Αριθμητική Ανάλυση
Πρώτη Υποχρεωτική Εργασία

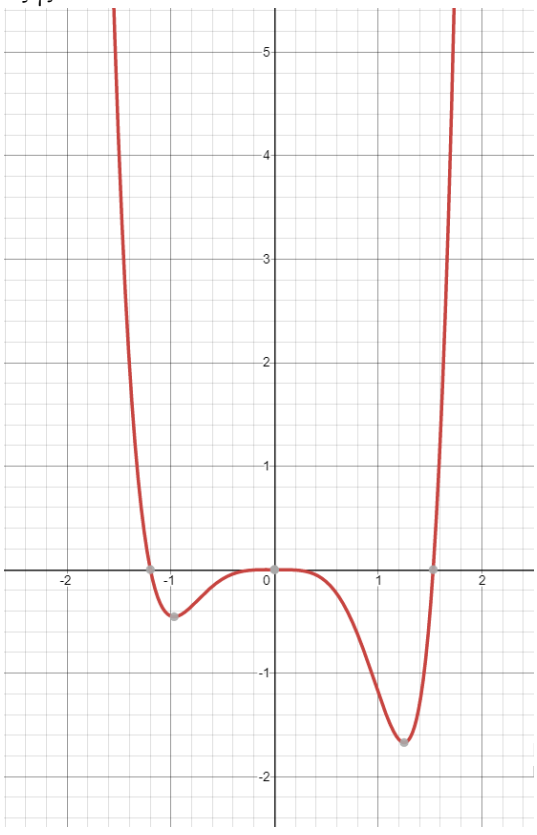
Θοδωρής Δουγαλής, ΑΕΜ: 3544

21 Οκτωβρίου 2024

Ο κώδικας γράφηκε στο *IDLE* της *Python* και δοκιμάστηκε στο *Shell*

1 Πρώτη Άσκηση

Η γραφική παράσταση της $f(x) = e^{sin^3 x} + x^6 - 2x^4 - x^3 - 1, x\epsilon[-2, 2]$ είναι η εξής:



(α) Για τη μέθοδο της διχοτόμησης, έχω φτιάξει τρεις συναρτήσεις:

(1) Την f που δέχεται ένα όρισμα x και επιστρέφει το αποτέλεσμα της

$$f(x) = e^{sin^3 x} + x^6 - 2x^4 - x^3 - 1$$

(2) Την $minRepeatsNeeded$ που δέχεται δύο ορίσματα a, b τα άκρα του διαστήματος και επιστρέφει τον αριθμό από τον οποίο μεγαλύτερο πρέπει να είναι το πλήθος των επαναλήψεων για τη ζητούμενη ακρίβεια.

(3) Την $dixotomisif$ που δέχεται τρία ορίσματα a, b τα άκρα του διαστήματος και r το πλήθος των επαναλήψεων και επιστρέφει είτε την προσέγγιση της ρίζας μετά από αυτές τις επαναλήψεις, είτε ακριβή ρίζα εφόσον αυτή προσεγγιστεί ακριβώς, είτε μήνυμα ότι δεν είναι προσεγγίσιμη κάποια ρίζα.

Από τη γραφική παράσταση φαίνεται ότι υπάρχουν τρεις ρίζες:
$x_a \, \epsilon[-2, -1]$, $x_b \, \epsilon[-1, 1]$, $x_c \, \epsilon[1, 2]$.

- Ρίζα x_a: Η $minRepeatsNeeded$ για το διάστημα $[-2, -1]$ επέστρεψε 17.609640474436812 άρα κάνω 18 επαναλήψεις και η $dixotomisif$ επέστρεψε ως προσέγγιση της ρίζας το $x_a = -1.1976203918457031$.
- Ρίζα x_b: Η $minRepeatsNeeded$ για το διάστημα $[-1, 1]$ επέστρεψε 18.302787654996756 άρα κάνω 19 επαναλήψεις, αλλά αναμενόμενα η $dixotomisif$ επέστρεψε ότι δεν μπορεί να προσεγγιστεί η ρίζα x_b αφού δεν εφαρμόζεται το θεώρημα $Bolzano$ στο συγκεκριμένο διάστημα.
- Ρίζα x_c: Η $minRepeatsNeeded$ για το διάστημα $[1, 2]$ επέστρεψε 17.609640474436812 άρα κάνω 18 επαναλήψεις και η $dixotomisif$ επέστρεψε ως προσέγγιση της ρίζας το $x_c = 1.5301322937011719$.

