

Ενότητα 1: Εισαγωγή Ασκήσεις και Λύσεις

ΗΥ240: Δομές Δεδομένων

Παναγιώτα Φατούρου

Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2024-25

Άσκηση 1

Θεωρήστε την ακολουθία αριθμών Fibonacci πρώτης τάξης που ορίζεται ως εξής:

$$F_i = \begin{cases} 0 & \text{αν } i = 0 \\ 1 & \text{αν } i = 1 \\ F_{i-1} + F_{i-2} & \text{αν } i > 1 \end{cases} \quad (1)$$

Η αναδρομική υλοποίηση της συνάρτησης αυτής παρουσιάζεται στη συνέχεια:

```
function Fib(integer n): integer
int n1, n2;
if (n <= 1) then return n;
else {
n1 = Fib(n-1);
n2 = Fib(n-2);
return (n1+n2);
}
```

Σας ζητείται να τρέξετε στο χαρτί (δηλαδή να κάνετε trace) την Fib(n) για την περίπτωση που $n = 5$. Πρέπει να παρουσιάσετε όλες τις αναδρομικές κλήσεις της Fib() καθώς και τις τιμές των μεταβλητών n, n1 και n2 σε κάθε εκτέλεσή της Fib().

Λύση

Ιχνηλάτηση για $n = 5$

Η συνάρτηση Fibonacci (Fib) υπολογίζει αναδρομικά τον n -οστό αριθμό Fibonacci. Ακολουθεί η ιχνηλάτηση της συνάρτησης για $n = 5$.

Αναδρομικές κλήσεις

Fib(5):

```
n1 = Fib(4)
  n1 = Fib(3)
    n1 = Fib(2)
      n1 = Fib(1) --> return 1
      n2 = Fib(0) --> return 0
      n1+n2 --> return 1 (1 + 0)
    n2 = Fib(1) --> return 1
    n1+n2 --> return 2 (1 + 1)
  n2 = Fib(2)
    n1 = Fib(1) --> return 1
    n2 = Fib(0) --> return 0
    n1+n2 --> return 1 (1 + 0)
  n1+n2 --> return 3 (2 + 1)
n2 = Fib(3)
  n1 = Fib(2)
    n1 = Fib(1) --> return 1
    n2 = Fib(0) --> return 0
    n1+n2 --> return 1 (1 + 0)
  n2 = Fib(1) --> return 1
  n1+n2 --> return 2 (1 + 1)
n1+n2 --> return 5 (3 + 2)
```

Σύνολο αναδρομικών κλήσεων:

Για $n = 5$, η συνάρτηση Fib καλεί τον εαυτό της συνολικά 15 φορές. Πολλές από τις κλήσεις είναι επαναλαμβανόμενες.

Ιχνηλάτηση και χρήση μνήμης για $Fib(5)$

Παρακάτω παρουσιάζεται η ιχνηλάτηση της συνάρτησης Fibonacci για $n = 5$ με την τιμή των μεταβλητών n , $n1$ και $n2$ σε κάθε κλήση:

Άσκηση 2

- a) Αποδείξτε τη μεταβατική και τη συμμετρική ιδιότητα του Θ .
- b) Αποδείξτε ότι $\lg(n!) = O(n \lg n)$.
- c) Ποια από τα ακόλουθα είναι αληθή και γιατί;

(i) $n^3 + 4n^2 + 5n + 10 = \Theta(n^3)$

(ii) $\lg(n^3) = \Theta(n \lg n)$

	κλήση 1	κλήση 2	κλήση 3	κλήση 4	κλήση 5	κλήση 6	κλήση 7	κλήση 8	κλήση 9	κλήση 10	κλήση 11	κλήση 12	κλήση 13	κλήση 14	κλήση 15
	n=5	n=5	n=5	n=5	n=5	n=5	n=5	n=5	n=5	n=5	n=5	n=5	n=5	n=5	n=5
Fib(5)	n1=?	n1=?	n1=?	n1=?	n1=?	n1=?	n1=?	n1=?	n1=?	n1=3	n1=3	n1=3	n1=3	n1=3	n1=3
	n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=2
		n=4	n=4	n=4	n=4	n=4	n=4	n=4	n=4	n=4					
Fib(4)		n1=?	n1=?	n1=?	n1=?	n1=?	n1=2	n1=2	n1=2	n1=2					
		n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=?	n2=1					
			n=3	n=3	n=3	n=3	n=3								
Fib(3)			n1=?	n1=?	n1=?	n1=1	n1=1								
			n2=?	n2=?	n2=?	n2=?	n2=1								
				n=2	n=2	n=2	n=2								
Fib(2)				n1=?	n1=1	n1=1									
				n2=?	n2=?	n2=0									
					n=1										
Fib(1)					n1=?										
					n2=?										
						n=0									
Fib(0)						n1=?									
						n2=?									
							n=1								
Fib(1)							n1=?								
							n2=?								
								n=2	n=2	n=2					
Fib(2)								n1=?	n1=1	n1=1					
								n2=?	n2=?	n2=0					
									n=1						
Fib(1)									n1=?						
									n2=?						
										n=0					
Fib(0)										n1=?					
										n2=?					
											n=3	n=3	n=3	n=3	n=3
Fib(3)											n1=?	n1=?	n1=?	n1=1	n1=1
											n2=?	n2=?	n2=?	n2=?	n2=1
												n=2	n=2	n=2	
Fib(2)												n1=?	n1=1	n1=1	
												n2=?	n2=?	n2=0	
													n=1		
Fib(1)													n1=?		
													n2=?		
														n=0	
Fib(0)														n1=?	
														n2=?	
															n=1
Fib(1)															n1=?
															n2=?

