Νίκος Μ. Χατζηγιαννάκης

Η γλώσσα Python σε βάθος

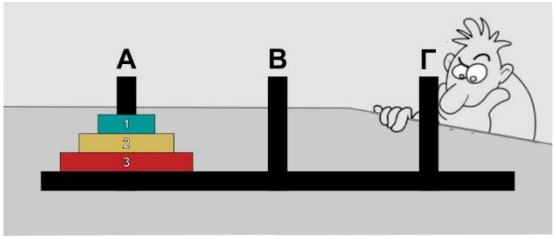
Περιλαμβάνει εισαγωγή
στην επιστήμη των υπολογιστών
και τον προγραμματισμό

3η εργασία – Πύργοι του Ανόϊ

ΚΛΕΙΔΑΡΙΘΜΟΣ

3η Εργασία - Πύργοι του Ανόϊ

Σκεπτικό



Θα προσπαθήσουμε να αναλύουμε τη διαδικασία θεωρώντας ότι έχουμε τρεις δίσκους στον στύλο Α.

Ας υποθέσουμε τώρα ότι θέλουμε να μετακινήσουμε μόνο έναν δίσκο, τον δίσκο 1, από τον στύλο Α στον στύλο Γ. Η λύση είναι απλή και περιλαμβάνει μόνο μια κίνηση χωρίς να χρησιμοποιεί τον βοηθητικό στύλο Β.

Αν τώρα θέλουμε να μετακινήσουμε τους δύο δίσκους 1 και 2, από τον στύλο Α στον στύλο Γ. Τώρα θα χρειαστούμε τρεις κινήσεις. Θα μετακινήσουμε τον δίσκο 1 στο στύλο Β, μετά τον δίσκο 2 στον στύλο Γ και τέλος τον δίσκο 2, από τον στύλο Β στον Γ.

Ας δούμε τώρα πως θα μπορέσουμε να μετακινήσουμε και τους τρεις δίσκους από τον στύλο Α στον στύλο Γ. Αν μετακινήσουμε τους δύο δίσκους 1 και 2 από τον στύλο Α στον Β, θα μείνει μόνο ο δίσκος 3 στον στύλο Α. Μετακινούμε τώρα τον δίσκο 3 από τον στύλο Α στον Γ και μετά τους δύο δίσκους 1 και 2 από τον στύλο Β στον Γ. Παρατηρούμε επομένως ότι το πρόβλημα της μετακίνησης των δύο δίσκων ανάγεται σε πρόβλημα μετακίνησης δύο δίσκων, το οποίο με τη σειρά του σε μετακίνηση μόνο ενός δίσκου. Αυτή είναι μια καθαρά αναδρομική διαδικασία η οποία συνοψίζεται στις παρακάτω προτάσεις:

Αν θέλεις να μετακινήσεις Ν δίσκους από τον στύλο Α στον Γ, με βοηθητικό τον στύλο Β, ακολούθησε τα παρακάτω βήματα:

1. *Μετακίνησε Ν-1 δίσκους από τον Α στον Β με βοηθητικό τον στύλο Γ*
2. *Μετακίνησε τον Ν δίσκο από τον στύλο Α στον στύλο Γ*
3. *Μετακίνησε Ν-1 δίσκους από τον Β στον Γ με βοηθητικό τον στύλο Α*

Βήματα

❶ Το κυριότερο και ίσως το μοναδικό ουσιαστικό βήμα είναι η δημιουργία της αναδρομικής συνάρτησης towers() η οποία θα δέχεται ως ορίσματα το πλήθος των δίσκων που θα μετακινήσουμε, τον στύλο από τον οποίο θα μετακινηθούν, τον στύλο στον οποίο θα καταλήξουν και τον βοηθητικό στύλο που μπορεί να χρησιμοποιηθεί.

❷ Η συνάρτηση towers() αρχικά καλεί αναδρομικά τον εαυτό της για να μετακινήσει num-1 δίσκους από τον αρχικό στύλο στον βοηθητικό.

❸ Αμέσως μετά γίνεται η μετακίνηση του δίσκου num στον προορισμό του.

❹ Στη συνέχεια η συνάρτηση towers() καλεί πάλι αναδρομικά τον εαυτό της για να μετακινήσει τους num-1 δίσκους από τον βοηθητικό στύλο στον τελικό προορισμό τους.

❺ Στην περίπτωση που η συνάρτηση towers() κληθεί με πλήθος 1, προς μετακίνηση δίσκων, τότε απλά γίνεται η μετακίνηση του δίσκου χωρίς αναδρομικές κλήσεις της συνάρτησης.

❻ Η κλήση της συνάρτησης towers() γίνεται από το κυρίως πρόγραμμα ώστε να μετακινήσει n πλήθος δίσκων από τον στύλο A στον Γ με βοηθητικό στύλο τον B. Το πλήθος n των δίσκων δίδεται από τον χρήστη.

Κώδικας

E3.py

```
def towers(num,stylos_apo,stylos_se,stylos_temp):
    if num==1:
        print('Μετακίνησε τον δίσκο 1 από τον στύλο ',stylos_apo,' στον στύλο ', stylos_se)
        return
    # Αναδρομική κλήση της συνάρτησης
    towers(num-1, stylos_apo, stylos_temp, stylos_se)
    print('Μετακίνησε τον δίσκο',num,'από τον στύλο ',stylos_apo, ' στον στύλο ',stylos_se)
    # Αναδρομική κλήση της συνάρτησης
    towers(num-1, stylos_temp, stylos_se, stylos_apo)

# Κυρίως πρόγραμμα
n=int(input('Δώσε αριθμό δίσκων :'))
print('Η σειρά των απαιτούμενων κινήσεων είναι:')
towers(n, 'Α', 'Γ', 'Β')
```

❺ Μη αναδρομική περίπτωση.

❷ Καλείται αναδρομικά με πλήθος δίσκων (num-1). Μετακινεί **num-1** δίσκους από τον αρχικό στύλο στον βοηθητικό.

❸ Γίνεται η μετακίνηση του δίσκου **num**.

❹ Καλείται αναδρομικά για μα μετακινήσει τους **num-1** δίσκους από τον βοηθητικό στον τελικό προορισμό τους.

❻ Κλήση της συνάρτησης **towers()** για τη μετακίνηση **n** δίσκων από τον στύλο **A** στον **G** με βοηθητικό τον **B**.

Προτάσεις

Μια παραλλαγή του κλασικού προβλήματος είναι οι κυκλικοί πύργοι του Ανόϊ (Cyclic Hanoi), όπου φανταστείτε τους στύλους Α, Β και Γ να βρίσκονται κυκλικά τοποθετημένοι, σύμφωνα με τη σειρά των δεικτών του ωρολογίου. Ο έξτρα περιορισμός που τίθεται είναι ότι μετακινήσεις δίσκων μπορούν να γίνουν μόνο κατά την φορά των δεικτών του ωρολογίου. Δηλαδή από τον Α στον Β, ή από τον Β στον Γ, ή από τον Γ στον Α.

Καλό κουράγιο!