Διδακτικής της Πληροφορικής

Διδακτικές Προσεγγίσεις σε ζητήματα Αλγοριθμικής & Προγραμματισμού

Αναζήτηση Στοιχείου/Στοιχείων Πίνακα

Σπυρίδων Δουκάκης
sdoukakis@ionio.gr

Αναζήτηση σε Μονοδιάστατο Πίνακα (Ι)

Η λειτουργία της αναζήτησης αποσκοπεί στον εντοπισμό κάποιου στοιχείου σε έναν πίνακα. Η αναζήτηση έχει διάφορες παραλλαγές, που καλύπτουν τις ακόλουθες περιπτώσεις:

- Τα στοιχεία του πίνακα δεν είναι ταξινομημένα και κάθε στοιχείο υπάρχει μόνο μία φορά.
- Κάθε στοιχείο υπάρχει στον πίνακα περισσότερες από μία φορές.
- Ο πίνακας είναι ταξινομημένος είτε σε αύξουσα είτε σε φθίνουσα διάταξη.

Αναζήτηση σε Μονοδιάστατο Πίνακα (ΙΙ)

Να αναπτύξετε αλγόριθμο ο οποίος θα διαβάζει τα 200 αριθμητικά στοιχεία του μονοδιάστατου πίνακα και θα μετράει και θα εκτυπώνει πόσα από αυτά είναι θετικά, πόσα αρνητικά και πόσα μηδέν.

Δίνεται ένας μονοδιάστατος πίνακας Π που περιέχει Ν στοιχεία, τα οποία είναι διαφορετικά μεταξύ τους και δεν είναι ταξινομημένα. Να αναπτύξετε τμήμα αλγορίθμου το οποίο θα αναζητά αν υπάρχει το δεδομένο στοιχείο με όνομα Ζητούμενο στον πίνακα. Ανάλογα με το αποτέλεσμα της αναζήτησης, θα εμφανίζει κατάλληλο μήνυμα.

Αναζήτηση σε Μονοδιάστατο Πίνακα (ΙΙΙ)

```
...
πλ ← 0
Για i από 1 μέχρι N
  Αν Π[i] = Z τότε
    πλ ← πλ + 1
  Τέλος_αν
Τέλος_επανάληψης
Αν πλ ≠ 0 τότε
  Εμφάνισε "Υπάρχει"
αλλιώς
  Εμφάνισε "Δεν υπάρχει"
Τέλος_αν
...
```

```
...
Για i από 1 μέχρι N
  Αν Π[i] = Ζητούμενο τότε
    Εμφάνισε "Υπάρχει"
  αλλιώς
    Εμφάνισε "Δεν υπάρχει"
  Τέλος_αν
Τέλος_επανάληψης
...
```

Λάθος

Αναζήτηση σε Μονοδιάστατο Πίνακα (IV)

```
...
Υπάρχει ← Ψευδής
Για i από 1 μέχρι Ν
  Αν Π[i] = Ζ τότε
    Υπάρχει ← Αληθής
  Τέλος_αν
Τέλος_επανάληψης
Αν Υπάρχει = Αληθής τότε
  Εμφάνισε "Υπάρχει"
αλλιώς
  Εμφάνισε "Δεν υπάρχει"
Τέλος_αν
...
```

```
...
Υπάρχει ← Ψευδής
i ← 1
Όσο i ≤ Ν επανάλαβε
  Αν Π[i] = Ζ τότε
    Υπάρχει ← Αληθής
  Τέλος_αν
  i ← i + 1
Τέλος_επανάληψης
```

Αναζήτηση σε Μονοδιάστατο Πίνακα (V)

```
…
Υπάρχει ← Ψευδής
i ← 1
Όσο i ≤ N επανάλαβε
  Αν Π[i] = Ζ τότε
    Υπάρχει ← Αληθής
  Τέλος_αν
  i ← i + 1
Τέλος_επανάληψης
```

Αναζήτηση σε Μονοδιάστατο Πίνακα (V)

```
…
Υπάρχει ← Ψευδής
i ← 1
Όσο i ≤ N και Υπάρχει = Ψευδής επανάλαβε
  Αν Π[i] = Ζ τότε
    Υπάρχει ← Αληθής
  Τέλος_αν
  i ← i + 1
Τέλος_επανάληψης
Εμφάνισε "Ύπαρξη:", Υπάρχει
…
```

Αναζήτηση σε Μονοδιάστατο Πίνακα (VI)

Να αναπτύξετε τμήμα αλγορίθμου το οποίο θα αναζητά αν υπάρχει δεδομένο στοιχείο στον πίνακα. Ανάλογα με το αποτέλεσμα της αναζήτησης, θα εμφανίζει κατάλληλο μήνυμα.