Figure 1: Ιχνηλάτηση της συνάρτησης Fibonacci για $n = 5$ με την τιμή των μεταβλητών $n, n1$ και $n2$ σε κάθε κλήση:

- (iii) $\lg(\sqrt{n}) = \Theta(\lg(n))$
- (iv) $\sum_{i=1}^n (2^*(i+1)) = O(n^3)$
- (v) $\min(700, n^2) = \Theta(1)$

Λύση

a) Αποδείξτε τη μεταβατική και τη συμμετρική ιδιότητα του Θ .

Μεταβατική Ιδιότητα: Αν $f(n) = \Theta(g(n))$ και $g(n) = \Theta(h(n))$, τότε $f(n) = \Theta(h(n))$.

Για να ισχύει $f(n) = \Theta(h(n))$, πρέπει να δείξουμε ότι $f(n) = O(h(n))$ και $f(n) = \Omega(h(n))$.

- **Απόδειξη ότι $f(n) = O(h(n))$:**

Αφού $f(n) = \Theta(g(n)) \Rightarrow f(n) = O(g(n))$, άρα:

$$\exists c_1 \in \mathbb{R}^+, n_1 \in \mathbb{N} \text{ τέτοιο ώστε } f(n) \leq c_1 g(n), \forall n \geq n_1$$

όπου $\mathbb{N}$ το σύνολο των φυσικών αριθμών.

Αφού $g(n) = \Theta(h(n)) \Rightarrow g(n) = O(h(n))$, άρα:

$$\exists c_2 \in \mathbb{R}^+, n_2 \in \mathbb{N} \text{ τέτοιο ώστε } g(n) \leq c_2 h(n), \forall n \geq n_2$$

Επιλέγουμε $n_0 = \max\{n_1, n_2\}$, και επομένως:

$$f(n) \leq c_1 g(n) \leq c_1 c_2 h(n), \forall n \geq n_0$$

Άρα αν επιλέξουμε $c = c_1 c_2$, τότε ισχύει $f(n) = O(h(n))$.

- **Απόδειξη ότι $f(n) = \Omega(h(n))$:**

Αφού $f(n) = \Theta(g(n)) \Rightarrow f(n) = \Omega(g(n))$, άρα:

$$\exists c_1 \in \mathbb{R}^+, n_1 \geq 0 \text{ τέτοιο ώστε } f(n) \geq c_1 g(n), \forall n \geq n_1$$

Αφού $g(n) = \Theta(h(n)) \Rightarrow g(n) = \Omega(h(n))$, άρα:

$$\exists c_2 \in \mathbb{R}^+, n_2 \geq 0 \text{ τέτοιο ώστε } g(n) \geq c_2 h(n), \forall n \geq n_2$$

Επιλέγουμε $n_0 = \max\{n_1, n_2\}$, και επομένως:

$$f(n) \geq c_1 g(n) \geq c_1 c_2 h(n), \forall n \geq n_0$$

Άρα αν επιλέξουμε $c = c_1 c_2$, τότε ισχύει $f(n) = \Omega(h(n))$.

Οπότε, εφόσον $f(n) \in O(h(n))$ και $f(n) \in \Omega(h(n))$ ισχύει $f(n) \in \Theta(h(n))$.

Συμμετρική Ιδιότητα: $f(n) = \Theta(g(n))$, αν και μόνον αν $g(n) = \Theta(f(n))$.

Ξεκινώντας από το $f(n) = \Theta(g(n))$, θα προσπαθήσουμε να δείξουμε ότι $g(n) = \Theta(f(n))$. Συμμετρικά, θα αποδεικνύεται και το αντίστροφο, δηλαδή πως αν ισχύει $g(n) = \Theta(f(n))$ τότε θα ισχύει και το ότι $f(n) = \Theta(g(n))$.