(β) Για τη μέθοδο $Newton - Raphson$ χρησιμοποίησα την ίδια f και έφτιαξα επιπλέον τρεις συναρτήσεις:

(1) Την df που δέχεται ένα όρισμα x και επιστρέφει το αποτέλεσμα της $f'(x)$.

(2) Την ddf που δέχεται ένα όρισμα x και επιστρέφει το αποτέλεσμα της $f''(x)$.

(3) Την NRf που δέχεται ένα όρισμα x -το αρχικό σημείο x_0 της μεθόδου- και ελέγχει εάν είναι έγκυρο το σημείο ώστε να αρχίσει μία επαναληπτική διαδικασία μέχρι να φτάσει στο ζητούμενο σφάλμα και να επιστρέψει την προσεγγιστική τιμή (με όριο τις εκατό επαναλήψεις) ή επιστρέφει ότι δεν είναι έγκυρο το αρχικό σημείο και θα πρέπει να επιλεγεί άλλο.

Προσεγγίσεις ριζών:

- Ρίζα x_a: Η NRf με αρχικό σημείο το $x_0 = -1.2$ επέστρεψε ως προσέγγιση το $x_a = -1.19762372213357$ μετά από 3 επαναλήψεις.

- Ρίζα x_b: Η NRf με αρχικό σημείο το $x_0 = -0.1$ επέστρεψε ως προσέγγιση το $x_b = -3.875801218753405 \times 10^{-5}$ μετά από 27 επαναλήψεις.
- Ρίζα x_c: Η NRf με αρχικό σημείο το $x_0 = 1.6$ επέστρεψε ως προσέγγιση το $x_c = 1.5301335081674237$ μετά από 4 επαναλήψεις.

(γ) Για τη μέθοδο της τέμνουσας ($Secant$) χρησιμοποίησα την ίδια f και έφτιαξα ακόμη μία συνάρτηση:

- Την $Secantf$ που δέχεται δύο ορίσματα -τα αρχικά σημεία x_0, x_1 της μεθόδου- και ξεκινάει μία επαναληπτική διαδικασία κατά την οποία υπολογίζει τα επόμενα σημεία, μέχρι να βρεθεί και να επιστραφεί προσέγγιση με τη ζητούμενη ακρίβεια (με όριο τις εκατό επαναλήψεις).

Προσεγγίσεις ριζών:

- Ρίζα x_a: Η $Secantf$ με αρχικά σημεία τα $x_0 = -2, x_1 = -1.9$ επέστρεψε ως προσέγγιση το $x_a = -1.1976237229199709$ μετά από 10 επαναλήψεις.
- Ρίζα x_b: Η $Secantf$ με αρχικά σημεία τα $x_0 = -0.2, x_1 = -0.1$ επέστρεψε ως προσέγγιση το $x_b = -5.801062679076745 \times 10^{-5}$ μετά από 38 επαναλήψεις.
- Ρίζα x_c: Η $Secantf$ με αρχικά σημεία τα $x_0 = 1.9, x_1 = 2$ επέστρεψε ως προσέγγιση το $x_c = 1.5301335083948486$ μετά από 8 επαναλήψεις.

Για τις ρίζες x_a και x_c παρατηρώ ότι και οι τρεις μέθοδοι τις προσεγγίζουν με παρόμοιο αριθμό επαναλήψεων μεταξύ τους, διαφορετικό για κάθε μέθοδο, ενώ για την x_b, ότι δεν προσεγγίζεται με τη μέθοδο της διχοτόμησης ενώ προσεγγίζεται έπειτα από μεγάλο αριθμό επαναλήψεων από τις άλλες δύο μεθόδους.

Η x_b είναι η μόνη ρίζα στην οποία δεν συγκλίνει τετραγωνικά η μέθοδος $Newton-Raphson$ και αυτό ωφείλεται στο ότι δεν είναι απλή, αλλά πολλαπλή ρίζα. Έτσι όσο πλησιάζουμε στη ρίζα, η παράγωγος τίνει να μηδενιστεί με αποτέλεσμα η εφαπτομένη στο εκάστοτε σημείο να είναι σχεδόν παράλληλη με τον οριζόντιο άξονα και να έχουμε πιο αργή σύγκλιση.

