένων δομών χαρακτήρων μέσα στο string $text$. Η δομή αποτελείται από το χαρακτήρα G ακολουθούμενο από τον χαρακτήρα A ή T έως ότου υπάρξει ένα διπλό GG . Για παράδειγμα μπορείτε να δείτε ως $text = GAATAGG$ ή $text = GAATATTTGG$. [6μ]

5. Υποθέστε ότι έχετε κάποιες μετρήσεις $y_i, i=1,2,3,\dots,5$ ενός μεγέθους y . Οι μετρήσεις λήφθηκαν ανά δευτερόλεπτο. Αυτό που θα χρειαστεί να κάνετε είναι να γράψετε ένα πρόγραμμα το οποίο βρίσκει την καλύτερη ευθεία που περνά από τις μετρήσεις σας. Θα χρειαστείτε αυτά που μάθατε για γραφήματα στις πρώτες 2 κατ'οίκον εργασίες.

(α) Γράψτε μία συνάρτηση η οποία υπολογίζει το σφάλμα μεταξύ της τιμής y_θ που υπολογίζετε από την εξίσωση της ευθείας, $y_\theta^i = ax_i + b$ όπου x_i είναι η χρονική στιγμή i που αντιστοιχεί στην μέτρηση y_i , για όλες τις μετρήσεις:

$$e = \sum_{i=1}^5 (ax_i + b - y_i)^2$$

(β) Γράψτε μια συνάρτηση με ένα *loop* στην οποία εισάγετε τιμές για το a και b και οι αντίστοιχες τιμές του σφάλματος e τυπώνεται στην οθόνη με ακρίβεια 6 δεκαδικών ψηφίων, καθώς και το γράφημα της ευθείας $f(x) = ax + b$ μαζί με τις επιμέρους μετρήσεις. Θα πρέπει να έχετε δύο εντολές *plot* μία για τη συνάρτηση και μια για τις μετρήσεις.

(γ) Θεωρήστε ότι οι μετρήσεις που πήρατε βρίσκονται στο αρχείο *measurements.dat* σε μορφή δύο στηλών όπως στον παρακάτω πίνακα.

Time	y
0	0.5
1	2.0
2	1.0
3	1.5
4	7.5

Χρησιμοποιήστε την συνάρτηση του ερωτήματος (β) για να ψάξετε για τις τιμές του a και b που δίνουν το μικρότερο e (δηλαδή το ελαχιστοποιούν). Τότε θα έχετε βρει και την καλύτερη δυνατή ευθεία που περνά από τις πειραματικές μετρήσεις σας. Θεωρήστε ότι το αρχείο είναι γραμμένο ακριβώς με την μορφή του πίνακα. Υπάρχει δηλαδή μια γραμμή που αναφέρει *Time* και y και ακολουθούν ζεύγη τιμών χωρισμένες με κενό.