Ας ξεκινήσουμε από το $f(n) = \Theta(g(n))$. Για να δείξουμε ότι $g(n) = \Theta(f(n))$ θα πρέπει να αποδείξουμε ότι $g(n) = O(f(n))$ και $g(n) = \Omega(f(n))$.

- **Απόδειξη ότι $g(n) = O(f(n))$:**

Αφού $f(n) = \Theta(g(n))$, τότε $f(n) = \Omega(g(n))$, άρα υπάρχει $c_1 \in \mathbb{R}^+$ και $n_1 \geq 0$ τέτοιο ώστε:

$$f(n) \geq c_1 g(n) \Rightarrow g(n) \leq \frac{1}{c_1} f(n), \forall n \geq n_1$$

Άρα, υπάρχει $c = \frac{1}{c_1} \in \mathbb{R}^+$ τέτοιο ώστε:

$$g(n) \leq c f(n), \forall n \geq n_1$$

οπότε δείξαμε ότι $g(n) = O(f(n))$.

- **Απόδειξη ότι $g(n) = \Omega(f(n))$:**

Παρομοίως, αφού $f(n) = \Theta(g(n))$, τότε $f(n) = O(g(n))$, άρα υπάρχει $c_2 \in \mathbb{R}^+$ και $n_2 \in \mathbb{N}$ τέτοιο ώστε:

$$f(n) \leq c_2 g(n) \Rightarrow g(n) \geq \frac{1}{c_2} f(n), \forall n \geq n_2$$

Άρα, υπάρχει $c' = \frac{1}{c_2} \in \mathbb{R}^+$ τέτοιο ώστε:

$$g(n) \geq c' f(n), \forall n \geq n_2$$

οπότε δείξαμε ότι $g(n) = \Omega(f(n))$.

εφόσον λοιπόν $g(n) = O(f(n))$ και $g(n) = \Omega(f(n))$ προκύπτει το ζητούμενο $g(n) = \Theta(f(n))$.

b) **Αποδείξτε ότι $\lg(n!) = O(n \lg n)$.**

Πρέπει να αποδείξουμε ότι: $\lg(n!) = O(n \lg n)$ Με βάση τον ορισμό:

$$n! = 1 \cdot 2 \cdot 3 \cdots n \leq n \cdot n \cdot n \cdots n = n^n$$

άρα αν $n \geq 1$ θα έχουμε

$$\lg(n!) \leq \lg(n^n)$$

από τις ιδιότητες των λογαρίθμων $\lg(n^n) = n \cdot \lg(n)$ οπότε

$$\lg(n!) \leq n \cdot \lg(n).$$

Αν επιλέξω $c = 1$ και $n_0 = 2$ προκύπτει ότι

$$\lg(n!) \leq c \cdot n \cdot \lg(n), \forall n \geq n_0$$

που σύμφωνα με τον ορισμό συνεπάγεται ότι

$$\lg(n!) = O(n \cdot \lg n)$$

όπως απαιτείται.

c) **Ποια από τα ακόλουθα είναι αληθή και γιατί;**

(i) $n^3 + 4n^2 + 5n + 10 = \Theta(n^3)$

Για να δούμε αν το $n^3 + 4n^2 + 5n + 10 = \Theta(n^3)$ είναι αληθές θα χρησιμοποιήσουμε τον ορισμό του Θ . Για να είναι μια συνάρτησή $f(n)$ στο $\Theta(g(n))$ θα πρέπει να βρεθούν θετικές σταθερές c_1, c_2 και n_0 , τέτοια ώστε

$$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n), \quad \forall n \geq n_0$$

Αν θέσουμε $c_1 = 1$ και $n_1 = 0$ τότε $\forall n \geq n_1$

$$c_1 \cdot n^3 = 1 \cdot n^3 \leq n^3 + 4n^2 + 5n + 10,$$

το οποίο ισχύει για κάθε $n \geq 0$. Άρα για $c_1 = 1$ και $n_1 = 0$, η ανισότητα ισχύει, ομοίως $n^3 + 4n^2 + 5n + 10 \in \Omega(n^3)$.

Αν θέσουμε $c_2 = 20$ και $n_2 = 1$, τότε $\forall n \geq n_2$

$$n^3 + 4n^2 + 5n + 10 \leq n^3 + 4n^3 + 5n^3 + 10n^3 = 20 \cdot n^3 = c_2 \cdot n^3.$$

Άρα δείξαμε και ότι $n^3 + 4n^2 + 5n + 10 \in O(n^3)$,

οπότε σύμφωνα με τον ορισμό του Θ , ισχύει επομένως ότι

$$n^3 + 4n^2 + 5n + 10 \in \Theta(n^3)$$

(ii) $\lg(n^3) = \Theta(n \lg(n))$

Γνωρίζουμε ότι $\lg(n^3) = 3 \lg(n)$.