Για τη δημιουργία της γραφικής παράστασης και τον υπολογισμό των παραγόντων χρησιμοποιήθηκαν $online$ εργαλεία.

2 Δεύτερη Άσκηση

(α) Έφτιαξα την $modNR$ η οποία δέχεται πέντε ορίσματα:

- aF: Κάποια συνάρτηση.
- aDf: Η παράγωγος αυτής.
- $aDdf$: Η δεύτερη παράγωγος αυτής.

- x: Το αρχικό σημείο x_0 της μεθόδου.
- dig: Τα ψηφία ακρίβειας που ζητάω.

Ελέγχει εάν ισχύει η αρχική συνθήκη της $Newton-Raphson$ στο δοσμένο σημείο και όταν ισχύει ξεκινάει την αντίστοιχη επαναληπτική διαδικασία μέχρι να φτάσει σε προσέγγιση με τη ζητούμενη ακρίβεια και να την επιστρέψει, ενώ όταν δεν ισχύει επιστρέφει ότι πρέπει να επιλεγεί άλλο αρχικό σημείο x_0.

(β) Έφτιαξα την $modDix$ η οποία δέχεται τέσσερα ορίσματα:

- aF: Κάποια συνάρτηση.
- a, b: Τα άκρα του αρχικού διαστήματος.
- dig: Τα ψηφία ακρίβειας που ζητάω.

Ελέγχει εάν ισχύει για τα αρχικά σημεία η προϋπόθεση του $Bolzano$ και εφόσον ισχύει, ξεκινάει μία αντίστοιχη με αυτήν της κανονικής διχοτόμησης επαναληπτική διαδικασία επιστρέφοντας την ακριβή ρίζα άμα τη βρει ή προσέγγιση με τη ζητούμενη ακρίβεια. Αν δεν ισχύει η αρχική συνθήκη επιστρέφει ότι δεν μπορεί να προσεγγιστεί ρίζα.

(γ) Έφτιαξα την $modSecant$ η οποία δέχεται πέντε ορίσματα:

- aF: Κάποια συνάρτηση.
- x_0, x_1, x_2: Τα αρχικά σημεία της μεθόδου.
- dig: Τα ψηφία ακρίβειας που ζητάω.

Εφαρμόζει τον ζητούμενο τύπο μέχρι να επιτευχθεί η ζητούμενη ακρίβεια σε μια επαναληπτική διαδικασία αντίστοιχη με της κανονικής μεθόδου της τέμνουσας.

Για τα ερωτήματα που ακολουθούν έφτιαξα τις συναρτήσεις $f2, df2, ddf2$ που δέχονται ως όρισμα μία τιμή και υπολογίζουν τις $f(x), f'(x), f''(x)$ για αυτή την τιμή αντίστοιχα.

(1) **(α)** Με την $modNR$ και εισόδους $(f2, df2, ddf2, x_0, 5)$,

- με $x_0 = -1.5$ επέστρεψε προσέγγιση
  $x_a = -1.3812984820439946$ μετά από 4 επαναλήψεις.
- με $x_0 = -0.7$ επέστρεψε προσέγγιση
  $x_b = -0.6666683930525485$ μετά από 9 επαναλήψεις.
- με $x_0 = 0.2$ επέστρεψε προσέγγιση
  $x_c = 0.20518292468904759$ μετά από 2 επαναλήψεις.
- με $x_0 = 0.6$ επέστρεψε προσέγγιση
  $x_d = 0.5$ μετά από 4 επαναλήψεις.
- με $x_0 = 1.2$ επέστρεψε προσέγγιση
  $x_e = 1.176115557354947$ μετά από 3 επαναλήψεις.

