Κατ'οίκον Εργασία 1 – Σκελετοί Λύσεων

Άσκηση 1

Αφού ξέρουμε με ακρίβεια τον αριθμό των βασικών πράξεων που εκτελεί ο κάθε αλγόριθμος σε δεδομένα μεγέθους n, θα υπολογίσουμε τον χρόνο εκτέλεσης που ζητείται στα i και ii αντικαθιστώντας όπου n με $2n$ και n+1 αντίστοιχα

(α) i. $(2n)^2 = 4n^2$ $\Rightarrow$ Ο χρόνος εκτέλεσης του αλγορίθμου θα τετραπλασιαστεί.

ii. $(n+1)^2 = n^2 + 2n + 1$ $\Rightarrow$ Θα έχουμε $2n + 1$ περισσότερες πράξεις.

(β) i. $(2n)^3 = 8n^3$ $\Rightarrow$ Ο χρόνος εκτέλεσης του αλγορίθμου θα οκταπλασιαστεί.

ii. $(n+1)^3 = n^3 + 3n^2 + 3n + 1$ $\Rightarrow$ Θα έχουμε $3n^2 + 3n + 1$ περισσότερες πράξεις.

(γ) i. $100\cdot(2n)^2 = 100\cdot 4n^2$ $\Rightarrow$ Ο χρόνος εκτέλεσης θα τετραπλασιαστεί.

ii. $100\cdot(n+1)^2 = 100n^2+200n+100$ $\Rightarrow$ Θα έχουμε $200n+100$ περισσότερες πράξεις.

(δ) i. $2n\cdot\lg 2n = 2n\cdot(\lg 2 + \lg n) = 2n\cdot(1 + \lg n) = 2n + 2n\cdot\lg n$

Θα έχουμε $2n + n\cdot\lg n$ περισσότερες πράξεις.

ii. $(n+1)\cdot\lg(n+1) = n\cdot\lg(n+1) + \lg(n+1)$

Θα έχουμε $n\cdot(\lg(n+1) - \lg n) + \lg(n+1)$ περισσότερες πράξεις.

(ε) i. $2^{2n} = (2^n)^2$ $\Rightarrow$ Ο χρόνος εκτέλεσης θα υψωθεί στο τετράγωνο.

ii. $2^{n+1} = 2^n\cdot 2^1$ $\Rightarrow$ Ο χρόνος εκτέλεσης θα διπλασιαστεί.

(ζ) i. $2^{2^{2n}}$ $\Rightarrow$ Ο χρόνος εκτέλεσης θα πολλαπλασιαστεί επί $2^{2^{2n}-2^n}$.

ii. $2^{2^{n+1}} = 2^{2\cdot 2^n} = (2^{2^n})^2$ $\Rightarrow$ Ο χρόνος εκτέλεσης θα υψωθεί στο τετράγωνο.

Άσκηση 2

Πιο κάτω παρουσιάζεται η ζητούμενη σειρά των συναρτήσεων από τη μικρότερη τάξη προς τη μεγαλύτερη.

$$f_1(n) = 4\sqrt{n}$$
$$f_3(n) = n^{2.5}$$
$$f_2(n) = n^3 - 3\cdot n^{2.5}$$

Πιο κάτω αποδεικνύουμε τις σχέσεις που υπάρχουν ανάμεσα στις συναρτήσεις.

$\underline{4\sqrt{n} \in O(n^{2.5})}$

Πρέπει να δείξουμε ότι υπάρχουν σταθερές c και n_0 τέτοιες ώστε

$4\sqrt{n} \le c \cdot n^{2.5}$ για κάθε $n \ge n_0$, δηλαδή

$4n^{0.5} \le c \cdot n^{2.5}$ για κάθε $n \ge n_0$, δηλαδή

$4 \le c \cdot n^2$ για κάθε $n \ge n_0$

Προφανώς για $c = 4$ και $n_0 = 1$ η πρόταση ισχύει.