Στην εκφώνηση δεν επισημαίνεται ότι όλα τα στοιχεία είναι διάφορα μεταξύ τους και άρα ένα ζητούμενο στοιχείο μπορεί να υπάρχει περισσότερες από μία φορές. Παρ’ όλα αυτά η εκφώνηση του παραδείγματος αναζητά αν υπάρχει ένα ζητούμενο στοιχείο και όχι πόσες φορές υπάρχει. Έτσι, ο προηγούμενος αλγόριθμος μπορεί να αξιοποιηθεί και σε αυτή την περίπτωση.

Αναζήτηση σε Μονοδιάστατο Πίνακα (VII)

```
…
Υπάρχει ← Ψευδής
i ← 1
Όσο i ≤ N και Υπάρχει = Ψευδής επανάλαβε
  Αν Π[i] = Ζητούμενο τότε
    Υπάρχει ← Αληθής
  Τέλος_αν
  i ← i + 1
Τέλος_επανάληψης
Αν Υπάρχει = Αληθής τότε
  Εμφάνισε "Υπάρχει"
αλλιώς
  Εμφάνισε "Δεν υπάρχει"
Τέλος_αν
```

Αναζήτηση σε Μονοδιάστατο Πίνακα (VIII)

Να αναπτύξετε τμήμα αλγορίθμου το οποίο θα αναζητά αν υπάρχει δεδομένο στοιχείο στον πίνακα. Ανάλογα με το αποτέλεσμα της αναζήτησης, θα εμφανίζει κατάλληλο μήνυμα και, αν υπάρχει, θα εμφανίζει τη θέση όπου εντοπίστηκε.

Αναζήτηση σε Μονοδιάστατο Πίνακα (IX)

```
...
Υπάρχει ← Ψευδής
i ← 1
Όσο i ≤ N και Υπάρχει = Ψευδής επανάλαβε
  Αν Π[i] = Ζητούμενο τότε
    Υπάρχει ← Αληθής
    θ ← i
  Τέλος_αν
  i ← i + 1
Τέλος_επανάληψης
Αν Υπάρχει = Αληθής τότε
  Εμφάνισε "Υπάρχει", θ
αλλιώς
  Εμφάνισε "Δεν υπάρχει"
Τέλος_αν
```

ή

Αναζήτηση σε Μονοδιάστατο Πίνακα (X)

```
Υπάρχει ← Ψευδής
i ← 1
Όσο i ≤ N και Υπάρχει = Ψευδής επανάλαβε
  Αν Π[i] = Ζητούμενο τότε
    Υπάρχει ← Αληθής
  Τέλος_αν
  i ← i + 1
Τέλος_επανάληψης
Αν Υπάρχει = Αληθής τότε
  Εμφάνισε "Υπάρχει", i - 1
αλλιώς
  Εμφάνισε "Δεν υπάρχει"
Τέλος_αν
```

ή

Αναζήτηση σε Μονοδιάστατο Πίνακα (XI)

```
...
Υπάρχει ← Ψευδής
i ← 1
Όσο i ≤ N και Υπάρχει = Ψευδής επανάλαβε
  Αν Π[i] = Ζητούμενο τότε
    Υπάρχει ← Αληθής
    θ ← i
  αλλιώς
    i ← i + 1
  Τέλος_αν
Τέλος_επανάληψης
...
```

ή

Αναζήτηση σε Μονοδιάστατο Πίνακα (XII)

```
...
Υπάρχει ← Ψευδής
i ← 1
Όσο i ≤ N και Υπάρχει = Ψευδής επανάλαβε
  Αν Π[i] = Ζητούμενο τότε
    Υπάρχει ← Αληθής
  αλλιώς
    i ← i + 1
  Τέλος_αν
Τέλος_επανάληψης
Αν Υπάρχει = Αληθής τότε
  Εμφάνισε "Υπάρχει", i
αλλιώς
  Εμφάνισε "Δεν υπάρχει"
Τέλος_αν
```