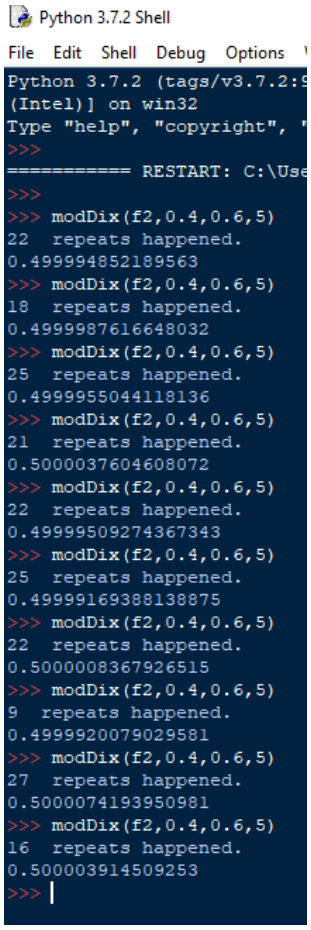
(β) Με την $modDix$ και εισόδους $(f2, a, b, 5)$,

- με $a = -1.4, b = -1.3$ επέστρεψε προσέγγιση $x_a = -1.3812954869691583$ μετά από 17 επαναλήψεις.
- με $a = -0.7, b = -0.6$ επέστρεψε ότι δεν μπορεί να προσεγγιστεί ρίζα.
- με $a = 0.2, b = 0.3$ επέστρεψε προσέγγιση $x_c = 0.20517731861936536$ μετά από 15 επαναλήψεις.
- με $a = 0.4, b = 0.6$ επέστρεψε προσέγγιση $x_d = 0.5000042116626419$ μετά από 21 επαναλήψεις.
- με $a = 1.1, b = 1.2$ επέστρεψε προσέγγιση $x_e = 1.1761152124547603$ μετά από 19 επαναλήψεις.

(γ) Με την $modSecant$ και εισόδους $(f2, x_0, x_1, x_2, 5)$,

- με $x_0 = -1.9, x_1 = -1.8, x_2 = -1.7$, επέστρεψε προσέγγιση $x_a = -1.3812984820499556$ μετά από 7 επαναλήψεις.
- με $x_0 = -1, x_1 = -0.9, x_2 = -0.8$, επέστρεψε προσέγγιση $x_b = -0.6666708261369593$ μετά από 19 επαναλήψεις.
- με $x_0 = 0, x_1 = 0.1, x_2 = 0.2$, επέστρεψε προσέγγιση $x_c = 0.20518292460196147$ μετά από 3 επαναλήψεις.
- με $x_0 = 0.4, x_1 = 0.45, x_2 = 0.55$, επέστρεψε προσέγγιση $x_d = 0.500000000000047$ μετά από 5 επαναλήψεις.
- με $x_0 = 1.7, x_1 = 1.8, x_2 = 1.9$, επέστρεψε προσέγγιση $x_e = 1.1761155573644855$ μετά από 8 επαναλήψεις.

(2) Όπως φαίνεται και στην παρακάτω εικόνα, δεν συγκλίνει κάθε φορά σε ίδιο αριθμό επαναλήψεων.



(3)

- Για τις μεθόδους διχοτόμησης, παρατηρώ πειραματικά ότι λόγω τυχαιότητας, υπήρξαν φορές που η τροποποιημένη σταμάτησε με λιγότερες επαναλήψεις από την κανονική αλλά η μεγάλη πλειοψηφία των δοκιμών, έβγαλε την κλασική μέθοδο διχοτόμησης να συγκλίνει σε λιγότερες επαναλήψεις. Γενικά η διαφορά ωστόσο δεν ήταν σημαντική σε καμία από τις περιπτώσεις. Οι συγκρίσεις έγιναν με τη χρήση της $minRepeatsNeeded$ και της $modDix$ για $dig = 5$ σε διάφορα διαστήματα.
- Για τη μέθοδο $Newton - Raphson$ και την τροποποιημένη έκδοσή της παρατηρώ ότι η τροποποιημένη σε όλες τις περιπτώσεις συγκλίνει γρηγορότερα από την κλασική μέθοδο στις δύο ρίζες στις οποίες συγκλίνει τετραγωνικά η κανονική μέθοδος στην πρώτη άσκηση. Δεν κατάφερε ωστόσο να προσεγγίσει την πολλαπλή ρίζα της συνάρτησης, βγάζοντας ότι δεν μπορεί να διαιρέσει με το μηδέν(ελέγχθηκε με τα απαραίτητα $print$ ότι γίνεται μετά από πλήθος επαναλήψεων, επειδή η παράγωγος τείνει στο μηδέν όσο προχωράνε οι επαναλήψεις). Οι συγκρίσεις έγιναν

με ίδια αρχικά σημεία και την f της πρώτης άσκησης, καθώς εκείνες τις μεθόδους τις είχα φτιάξει για τη συγκεκριμένη συνάρτηση και όχι να δέχονται όποια συνάρτηση θέλει ο χρήστης. Στη $modNR$ φρόντισα να έχω $dig = 5$ καθώς αυτή την ακρίβεια μόνο αναζητούν οι συναρτήσεις της πρώτης άσκησης.