Θα προσπαθήσουμε πρώτα να δείξουμε ότι $3 \lg(n) \in O(n \lg(n))$.

Αναζητούμε $c_1 \in \mathbb{R}^+$ και $n_1 \geq 0$ ακέραιο τέτοιο ώστε

$$3 \lg(n) \leq c_1 n \lg(n), \forall n \geq n_1$$

αν θέσουμε $n_1 \geq 2$ τότε $\lg(n) > 0$ οπότε

$$3 \cdot \cancel{\lg(n)} \leq c_1 n \cdot \cancel{\lg(n)} \Rightarrow 3 \leq c_1 n \Rightarrow$$

$$c_1 \cdot n \geq 3, \quad \forall n \geq n_1, n_1 \geq 2,$$

που ισχύει αν θέσουμε $c_1 = 2$. Άρα δείξαμε ότι $\lg(n^3) \in O(n \lg(n))$.

Θα εξετάσουμε τώρα αν ισχύει το κάτω φράγμα $\Omega(n \lg(n))$, θα πρέπει να βρούμε $c_2 \in \mathbb{R}^+$ και $n_2 \in \mathbb{N}$ τ.ω.

$$3 \lg(n) \geq c_2 n \lg(n), \forall n \geq n_2$$

αν θέσουμε $n_2 \geq 2$ τότε $\lg(n) > 0$ οπότε

$$3 \cdot \cancel{\lg(n)} \geq c_1 n \cdot \cancel{\lg(n)} \Rightarrow 3 \geq c_2 n \Rightarrow$$

$$3 \geq c_2 \cdot n, \quad \forall n \geq n_2$$

το οποίο δεν ισχύει γιατί το c_2 θα πρέπει να μικραίνει όσο μεγαλώνει το n , άρα δεν υπάρχει σταθερά c_2 . Οπότε η $\lg(n^3) \notin \Omega(n \lg(n))$, άρα δεν καλύπτονται οι προϋποθέσεις, και $\lg(n^3) \notin \Theta(n \lg(n))$.

(iii) $\lg(\sqrt{n}) = \Theta(\lg(n))$

Θέλουμε να δείξουμε ότι $\lg(\sqrt{n}) \in \Theta(\lg(n))$, ή ισοδύναμα να δείξουμε ότι

$$\lg(\sqrt{n}) = \frac{1}{2} \lg(n) \in \Theta(\lg(n)).$$

Για να ισχύει $\frac{1}{2} \lg n = \Theta(\lg(n))$, πρέπει να βρούμε τις σταθερές $c_1, c_2 \in \mathbb{R}^+$ και $n_0 \in \mathbb{N}$ τέτοιες ώστε:

$$c_1 \lg n \leq \frac{1}{2} \lg n \leq c_2 \lg n, \quad \forall n \geq n_0$$

Είναι φανερό ότι η παραπάνω ανισότητα ισχύει αν επιλέξουμε $c_1 = \frac{1}{2}$ και $c_2 = \frac{1}{2}$, καθώς:

$$\frac{1}{2} \lg n \leq \frac{1}{2} \lg n \leq \frac{1}{2} \lg n$$

Αυτό ισχύει για όλες τις τιμές του $n \geq n_0$, όπου n_0 μπορεί να είναι οποιοδήποτε θετικός ακέραιος

Αφού βρέθηκαν οι σταθερές c_1 και c_2 , καταλήγουμε ότι:

$$\lg(\sqrt{n}) = \Theta(\lg n)$$

(iv) $\sum_{i=1}^n (2(i+1)) = O(n^3)$

Μια συνάρτηση $f(n)$ είναι $O(g(n))$ αν υπάρχουν σταθερές $c > 0$ και $n_0 \in \mathbb{N}$ τέτοιες ώστε:

$$f(n) \leq c \cdot g(n), \quad \forall n \geq n_0$$

Στην περίπτωσή μας, πρέπει να ελέγξουμε αν το άθροισμα $\sum_{i=1}^n (2(i+1))$ είναι στο $O(n^3)$.