Αναζήτηση σε Μονοδιάστατο Πίνακα (XIII)

```
...
Υπάρχει ← Ψευδής
i ← 1
θ ← 0
Όσο i ≤ Ν και Υπάρχει = Ψευδής επανάλαβε
  Αν Π[i] = Ζητούμενο τότε
    Υπάρχει ← Αληθής
    θ ← i
  αλλιώς
    i ← i + 1
  Τέλος_αν
Τέλος_επανάληψης
```

Αναζήτηση σε Μονοδιάστατο Πίνακα (XIV)

Ποια η λειτουργία του αλγορίθμου;

```
θ ← 0
Αρχή_επανάληψης
  θ ← θ + 1
Μέχρις_ότου Π[θ] = Ζ ή θ = Ν
Αν Π[θ] = Ζ τότε
  Εμφάνισε "Υπάρχει", θ
αλλιώς
  Εμφάνισε "Δεν υπάρχει"
Τέλος_αν
```

Αναζήτηση σε Μονοδιάστατο Πίνακα (XV)

Να αναπτύξετε τμήμα αλγορίθμου το οποίο θα αναζητά αν υπάρχει δεδομένο στοιχείο στον πίνακα και θα επιστρέφει το πλήθος των φορών που υπάρχει. Σε περίπτωση που υπάρχει, θα εκτυπώνει τη θέση ή τις θέσεις που υπάρχει το ζητούμενο στοιχείο.

```
πλ ← 0
Για i από 1 μέχρι N
  Αν Π[i] = Ζητούμενο τότε
    πλ ← πλ + 1
    Εκτύπωσε i
  Τέλος_αν
Τέλος_επανάληψης
```

Αναζήτηση σε Μονοδιάστατο Πίνακα (XVI)

Να αναπτύξετε τμήμα αλγορίθμου το οποίο θα αναζητά αν υπάρχει δεδομένο στοιχείο στον πίνακα και θα εμφανίζει το πλήθος των φορών που υπάρχει. Σε περίπτωση που υπάρχει, θα εκχωρεί σε έναν πίνακα τη θέση ή τις θέσεις που υπάρχει το ζητούμενο στοιχείο. Στη συνέχεια θα εμφανίζει τον πίνακα.

Αναζήτηση σε Μονοδιάστατο Πίνακα (XVII)

```
πλ ← 0
Για i από 1 μέχρι Ν
  Αν Π[i] = Ζητούμενο τότε
    πλ ← πλ + 1
    Θ[πλ] ← i
  Τέλος_αν
Τέλος_επανάληψης
Εμφάνισε πλ
Για i από 1 μέχρι πλ
  Εμφάνισε Θ[i]
Τέλος_επανάληψης
```

Αναζήτηση σε Μονοδιάστατο Πίνακα (XVIII)

Δίνεται ο μονοδιάστατος πίνακας Π με Ν ταξινομημένα κατά αύξουσα διάταξη στοιχεία. Να αναπτύξετε τμήμα αλγορίθμου το οποίο θα αναζητά αν υπάρχει δεδομένο στοιχείο στον πίνακα. Ο αλγόριθμος θα επιστρέφει την ύπαρξη ή μη του ζητούμενου στοιχείου και τη θέση του στοιχείου, αν υπάρχει.

Αναζήτηση σε Μονοδιάστατο Πίνακα (XIX)

```
ΥΠ ← Ψευδής
ΞΠ ← Ψευδής
i ← 1
θ ← 0
Όσο i ≤ N και ΥΠ = Ψευδής και ΞΠ = Ψευδής επανάλαβε
  Αν Π[i] = Ζ τότε
    ΥΠ ← Αληθής
    θ ← i
  αλλιώς_αν Π[i] > Ζ τότε
    ΞΠ ← Αληθής
  αλλιώς
    i ← i + 1
  Τέλος_αν
Τέλος_επανάληψης
```

Μέθοδος Διαίρει και Βασίλευε (Ι)

Η «Διαίρει και Βασίλευε» (Divide and Conquer) αποτελεί μια μέθοδο σχεδίασης αλγορίθμων στην οποία εντάσσονται οι τεχνικές που υποδιαιρούν ένα πρόβλημα σε μικρότερα υποπροβλήματα, που έχουν την ίδια τυποποίηση με το αρχικό πρόβλημα, αλλά είναι μικρότερα σε μέγεθος.