- Παρατηρήθηκε αντίστοιχη συμπεριφορά με τις $NRf - modNR$. Η τροποποιημένη μέθοδος τέμνουσας συγκλίνει σε όλες τις δοκιμές γρηγορότερα από την κλασική για τις δύο απλές ρίζες, ενώ στην πολλαπλή μετά από πλήθος επαναλήψεων, βγάζει ότι δεν μπορεί να διαιρέσει με το μηδέν. Οι συγκρίσεις έγιναν με τα ίδια αρχικά σημεία και το ενδιάμεσό τους για την τροποποιημένη μέθοδο που δέχεται ένα παραπάνω αρχικό σημείο, με χρήση της f από την πρώτη άσκηση και $dig = 5$ στην τροποποιημένη μέθοδο.

Οι συγκρίσεις μπορούσαν να γίνουν μόνο για την

$$f(x) = e^{sin^3 x} + x^6 - 2x^4 - x^3 - 1, x\epsilon[-2, 2]$$

λόγω της υλοποίησης μου για τις συναρτήσεις της πρώτης άσκησης.

3 Τρίτη Άσκηση

(α) Η συνάρτηση axb παίρνει στην είσοδο έναν πίνακα A και ένα διάνυσμα β και επιστρέφει τη λύση χ του συστήματος

$$Ax = b$$

όπως φαίνεται στο παράδειγμα παρακάτω:

```
>>> A=[[2,1,5],[4,4,-4],[1,3,1]]
>>> b=[[3],[4],[5]]
>>> axb(A,b)
[[-0.25], [1.625], [0.375]]
```

Πιο αναλυτικά, έχω φτιάξει τις συναρτήσεις

- $createID$: Δέχεται σαν παράμετρο εισόδου μία διάσταση dim και δημιουργεί έναν μοναδιαίο $dim \times dim$ πίνακα, τον οποίο επιστρέφει.
- $swapRows$: Δέχεται τρία ορίσματα, A ένας πίνακας, $r1, r2$ δύο γραμμές του A και τους αλλάζει θέση.
- $multiply$: Δέχεται δύο ορίσματα A,B δύο πίνακες και εκτελεί πολλαπλασιασμό πινάκων. Επιστρέφει το γινόμενο του B πολλαπλασιασμένου από αριστερά με τον A.
- $createCopy$: Δέχεται ένα όρισμα A έναν τετραγωνικό πίνακα και δημιουργεί ένα αντίγραφο αυτού, το οποίο και επιστρέφει.
- $pivot$: Δέχεται δύο ορίσματα A,P όπου A ο πίνακας συντελεστών του συστήματος και P ένας μοναδιαίος πίνακας ίδιων διαστάσεων με του A. Αυτή η συνάρτηση φέρνει τα μεγαλύτερα κατά απόλυτη τιμή στοιχεία

στη διαγώνιο του πίνακα και ταυτόχρονα κάνει τις ίδιες αντιμεταθέσεις γραμμών στον πίνακα P. Επιστρέφει τον A και διατηρούνται οι αλλαγές στον P αφού έχει αρχικοποιηθεί εκτός της συνάρτησης.

- $getLU$: Δέχεται δύο ορίσματα PA, L ο πίνακας συντελεστών μετά από τις αντιμεταθέσεις και ένας μοναδιαίος πίνακας ίδιων διαστάσεων. Αυτή η συνάρτηση μηδενίζει ένα ένα τα στοιχεία κάτω από τη διαγώνιο του πίνακα, κρατάει τα λ στον πίνακα L και μετατρέπει τον PA στον πίνακα U, τον οποίο και επιστρέφει. Ο L αρχικοποιείται εκτός της συνάρτησης επομένως οι αλλαγές πάνω του διατηρούνται.
- $palu$: Δέχεται ένα όρισμα A τον πίνακα συντελεστών του συστήματος και χρησιμοποιώντας τις προηγούμενες, επιστρέφει τους πίνακες P, L, U χωρίς να πειράξει τον A.
- axb: Δέχεται δύο ορίσματα A,b ο πίνακας συντελεστών και το διάνυσμα με τους σταθερούς παράγοντες του συστήματος, κάνει την LU αποσύνθεση, λύνει αρχικά ένα σύστημα με τον κάτω τριγωνικό πίνακα και με αντικατάσταση λύνει άλλο ένα με τον άνω τριγωνικό και βρίσκει το ζητούμενο διάνυσμα των αγνώστων του συστήματος, το οποίο και επιστρέφει.