$n^{2.5} \notin O(4\sqrt{n})$

Έστω ότι ισχύει, δηλαδή $n^{2.5} \in O(4\sqrt{n})$. Τότε

$\exists\ c, n_0$ τ.ω. $n^{2.5} \le c \cdot 4\sqrt{n}$ για κάθε $n \ge n_0$, δηλαδή

$\sqrt{n} \cdot n^2 \le c \cdot 4\sqrt{n}$ για κάθε $n \ge n_0$, δηλαδή

$n^2 \le 4c$ για κάθε $n \ge n_0$

Δεν είναι δυνατόν να υπάρχει θετική σταθερά c που να ικανοποιεί την πιο πάνω σχέση. Αυτό μας οδηγεί σε αντίφαση και συμπεραίνουμε ότι $n^{2.5} \notin O(4\sqrt{n})$

$n^{2.5} \in O(n^3 - 3 \cdot n^{2.5})$

Πρέπει να δείξουμε ότι υπάρχουν σταθερές c και n_0 τέτοιες ώστε

$n^{2.5} \le c \cdot (n^3 - 3 \cdot n^{2.5})$, για κάθε $n \ge n_0$, δηλαδή

$1 \le c \cdot (n^{0.5} - 3)$, για κάθε $n \ge n_0$, δηλαδή

$$\frac{1}{\sqrt{n} - 3} \le c, \text{ για κάθε } n \ge n_0$$

Έστω ότι $c = 1$. Τότε έχουμε

$$\frac{1}{\sqrt{n} - 3} \le 1 \Rightarrow 1 \le \sqrt{n} - 3 \Rightarrow 4 \le \sqrt{n} \Rightarrow 16 \le n.$$

Κατά συνέπεια, το ζητούμενο ισχύει για $c = 1$ και $n_0 = 16$.

$n^3 - 3 \cdot n^{2.5} \notin O(n^{2.5})$

Έστω ότι ισχύει, δηλαδή $n^3 - 3 \cdot n^{2.5} \in O(n^{2.5})$. Τότε

$\exists\ c, n_0$ τ.ω. $n^3 - 3 \cdot n^{2.5} \le c \cdot n^{2.5}$ για κάθε $n \ge n_0$, δηλαδή

$n^3 \le (c + 3) \cdot n^{2.5}$ για κάθε $n \ge n_0$, δηλαδή

$\sqrt{n} \le c + 3$ για κάθε $n \ge n_0$

Δεν είναι δυνατόν να υπάρχει θετική σταθερά c που να ικανοποιεί την πιο πάνω σχέση, αφού καθώς το n τείνει στο ∞ το $\sqrt{n} - 3$ επίσης τείνει στο ∞. Αυτό μας οδηγεί σε αντίφαση και συμπεραίνουμε ότι $n^3 - 3 \cdot n^{2.5} \notin O(n^{2.5})$.

Άσκηση 3

(i) Η πρόταση δεν είναι αληθής όπως φαίνεται στο πιο κάτω αντιπαράδειγμα.

$$f(n) = 5n, \qquad f'(n) = n$$

$$g(n) = 5n^3, \qquad g'(n) = n^3$$

Παρατηρούμε ότι ενώ $5n \in \Theta(n)$ και $5n^3 \in \Theta(n^3)$, $5n + 5n^3 \notin \Theta(n)$

(ii) Η πρόταση δεν είναι αληθής όπως φαίνεται στο πιο κάτω αντιπαράδειγμα.

$$f(n) = n^2,$$

$$g(n) = \begin{cases} n & \text{αν } n \bmod 2 = 1 \\ n^3 & \text{αν } n \bmod 2 = 0 \end{cases}$$

Έστω ότι $f(n) \in \Omega(g(n))$. Τότε πρέπει να υπάρχουν σταθερές c και n_0, τέτοιες ώστε

$$f(n) \geq c \cdot g(n) \qquad \text{για κάθε } n \geq n_0$$

δηλαδή,

$$n^2 \geq c \cdot n \qquad \text{για κάθε περιττό } n \geq n_0 \qquad (1)$$

και

$$n^2 \geq c \cdot n^3 \qquad \text{για κάθε άρτιο } n \geq n_0 \qquad (2)$$

Παρατηρούμε ότι δεν υπάρχουν σταθερές c και n_0 που να ικανοποιούν την πρόταση (2) ($n^2 \geq c \cdot n^3 \Leftrightarrow 1 \geq cn$ και δεν υπάρχει σταθερά $c > 0$ μικρότερη του $1/n$ για κάθε n).