Με όμοιο τρόπο, τα υποπροβλήματα αυτά μπορούν να διαιρεθούν σε ακόμη μικρότερα υποπροβλήματα κ.ο.κ. Έτσι η επίλυση ενός προβλήματος έγκειται στη σταδιακή επίλυση των όσο το δυνατόν μικρότερων υποπροβλημάτων, ώστε τελικά να προκύψει η συνολική λύση του αρχικού ευρύτερου προβλήματος.

Η προσέγγιση αυτή ονομάζεται «**από πάνω προς τα κάτω**» (top-down).

Μέθοδος Διαίρει και Βασίλευε (ΙΙ)

Χρησιμοποιώντας τη μέθοδο «Διαίρει και Βασίλευε» δοκιμάστε να παίξετε το παιχνίδι «Μάντεψε τον αριθμό», ως εξής: Σκεφθείτε έναν αριθμό από το 1 έως το 100 και ζητήστε από κάποιον/α άλλο/η (Π1) να μαντέψει τον αριθμό αυτόν με όσο το δυνατόν λιγότερες προσπάθειες.

Μέθοδος Διαίρει και Βασίλευε (ΙΙΙ)

Ποια διαδικασία πρέπει να ακολουθήσει ο Π1 για να βρει τον αριθμό; Τι είδους ερωτήσεις πρέπει να κάνει, ώστε να ελαχιστοποιήσει τις προσπάθειες που θα χρειαστούν;

Ο Π1 πρέπει να ακολουθήσει τη μέθοδο Διαίρει και Βασίλευε.

Κάθε φορά βρίσκει τον μέσο όρο (ακέραιο μέρος) των άκρων του διαστήματος που εξετάζει και ρωτάει αν βρήκε τον αριθμό.

Αν δεν τον βρήκε ρωτάει αν ο αριθμός που ψάχνει είναι μεγαλύτερος ή μικρότερος από τον μέσο όρο και επιλέγει το κατάλληλο διάστημα που βρίσκεται ο αριθμός και επαναλαμβάνει τη διαδικασία.

Για παράδειγμα αν ο αριθμός που αναζητείται είναι ο 67 τότε οι μέσοι όροι και τα διαστήματα που προκύπτουν είναι:

Διάστημα	**Μέσος Όρος**
[1, 100]	50
[51, 100]	75
[51, 74]	62
[63, 74]	68
[63, 67]	65
[66,67]	66
[67,67]	67

Μέθοδος Διαίρει και Βασίλευε (IV)

Ποιος είναι ο μέγιστος αριθμός προσπαθειών που θα χρειαστεί ο Π1 για να μαντέψει τον αριθμό;

$$[\log_2(100) + 1] = [6{,}643856 + 1] = [7{,}643856] = 7$$

Μέθοδος Διαίρει και Βασίλευε (V)

Επαναλάβετε το ίδιο παιχνίδι με έναν αριθμό από το 1 έως το 1000.

Ποιος είναι τώρα ο μέγιστος αριθμός προσπαθειών που θα χρειαστεί ο παίκτης-1 για να μαντέψει τον αριθμό;

$$[\log_2(1000) + 1] = [9{,}965784 + 1] = [10{,}965784] = 10$$

Μέθοδος Διαίρει και Βασίλευε (VI)

Τι παρατηρείτε σχετικά με την αύξηση του αριθμού των προσπαθειών σε σχέση με την αύξηση του πλήθους των δεδομένων;

Ενδεικτικά παρατίθεται ο παρακάτω πίνακας που δείχνει τη σχέση του αριθμού των προσπαθειών και του αριθμού των στοιχείων. Οι αριθμοί των προσπαθειών έχουν υπολογιστεί με βάση τον τύπο.

Στοιχεία N	Αριθμός Προσπαθειών
10	4
100	7
1.000	10
10.000	14
100.000	17
1.000.000	20
10.000.000	24
100.000.000	27
1.000.000.000	30

Μέθοδος Διαίρει και Βασίλευε (VII)

Με βάση τις παρατηρήσεις σας, θεωρείτε ότι η τεχνική «Διαίρει και Βασίλευε» είναι αποτελεσματική για μεγάλο όγκο δεδομένων;

Η μέθοδος Διαίρει και Βασίλευε είναι αποτελεσματική για μεγάλο όγκο δεδομένων όπως φαίνεται και από τον πίνακα.