Αρχικά, μπορούμε να αναδιατυπώσουμε το άθροισμα ως:

$$\sum_{i=1}^n (2(i+1)) = \sum_{i=1}^n 2(i+1) = 2 \sum_{i=1}^n (i+1)$$

Αυτό ισούται με:

$$\begin{aligned} & 2 \sum_{i=1}^n (i+1) = \\ & = 2 \left(\sum_{i=1}^n i + \sum_{i=1}^n 1 \right) = 2 [(1+2+3+4+\dots+n) + (1+1+1+1\dots+1)] = \\ & = 2 [(1+2+3+4+\dots+n) + n] = 2 \sum_{i=1}^n i + 2n = 2 \frac{n(n+1)}{2} + 2n = \end{aligned}$$

Απλοποιώντας την έκφραση, έχουμε:

$$n(n+1) + 2n = n^2 + n + 2n = n^2 + 3n$$

Άρα, το άθροισμα γίνεται:

$$\sum_{i=1}^n (2(i+1)) = n^2 + 3n$$

Τώρα, πρέπει να δείξουμε αν η συνάρτηση $n^2 + 3n$ είναι $O(n^3)$. Για μεγάλες τιμές του n , έχουμε:

$$n^2 + 3n \leq n^3, \quad \forall n \geq n_0$$

Αυτό είναι αληθές για μεγάλες τιμές του n , αφού $n^2 + 3n$ αυξάνεται πολύ πιο αργά από το n^3 .

Θέλουμε να βρούμε $c \in \mathbb{R}^+$ και $n_0 \in \mathbb{N}$ τέτοια ώστε

$$n^2 + 3n \leq c \cdot n^3, \quad \forall n \geq n_0$$

μεταφέρουμε όλους τους όρους στην αριστερή πλευρά της ανισότητας:

$$n^2 + 3n \leq c \cdot n^3 \implies c \cdot n^3 - n^2 - 3n \geq 0$$

γνωρίζουμε ότι

$$c \cdot n^3 - n^2 - 3n \geq c \cdot n^3 - n^2 - 3n^2 \geq 0 \implies$$

$$c \cdot n^3 - 4n^2 \geq 0 \implies n^2(cn - 4) \geq 0 \implies$$

$$(cn - 4) \geq 0 \implies cn \geq 4 \implies$$

το οποίο ισχύει αν θέσουμε $c = 1$ και $n \geq 4$.

Αυτό σημαίνει ότι η ανισότητα $n^3 - n^2 - 3n \geq 0$ ισχύει για $c = 1$ και $n \geq 4$.

Επομένως, μπορούμε να πούμε ότι:

$$\sum_{i=1}^n (2(i+1)) \leq 1 \cdot n^3, \quad \forall n \geq 4$$

αυτό ικανοποιεί τον ορισμό του

$$\sum_{i=1}^n (2(i+1)) = O(n^3)$$

(v) $\min(700, n^2) = \Theta(1)$

Θέλουμε να δείξουμε ότι $\min(700, n^2) \in \Theta(1)$.

Η συνάρτηση $\min(700, n^2)$ παίρνει δύο τιμές, ανάλογα με την τιμή του n :

- Όταν $n^2 \leq 700$, τότε $\min(700, n^2) = n^2$.
- Όταν $n^2 > 700$, τότε $\min(700, n^2) = 700$.

Για $n \leq \sqrt{700}$, η συνάρτηση είναι n^2 , ενώ για $n > \sqrt{700}$, η συνάρτηση γίνεται σταθερή και ίση με 700.

Για να δείξουμε ότι $\min(700, n^2) = \Theta(1)$, πρέπει να βρούμε θετικές σταθερές c_1, c_2 και n_0 τέτοιες ώστε:

$$c_1 \leq \min(700, n^2) \leq c_2, \quad \forall n \geq n_0$$

Για $n > \sqrt{700}$, για παράδειγμα $n_0 = \sqrt{700} + 1$, έχουμε:

$$\min(700, n^2) = 700, \forall n \geq n_0$$

Επιλέγοντας $c_1 = 699$, $c_2 = 701$ και $n_0 = \sqrt{700}$, έχουμε:

$$699 \cdot 1 \leq \min(700, n^2) \leq 701 \cdot 1, \quad \forall n \geq n_0$$

Αυτό ικανοποιεί τον ορισμό του $\Theta(1)$.

Άσκηση 3

a) Αποδείξτε επαγωγικά ότι αν $T(0) = 0$ και $T(n) = 2T(n-1) + 1$, $n > 0$, τότε

$$T(n) = 2^n - 1$$

.

b) Θεωρήστε τη συνάρτηση $f: \mathbb{N} \rightarrow \mathbb{N}$ που ορίζεται ως εξής:

$$f(0) = 1, f(1) = 2 \text{ και } f(n) = 4 \cdot f(n-2) + 2^n, \text{ αν } n > 1$$

Αποδείξτε επαγωγικά ότι για κάθε ακέραιο $n \geq 3$, $f(n) \leq 3n \cdot 2^{n-2}$.

