MEM-704 - Μηχανική Μάθηση (Εαρινό 2022)
Python: Ασκήσεις Επανάληψης
Λίστες, Λεξικά, Κλάσεις, Numpy

Παρασκευή 25/02/2022

1. Γράψτε ένα πρόγραμμα που να διαβάζει από τον χρήστη μία λίστα με λίστες, της μορφής $[[1,2,3],[4,5,6],[7,8,9]]$. Υποθέστε ότι αυτή η λίστα παριστάνει τον πίνακα
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$
Στην συνέχεια κατασκευάστε μία συνάρτηση με όνομα $sum\_elements$ η οποία με όρισμα μια τέτοια λίστα θα επιστρέφει το άθροισμα των στοιχείων του πίνακα.

2. ($\star$) Σε συνέχεια της προηγούμενης άσκησης κατασκευάστε την συνάρτηση matrix_extend η οποία παίρνει ως όρισμα μία τέτοια λίστα-πίνακα όπως προηγουμένως και παράγει τον $(n+1)\times(n+1)$ πίνακα όπου, τα στοιχείο $A[i, n+1]$ θα είναι το άθροισμα των στοιχείων της i-οστής γραμμής, τα στοιχεία $A[n+1, j]$ θα είναι το άθροισμα της j-οστής στήλης, και τέλος το στοιχείο $A[n+1, n+1]$ θα είναι το άθροισμα των στοιχείων της διαγωνίου.

   Για το παραπάνω παράδειγμα το αποτέλεσμα θα είναι
$$\begin{bmatrix} 1 & 2 & 3 & 6 \\ 4 & 5 & 6 & 15 \\ 7 & 8 & 9 & 24 \\ 12 & 15 & 18 & 15 \end{bmatrix}$$

3. Φτιάξτε μία συνάρτηση η οποία παίρνει ως όρισμα μία λίστα με στοιχεία tuples που περιγράφουν συντεταγμένες στο επίπεδο και επιστρέφει ένα λεξικό με κλειδιά τα σημεία στο x και τιμές τα αντίστοιχα y, πχ αν L=[(1,2), (3,5), (0,1)] τότε d={1:2, 3:5, 0:1}

4. ($\star$) Φτιάξτε ένα πρόγραμμα που θα ζητάει από τον χρήστη δύο αριθμούς x, y οι οποίοι παριστάνουν συντεταγμένες στο επίπεδο και θα τα αποθηκεύει σε ένα λεξικό μέσα σε tuples στην μορφή { 'point 1': (x_1, y_1), 'point 2': (x_2, y_2), …}, όπου x_i, y_i οι αριθμοί που δίνει ο χρήστης. Η είσοδος των σημείων θα σταματάει όταν ο χρήστης δώσει την λέξη 'end'.

   Στη συνέχεια κατασκευάστε την συνάρτηση in_cyrcle η οποία με όρισμα το λεξικό και έναν άριθμό r θα επιστρέφει μία λίστα με όλα τα σημεία τα οποία βρίσκονται μέσα στον κύκλο κέντρου $(0,0)$ και ακτίνας r.

5. Ορίστε την κλάση Quadratic για τη λύση της εξίσωσης $ax^2 + bx + c = 0$, όπου a, b, c πραγματικοί αριθμοί με a διαφορετικό από το μηδέν. Ορίστε τη μέθοδο $discr$ για τον υπολογισμό της διακρίνουσας της δευτεροβάθμιας εξίσωσης. Ορίστε τη μέθοδο solve η οποία να επιστρέφει τις λύσεις της εξίσωσης αν αυτές είναι πραγματικοί αριθμοί ή None, σε αντίθετη περίπτωση. Κατασκευάστε τη δευτεροβάθμια εξίσωση Q όπου ο συντελεστής του δευτροβάθμιου όρου είναι το πρώτο ψηφίο του αριθμού μητρώου σας και οι υπόλοιποι συντελεστές είναι τυχαίοι πραγματικοί αριθμοί στο διάστημα $[-3, 4)$. Λύστε τη δευτεροβάθμια εξίσωση Q και αποθηκεύστε στην πλειάδα r τις ρίζες της.