Στοιχεία N	Αριθμός Προσπαθειών
10	4
100	7
1.000	10
10.000	14
100.000	17
1.000.000	20
10.000.000	24
100.000.000	27
1.000.000.000	30

Δυαδική Αναζήτηση

Αριστερά ← 1
Δεξιά ← Ν
Υπάρχει ← **Ψευδής**
θ ← 0
Όσο Αριστερά ≤ Δεξιά **και** Υπάρχει = **Ψευδής επανάλαβε**
  Μέσο ← (Αριστερά + Δεξιά) **div** 2
  Αν Π[Μέσο] = Ζητούμενο **τότε**
    θ ← Μέσο
    Υπάρχει ← **Αληθής**
  αλλιώς_αν Π[Μέσο] < Ζητούμενο **τότε**
    Αριστερά ← Μέσο + 1
  αλλιώς
    Δεξιά ← Μέσο - 1
  Τέλος_αν
Τέλος_επανάληψης

Μέθοδος διχοτόμησης (Ι)

Σύμφωνα με το θεώρημα Bolzano αν μια συνάρτηση f ορισμένη στο διάστημα $[\alpha, \beta]$ είναι συνεχής στο $[\alpha, \beta]$ και ισχύει $f(\alpha) \cdot f(\beta) < 0$ τότε υπάρχει τουλάχιστον ένα $x0 \in (\alpha, \beta) : f(x0) = 0$. Δηλαδή η εξίσωση $f(x) = 0$ έχει στο (α, β) μια τουλάχιστον λύση.

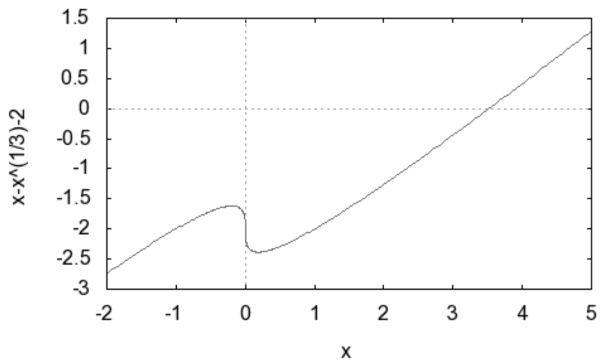
Για να βρεθεί προσεγγιστικά μία ρίζα μιας εξίσωσης $f(x) = 0$ με βάση το θεώρημα Bolzano ή τη μέθοδο της διχοτόμησης όπως αλλιώς ονομάζεται βρίσκεται ένα σημείο $x1 = (\alpha + \beta)/2$ στο μέσο του διαστήματος $[\alpha, \beta]$ και εξετάζεται, αν $f(\alpha) \cdot f(x1) < 0$. Αν ναι, τότε η ρίζα υπάρχει στο διάστημα $[\alpha, x1]$, αλλιώς στο $[x1, \beta]$.

Στη συνέχεια βρίσκεται το μέσο $x2$ του υποδιαστήματος που υπάρχει η ρίζα και επαναλαμβάνονται τα ίδια. Έτσι σε κάθε επανάληψη εξαιρείται από την έρευνα το μισό διάστημα και προφανώς η ακολουθία τιμών $x1, x2, \ldots$ συγκλίνει προς τη ρίζα της εξίσωσης. Η διαδικασία τερματίζεται, όταν η προσέγγιση της ρίζας θεωρείται ικανοποιητική.

Μέθοδος διχοτόμησης (II)

Δίνεται η συνάρτηση $f(x) = x - x^{1/3} - 2$

Όπως φαίνεται από τη γραφική παράσταση παρακάτω υπάρχει μία ρίζα μεταξύ του 3 και του 4 διάστημα στο οποίο η συνάρτηση είναι συνεχής.