Λύση:

a) Με επαγωγή ως προς n .

Βάση επαγωγής ($n=1$):

Για $n = 1$, η αναδρομική σχέση μας δίνει:

$$T(1) = 2 \cdot T(0) + 1 = 2 \cdot 0 + 1 = 1.$$

Επιπλέον, ισχύει $2^1 - 1 = 2 - 1 = 1 = T(1)$, όπως απαιτείται.

Επαγωγική Υπόθεση:

Έστω κάποιο $n > 1$. Υποθέτουμε ότι ο ισχυρισμός ισχύει για $n-1$, δηλαδή ότι $T(n-1) = 2^{n-1} - 1$.

Επαγωγικό Βήμα:

Θα δείξουμε ότι ο ισχυρισμός ισχύει για την τιμή n , δηλαδή ότι $T(n) = 2^n - 1$. Από την αναδρομική σχέση, έχουμε:

$$T(n) = 2 \cdot T(n-1) + 1.$$

Από την επαγωγική υπόθεση, έχουμε ότι $T(n-1) = 2^{n-1} - 1$. Αντικαθιστώντας στην παραπάνω σχέση, προκύπτει ότι:

$$T(n) = 2 \cdot (2^{n-1} - 1) + 1 = 2^n - 2 + 1 = 2^n - 1.$$

Άρα, ο ισχυρισμός ισχύει για $T(n)$.

b) **Επαγωγική Απόδειξη για $f(n)$**

Βάση επαγωγής ($n=3$):

Για $n = 3$, από την αναδρομική σχέση, προκύπτει:

$$f(3) = 4 \cdot f(1) + 2^3 = 4 \cdot 2 + 8 = 16.$$

Επιπλέον, ισχύει:

$$3 \cdot 3 \cdot 2^1 = 9 \cdot 2 = 18 \geq 16 = f(3).$$

Άρα, ο ισχυρισμός ισχύει για $n = 3$.

Επαγωγική υπόθεση: Θεωρούμε οποιονδήποτε ακέραιο $n > 3$ και υποθέτουμε ότι ο ισχυρισμός ισχύει για κάθε τιμή n' τέτοια ώστε $3 \leq n' < n$, δηλαδή:

$$f(n') \leq 3 \cdot n' \cdot 2^{n'-2}, \quad \forall n' \text{ με } 3 \leq n' < n.$$

Επαγωγικό βήμα: Θα δείξουμε ότι ο ισχυρισμός ισχύει για n . Από την αναδρομική σχέση, προκύπτει ότι:

$$f(n) = 4 \cdot f(n-2) + 2^n.$$

Διακρίνουμε περιπτώσεις.

Περίπτωση 1: $n = 4$. Στην περίπτωση αυτή, ισχύει ότι:

$$f(4) = 4 \cdot f(2) + 2^4 = 4 \cdot f(2) + 16.$$

Από την αναδρομική σχέση προκύπτει ότι:

$$f(2) = 4 \cdot f(0) + 2^2 = 4 \cdot 1 + 4 = 8.$$

Επομένως, $f(4) = 4 \cdot 8 + 16 = 48$.

Επιπρόσθετα, ισχύει ότι:

$$3 \cdot 4 \cdot 2^2 = 12 \cdot 4 = 48 \geq f(4).$$

Επομένως, ο ισχυρισμός ισχύει σε αυτή την περίπτωση. (Είναι αξιοσημείωτο ότι σε αυτή την περίπτωση, δεν μπορούμε να εφαρμόσουμε την επαγωγική υπόθεση για $n' = 2$, αφού η επαγωγική υπόθεση ισχύει μόνο για $n' \geq 3$).

Περίπτωση 2: $n > 4$.

Από (3), έχουμε:

$$f(n) = 4 \cdot f(n-2) + 2^n$$

$$f(n) \leq 4 \cdot (3 \cdot (n-2) \cdot 2^{n-4}) + 2^n \quad (\text{από επαγωγική υπόθεση, όπου } n' = n-2)$$

$$f(n) \leq 2^2 \cdot (3 \cdot (n-2) \cdot 2^{n-4}) + 2^2 \cdot 2^{n-2}$$

$$f(n) \leq 3 \cdot (n-2) \cdot 2^{n-2} + 4 \cdot 2^{n-2} \quad (\text{από ιδιότητες δυνάμεων όπου } 2^2 \cdot 2^{n-4} = 2^{n-2})$$

$$f(n) \leq 3 \cdot n \cdot 2^{n-2} - 6 \cdot 2^{n-2} + 4 \cdot 2^{n-2}$$

$$f(n) \leq 3 \cdot n \cdot 2^{n-2} - 2 \cdot 2^{n-2}$$

$$f(n) \leq 3 \cdot n \cdot 2^{n-2} - 2 \cdot 2^{n-2} < 3 \cdot n \cdot 2^{n-2},$$

$$\text{άρα } f(n) \leq 3 \cdot n \cdot 2^{n-2},$$

όπως απαιτείται. (Είναι αξιοσημείωτο ότι, αφού $n > 4$ σε αυτή την περίπτωση, ισχύει ότι $n-2 \geq 3$ και άρα μπορούμε να εφαρμόσουμε την επαγωγική υπόθεση).