Το πρώτο ερώτημα αυτής της άσκησης με έβγαλε εκτός χρονοδιαγράμματος και δεν πρόλαβα να ασχοληθώ εντός προθεσμίας με τα επόμενα δύο.

4 Τέταρτη Άσκηση

Για την τέταρτη άσκηση, έφτιαξα δύο βασικές συναρτήσεις που χρειάζονται σε όλα τα ερωτήματα:

(α) $powerM$: Δέχεται ορίσματα A ο πίνακας γειτνίασης,b το αρχικό διάνυσμα της μεθόδου και dig τα ζητούμενα ψηφία ακρίβειας και υλοποιεί τη μέθοδο των δυνάμεων, επιστρέφοντας την προσέγγιση της ιδιοτιμής και το αντίστοιχο ιδιοδιάνυσμα.

(β) $AtoG$: Δέχεται ορίσματα A τον εκάστοτε πίνακα γειτνίασης και q την πιθανότητα ο χρήστης να μπει σε μία τυχαία σελίδα και δημιουργεί με το δοσμένο τύπο τον πίνακα G, τον οποίο και επιστρέφει.

(1) Η συνάρτηση $task1$ δημιουργεί τον ζητούμενο πίνακα γειτνίασης, από αυτόν δημιουργεί τον πίνακα G και επιστρέφει $True$ μόνο αν το άθροισμα των στοιχείων κάθε στήλης του πίνακα G ισούται με μονάδα.

```
>>> task1()
True
```

(2) Η συνάρτηση $task2$ δημιουργεί τον ζητούμενο πίνακα γειτνίασης, από αυτόν δημιουργεί τον πίνακα G, εφαρμόζει τη μέθοδο των δυνάμεων στον G, κανονικοποιεί το ιδιοδιάνυσμα διαιρώντας κάθε στοιχείο του με το άθροισμα όλων των στοιχείων και επιστρέφει το κανονικοποιημένο διάνυσμα, το οποίο

είναι ίδιο με το ζητούμενο.

```
>>> task2()
[[0.026824566615597813], [0.029861080202273117], [0.02986108020227312], [0.02682
456661559782], [0.03958721556611251], [0.03958721556611251], [0.0395872155661125
06], [0.039587215566112506], [0.07456438650165337], [0.10631995294052222], [0.10
631995294052222], [0.07456438650165337], [0.12509163691770422], [0.1163278913800
4858], [0.12509163691770422]]
```

(3) Στην *task*3 επέλεξα να ανεβάσω το βαθμό σημαντικότητας της πρώτης σελίδας. Έβαλα λοιπόν τις τέσσερις πιο σημαντικές σελίδες προ αλλαγών $(15, 14, 13, 11)$ να έχουν πλέον σύνδεσμο που οδηγεί στην πρώτη ενώ αφαίρεσα έναν από τους συνδέσμους που έδειχναν άλλη σελίδα μέσα στην πρώτη(Τον σύνδεσμο που οδηγούσε στη δεύτερη). Με το νέο πίνακα γειτνίασης λοιπόν, η συνάρτηση δημιουργεί ένα διαφορετικό G, εφαρμόζει σε αυτόν τη μέθοδο των δυνάμεων, κανονικοποιεί το ιδιοδιάνυσμα που επιστρέφει και αν η τάξη της πρώτης σελίδας έχει αυξηθεί, επιστρέφει θετικό μήνυμα.

```
>>> task3()
'Page 1 successfully went viral!'
```

(4)