Να γράψετε πρόγραμμα το οποίο

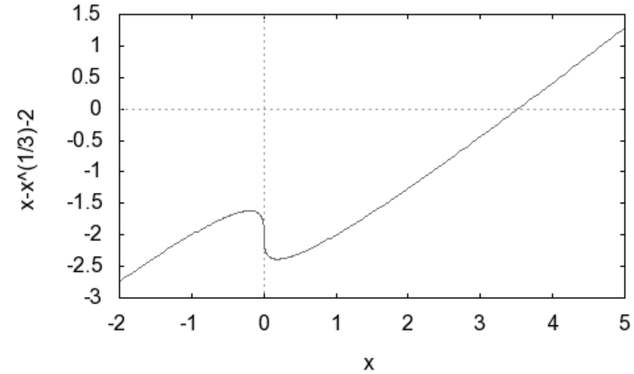
A. εφαρμόζοντας τη μέθοδο της διχοτόμησης (Διαίρει και Βασίλευε) που περιγράφηκε παραπάνω να βρίσκει μία προσεγγιστική λύση η οποία να προσδιορίζει σωστά τα πρώτα 10 δεκαδικά ψηφία της ρίζας. Για να βρεθούν σωστά τα πρώτα 10 δεκαδικά ψηφία πρέπει στο υποδιάστημα που εξετάζεται να ισχύει $|\beta - \alpha| < 10^{-10}$.

B. να εμφανίζει επίσης κάθε φορά και το υποδιάστημα που εξετάζεται.

Μέθοδος διχοτόμησης (III)

Δίνεται η συνάρτηση $f(x) = x - x^{1/3} - 2$

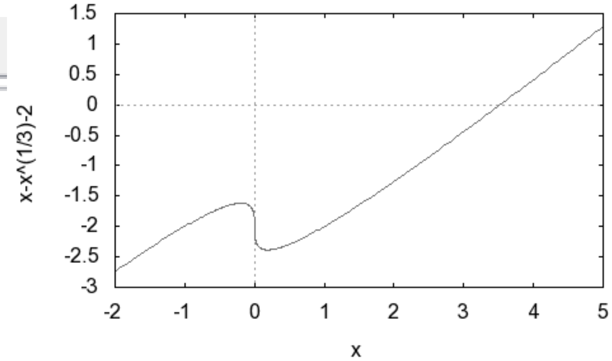
Υπολογίζεται προσεγγιστική λύση στο διάστημα [3,4]




```
ΠΡΟΓΡΑΜΜΑ Bolzano
ΜΕΤΑΒΛΗΤΕΣ
  ΠΡΑΓΜΑΤΙΚΕΣ: L, R, M
ΑΡΧΗ
L <- 3
R <- 4
ΑΝ (L - L^(1/3) - 2)*(R - R^(1/3) - 2) < 0 ΤΟΤΕ
  ΟΣΟ Α_Τ(R - L) > 10^(-10) ΕΠΑΝΑΛΑΒΕ
    ΓΡΑΨΕ "Εξετάζεται το διάστημα [", L, ",", R, "]"
    M <- (L + R)/ 2
    ΑΝ (L - L^(1/3) - 2)*(M - M^(1/3) - 2) < 0 ΤΟΤΕ
        R <- M
    ΑΛΛΙΩΣ
        L <- M
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ  "Η λύση βρίσκεται στο διάστημα [", L, ",", R, "]"
ΑΛΛΙΩΣ
  ΓΡΑΨΕ "Η λύση δεν βρίσκεται στο διάστημα [", L, ",", R, "]"
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Μέθοδος διχοτόμησης (IV)

Οθόνη εκτέλεσης




```
Εξετάζεται το διάστημα [3.0000000000,4.0000000000]
Εξετάζεται το διάστημα [3.5000000000,4.0000000000]
Εξετάζεται το διάστημα [3.5000000000,3.7500000000]
Εξετάζεται το διάστημα [3.5000000000,3.6250000000]
Εξετάζεται το διάστημα [3.5000000000,3.5625000000]
Εξετάζεται το διάστημα [3.5000000000,3.5312500000]
Εξετάζεται το διάστημα [3.5156250000,3.5312500000]
Εξετάζεται το διάστημα [3.5156250000,3.5234375000]
Εξετάζεται το διάστημα [3.5195312500,3.5234375000]
Εξετάζεται το διάστημα [3.5195312500,3.5214843750]
```

…

…

```
Εξετάζεται το διάστημα [3.5213928148,3.5213928223]
Εξετάζεται το διάστημα [3.5213928185,3.5213928223]
Εξετάζεται το διάστημα [3.5213928204,3.5213928223]
Εξετάζεται το διάστημα [3.5213928213,3.5213928223]
Εξετάζεται το διάστημα [3.5213928218,3.5213928223]
Εξετάζεται το διάστημα [3.5213928220,3.5213928223]
Η λύση βρίσκεται στο διάστημα [3.5213928221,3.5213928223]
```