Άσκηση 4

Βρείτε την τάξη (βάσει των συμβολισμών O , Ω και Θ) της χρονικής πολυπλοκότητας $T(n)$ των ακόλουθων αλγορίθμων:

a) Procedure Mystery(integer n) {

```
    for (i = 1; i <= n; i++) {  
        for (j = i + 1; j <= n; j++)  
            x = x + 1  
    }
```

b) Procedure Puzzle(integer n) {

```
    for (i = 1; i <= n; i++)  
        for (k = n; k <= n + 5; k++)  
            x = x + 1  
}
```

Λύση:

a) Procedure Mystery(integer n)

Ας πάρουμε για παράδειγμα μια ιχνηλάτηση της εκτέλεσης του αλγορίθμου για n :

Εξωτερικός βρόγχος $i = 1$:

Εσωτερικός βρόγχος ($j = 2$ έως n):

$j = 2, x = x + 1$

$j = 3, x = x + 1$

.

.

.

$j = n, x = x + 1$

(τέλος εσωτερικού βρόγχου) ($n-1$ εκτελέσεις)

Εξωτερικός βρόγχος $i = 2$:

Εσωτερικός βρόγχος ($j = 3$ έως n):

$j = 3, x = x + 1$

.

.

.

$j = n, x = x + 1$

(τέλος εσωτερικού βρόγχου) ($n-2$ εκτελέσεις)

.

.

.

Εξωτερικός βρόγχος $i = n - 1$:

Εσωτερικός βρόγχος ($j = n$):

$j = n, x = x + 1$

(τέλος εσωτερικού βρόγχου) (1 εκτέλεση)

Εξωτερικός βρόγχος $i = n$:

Εσωτερικός βρόγχος δεν εκτελείται ($j = n + 1 > n$)

Ο εξωτερικός βρόγχος $for(i = 1; i \leq n; i++)$ εκτελείται n φορές, με το i να παίρνει τις τιμές $1, 2, \dots, n$. Για κάθε τιμή του i , ο εσωτερικός βρόγχος $for(j = i+1; j \leq n; j++)$ εκτελείται $n - i$ φορές. Επομένως, για κάθε τιμή του i , η εντολή $x = x + 1$ εκτελείται $n - i$ φορές.

Η συνολική χρονική πολυπλοκότητα $T(n)$ προκύπτει από το άθροισμα των εκτελέσεων της εντολής $x = x + 1$ για κάθε i , δηλαδή:

$$T(n) = \sum_{i=1}^n (n - i) = (n - 1) + (n - 2) + \dots + 1 + 0$$

Αυτό το άθροισμα είναι άθροισμα όρων αριθμητικής προόδου (με βήμα 1), το οποίο είναι γνωστό ότι ισούται με:

$$T(n) = \frac{(n - 1) \cdot n}{2}$$

Αφού ο κυρίαρχος όρος είναι $n^2/2$, θα αποδείξουμε ότι η χρονική πολυπλοκότητα είναι:

$$T(n) = O(n^2)$$

Επομένως, θα αποδείξουμε ότι:

$$\frac{n(n - 1)}{2} = O(n^2)$$

Αρχικά, μπορούμε να αναπτύξουμε την παράσταση $\frac{n(n-1)}{2}$:

$$f(n) = \frac{n(n - 1)}{2} = \frac{n^2 - n}{2} = \frac{n^2}{2} - \frac{n}{2}$$

Σύμφωνα με τον ορισμό του $O(n^2)$, πρέπει να βρούμε μία σταθερά $c_1 > 0$ και ένα $n_1 \geq 0$ τέτοια ώστε:

$$f(n) \leq c_1 \cdot n^2, \quad \forall n \geq n_1$$

$$0 \leq c_1 \cdot n^2 - \frac{n^2}{2} + \frac{n}{2}, \quad \forall n \geq n_1$$

$$0 \leq \left(c_1 - \frac{1}{2}\right) \cdot n^2 + \frac{n}{2}, \quad \forall n \geq n_1$$