- Η συνάρτηση *task*4*a* δημιουργεί διαφορετικό πίνακα G για $q = 0.02$ στον οποίο παρατηρείται το εξής φαινόμενο. Οι σελίδες που είχαν χαμηλό βαθμό σημαντικότητας έχουν πλέον ακόμη χαμηλότερο, ενώ αυτές που είχαν υψηλό έχουν αυξημένο, συγκριτικά με τον G για $q = 0.15$, πράγμα λογικό αφού μείωση του q σημαίνει ότι μειώνεται η πιθανότητα να μπει σε κάποια τυχαία σελίδα ο χρήστης και αυξάνεται η πιθανότητα να μπει σε κάποια από τις σελίδες για τις οποίες υπάρχει σύνδεσμος στην τρέχουσα σελίδα.
- Η συνάρτηση *task*4*b* δημιουργεί διαφορετικό πίνακα G για $q = 0.6$ στον οποίο παρατηρείται η αντίθετη επίδραση, πάλι αναμεννόμενα. Με την αύξηση της πιθανότητας ο χρήστης να μπει σε τυχαία σελίδα, μειώνεται η επιρροή των συνδέσμων στο βαθμό σημαντικότητας κάθε σελίδας. Επομένως αυτά που είχαν χαμηλό βαθμό σημαντικότητας έχουν μεγαλύτερο από πριν και όσες είχαν υψυλό έχουν χαμηλότερο.

Όσο υψηλότερη τιμή έχει το q, τόσο μικρότερη σημασία έχουν οι συνδέσεις μεταξύ σελιδών.

```
>>> task4a()
[[0.01710633609180312], [0.01442886834403485], [0.014428868344034852], [0.017106
33609180312], [0.03218980154789739], [0.03218980154789739], [0.03218980154789739
], [0.03218980154789739], [0.080029707822999806], [0.10957603836854915], [0.10957
603836854915], [0.080029707822999808], [0.14349850980547213], [0.14196187294269583], [0.14349850980547216]]
>>> task4b()
[[0.05133212247573803], [0.0579997205713245], [0.05799972057132445], [0.05133212
247573791], [0.05666061237869055], [0.05666061237869053], [0.056660612378690155]
, [0.056660612378690134], [0.0669548722688541], [0.09026137328446875], [0.090261
37328446822], [0.06695487226885379], [0.08344223886852901], [0.07337689554741221
], [0.08344223886852775]]
```

(5) Η συνάρτηση *task*5 δημιουργεί τον ζητούμενο πίνακα γειτνίασης, από αυτόν δημιουργεί τον πίνακα G, εφαρμόζει τη μέθοδο των δυνάμεων στον G, κανονικοποιεί το ιδιοδιάνυσμα διαιρώντας κάθε στοιχείο του με το άθροισμα όλων των στοιχείων και επιστρέφει το κανονικοποιημένο διάνυσμα. Αν ο βαθμός

σημαντικότητας της σελίδας 11 είναι μεγαλύτερος αυτού της σελίδας 10 η συνάρτηση επιστρέφει θετικό μήνυμα, αλλιώς αρνητικό. Στη συγκεκριμένη περίπτωση πέτυχε η στρατηγική.

```
>>> task5()
'It worked'
```

(6) Η συνάρτηση $task6$ δημιουργεί τον πίνακα γειτνίασης δίχως τη δέκατη γραμμή και τη δέκατη στήλη, από αυτόν δημιουργεί τον πίνακα G, εφαρμόζει τη μέθοδο των δυνάμεων στον G, κανονικοποιεί το ιδιοδιάνυσμα διαιρώντας κάθε στοιχείο του με το άθροισμα όλων των στοιχείων και επιστρέφει το κανονικοποιημένο διάνυσμα. Συγκρίνοντας με το ιδιοδιάνυσμα που προκύπτει από τον αρχικό πίνακα, παρατηρώ ότι:

- Αυξήθηκαν για τις σελίδες:1,2,3,4,5,6,7,8,12,15
- Μειώθηκαν για τις σελίδες:9,11,13,14

```
>>> task6()
[[0.047094987462553 08], [0.040911415426832326], [0.03593562367398188], [0.032070
04765982825], [0.042800825586196 92], [0.041391017922889295], [0.0516586592998782
85], [0.050248851636570 65], [0.048223463923863 65], [0.170962695908554 54], [0.103
5981384046709], [0.041161902442336126], [0.107462176687236 75], [0.186480193964607
75]]
```