6. Με χρήση της numpy κατασκευάστε ένα πίνακα $A \in \mathbb{R}^{n,n}$ για n δοθέν με τυχαίους ακέραιους στο διάστημα $[1, 10]$. Στη συνέχεια υπολογίστε τις νόρμες $||A||_1, ||A||_2, ||A||_\infty$ καθώς και τον αντίστροφο του A.

7. Δημιουργείστε μια συνάρτηση με όνομα $func$ η οποία να δέχεται ως όρισμα έναν αριθμό n και θα κατασκευάζει δύο $n \times n$ πίνακες A, B σε μορφή array της numpy με τυχαίους ακέραιους αριθμούς στο διάστημα $[0, 10]$. Στη συνέχεια η συνάρτηση θα πρέπει να επιστρέφει το 2-διάστατο array x (διαστάσεων $n \times 1$), το οποίο αποτελεί λύση του γραμμικού συστήματος $Bx = b$ με b το 2-διάστατο array (n x 1) $b = (1, 1, ..., 1)^T$, π.χ. αν $n = 2$ και A=numpy.array([[1,1],[1,2]]), B=numpy.array([[1,0],[0,2]]) τότε func(n) θα επιστρέψει το numpy.array([[1],[0]]).

8. $(\star)$ Έστω $A \in \mathbb{R}^{M,M}$. Υλοποιήστε τον παρακάτω κώδικα σε python με τη βοήθεια της numpy. Αρχικοποιούμε $z \in \mathbb{R}^M$ τέτοιο ώστε $||z||_\infty$, πχ $z = e_1 = (1, 0, 0, \ldots, 0)$. Επίσης επιλέγουμε $\varepsilon > 0$, πχ $\varepsilon = 10^{-6}$. Θέτουμε $y^{(0)} = z$. Για $n \in \mathbb{N}$ δεδομένο, υπολογίζουμε τα διανύσματα
$$x = Ay^{(n-1)}$$
$$y^{(n)} = \frac{1}{||x||_\infty} x$$
Αν $||y^{(m)} - y^{(m-1)}||_\infty < \varepsilon$ σταματάμε και επιστρέφουμε το $||Ay||_\infty$ και το $\frac{Ay}{||Ay||_2}$.

   Παράδειγμα, για
$$\begin{bmatrix} 1 & -3 \\ -3 & 1 \end{bmatrix} \in \mathbb{R}^{2,2}$$
   το αποτέλεσμα του αλγορίθμου θα πρέπει να είναι $||y||_\infty \approx 4$ και
   $\frac{Ay}{||Ay||_2} \approx [[-0.70710672], [0.70710684]]$

9. Κατασκευάστε με χρήση της numpy, ένα πενταδιαγώνιο πίνακα $\in \mathbb{R}^{n,n}$ για n δοθέν, που θα έχει 6 στην διαγώνιο, -4 στην πρώτη υπερδιαγώνιο και υποδιαγώνιο

και 1 στην δεύτερη υπερδιαγώνιο και υποδιαγώνιο

$$B = \begin{bmatrix} 6 & -4 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ -4 & 6 & -4 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & -4 & 6 & -4 & 1 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & -4 & 6 & -4 & 1 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 1 & -4 & 6 & -4 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 & -4 & 6 \end{bmatrix}$$

10. ($\star$) Σε συνέχεια της προηγούμενης άσκησης, επιπλέον του πίνακα που κατασκευάσατε, κατασκευάστε $b \in \mathbb{R}^{n,1}$ με τυχαίους ακέραιους στο διάστημα [0,100]. Υλοποιήστε τον αλγόριθμο που φαίνεται παρακάτω.
    Διαλέξτε $k_{max}, \varepsilon \in \mathbb{R}$
    Αρχικοποιήστε: $k = 0, x_0 = [0, \ldots, 0]^T \in \mathbb{R}^{n,1}, r = b - Ax_0, \delta = r^T r, \delta_0 = \delta$
    Όσο ($k < k_{max}$ και $\delta > \varepsilon^2 \delta_0$):
    | $q \leftarrow Ar$
    | $a \leftarrow \frac{\delta}{r^T q}$
    | $x \leftarrow x + ar$
    | $r \leftarrow r - aq$
    | $\delta \leftarrow r^T r$
    | $k \leftarrow k + 1$
    επέστρεψε x.

    Δείτε τι αποτέλεσμα δίνει ο αλγόριθμος σας και συγκρίνετε με την λύση του συστήματος $Ax = b$ που δίνει η numpy.