Η οποία ισχύει για τιμές $c_1 \geq \frac{1}{2}, \forall n \geq 0$

Άρα, σύμφωνα με τον ορισμό του O , ισχύει ότι:

$$T(n) = \frac{n(n - 1)}{2} \in O(n^2)$$

Θέλουμε επίσης να δείξουμε ότι:

$$\frac{n(n-1)}{2} = \Omega(n^2)$$

αναζητώ μία σταθερά $c_2 > 0$ και ένα $n_2 \geq 0$ τέτοια ώστε:

$$f(n) \geq c_2 \cdot n^2, \quad \forall n \geq n_2$$

$$\frac{n^2}{2} - \frac{n}{2} \geq c_2 \cdot n^2, \quad \forall n \geq n_2$$

$$0 \leq \left(\frac{1}{2} - c_2\right) \cdot n^2 - \frac{n}{2}, \quad \forall n \geq n_2$$

$$0 \leq \left(\left(\frac{1}{2} - c_2\right) \cdot n - \frac{1}{2}\right) n, \quad \forall n \geq n_2$$

$$0 \leq \left(\frac{1}{2} - c_2\right) \cdot n - \frac{1}{2}, \quad \forall n \geq n_2$$

αν θέσω $c_2 = \frac{1}{4}$

$$0 \leq \left(\frac{1}{2} - \frac{1}{4}\right) \cdot n - \frac{1}{2}, \quad \forall n \geq n_2$$

$$0 \leq \frac{1}{4} \cdot n - \frac{1}{2}, \quad \forall n \geq n_2$$

$$\frac{1}{2} \leq \frac{n}{4}, \quad \forall n \geq n_2$$

$$2 \leq n,$$

άρα αν θέσω $n_2 = 2$ και $c_2 = \frac{1}{4}$ η ανισότητα ισχύει, άρα

$$T(n) \in \Omega(n^2).$$

Επομένως, αν θέσω $n_0 = \max(n_1, n_2) = \max(0, 2) = 2$ και $c_1 = \frac{1}{2}$ και $c_2 = \frac{1}{4}$ τότε

$$c_1 \cdot n^2 \leq T(n) \leq c_2 \cdot n^2, \quad \forall n \geq 2$$

οπότε έχω αποδείξει ότι

$$T(n) \in \Theta(n^2)$$

b) Procedure Puzzle(integer n)

Το i θα πάρει n διαφορετικές τιμές ($i = 1, 2, \dots, n$). Για κάθε μία από αυτές τις τιμές, θα εκτελεστεί ο εσωτερικός for βρόγχος. Άρα, ο εσωτερικός for βρόγχος θα εκτελεστεί συνολικά n φορές. Κάθε φορά που εκτελείται ο εσωτερικός for βρόγχος, η μεταβλητή k παίρνει 6 διαφορετικές τιμές ($k = n, n + 1, n + 2, n + 3, n + 4, n + 5$). Επομένως, κάθε φορά που εκτελείται ο εσωτερικός for βρόγχος, η εντολή $x = x + 1$ εκτελείται 6 φορές. Συμπεραίνουμε πως ο συνολικός αριθμός φορών που θα εκτελεστεί η εντολή $x = x + 1$ είναι $6 \cdot n$. Άρα, η χρονική πολυπλοκότητα $T(n)$ της *Puzzle()* είναι $T(n) = O(n)$.

Συνοπτική ιχνηλάτηση της εκτέλεσης της *Puzzle()* παρουσιάζεται στη συνέχεια.

Εξωτερικό for loop (ανακύκλωση $i = 1$):

(Εσωτερικό for loop:)

$k = n$

$k = n + 1$

$k = n + 2$

$k = n + 3$

$k = n + 4$

$k = n + 5$

(τέλος εκτέλεσης εσωτερικού for loop)

Εξωτερικό for loop (ανακύκλωση $i = 2$):

(Εσωτερικό for loop:)

$k = n$

$k = n + 1$

$k = n + 2$

$k = n + 3$

$k = n + 4$

$k = n + 5$

(τέλος εκτέλεσης εσωτερικού for loop)

...

Εξωτερικό for loop (ανακύκλωση $i = n$):

(Εσωτερικό for loop:)

$k = n$

$k = n + 1$

$k = n + 2$

$k = n + 3$

$k = n + 4$

$k = n + 5$

(τέλος εκτέλεσης εσωτερικού for loop)

(Τέλος εκτέλεσης εξωτερικού for loop)

Ευχαριστίες

Ευχαριστίες στον Αθανάσιο Ζαχαρόπουλο, αποσπασμένο εκπαιδευτικό του Τμήματος Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης, για την παραγωγή της ηλεκτρονικής έκδοσης του παραπάνω υλικού και τη συνεισφορά του σε κάποιες από τις λύσεις.