Παρόμοια, μπορούμε να δούμε ότι $g(n) \notin \Omega(f(n))$ και το ζητούμενο έπεται.

(iii) Η πρόταση είναι αληθής. Ακολουθεί η απόδειξη:

$$\sum_{i=1}^{n} i^k = 1^k + 2^k + 3^k + \dots\, n^k \leq n^k + n^k + n^k + \dots + n^k = n \,.\, n^k = n^{k+1}$$

Αφού $\sum_{i=1}^{n} i^k \leq n^{k+1}$ τότε $\sum_{i=1}^{n} i^k \in O(n^{k+1})$ για c=1 και n_0=1

Άσκηση 4

A. Έστω T(n) ο χρόνος εκτέλεσης της διαδικασίας selectionsort. Τότε,

$$T(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1 = \sum_{i=1}^{n-1} (n-i) = \sum_{i=1}^{n-1} n - \sum_{i=1}^{n-1} i$$

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

Β. Για να βρούμε τον χρόνο εκτέλεσης της διεργασίας, παρατηρούμε ότι ο χρόνος εκτέλεσης του μεσαίου βρόχου εξαρτάται από την τιμή του i, η οποία καθορίζεται από τον εξωτερικό βρόχο. Θεωρούμε χωρίς βλάβη της γενικότητας ότι το $2n$ είναι δύναμη του 2. Ο πιο κάτω πίνακας δείχνει τον χρόνο εκτέλεσης του μεσαίου βρόχου, $T_{2,3}(i)$, σαν συνάρτηση του i:

i	$2n$	n	$n/2$	$n/4$	…	2	1
$T_{2,3}(i)$	n^2	$n^2/2$	$n^2/4$	$n^2/8$	…	n	0

Ο συνολικός χρόνος εκτέλεσης της διεργασίας, T_{total}, είναι ίσος με το άθροισμα των χρόνων εκτέλεσης, $T_{2,3}(i)$, δηλαδή

$$T_{total} = n^2 + n^2/2 + n^2/4 + n^2/8 + \ldots + n = n\cdot(n + n/2 + n/4 + n/8 + \ldots + 1)$$
$$= n(2n\text{-}1) \in \Theta(n^2)$$

Γ. Για να βρούμε τον χρόνο εκτέλεσης του πρώτου βρόχου παρατηρούμε ότι ο χρόνος εκτέλεσης του εσωτερικού for εξαρτάται από την τιμή του j, η οποία καθορίζεται από τον μεσαίο βρόχο. Θεωρούμε χωρίς βλάβη της γενικότητας ότι το $\sqrt{n}$ είναι δύναμη του 2. Ο πιο κάτω πίνακας δείχνει τον χρόνο εκτέλεσης του εσωτερικού for, $T_3(j)$, σαν συνάρτηση του j:

j	1	2	4	…	$\sqrt{n}$
$T_3(j)$	1	2	4	…	$\sqrt{n}$

Ο συνολικός χρόνος εκτέλεσης της μεσαίου for, T_2, είναι ίσος με το άθροισμα των χρόνων εκτέλεσης, $T_3(j)$, δηλαδή

$$T_2 = 1 + 2 + 4 + \ldots + \sqrt{n} = \sqrt{n}\cdot(2\sqrt{n}\text{-}1)$$

Επομένως ο χρόνος εκτέλεσης του πρώτου βρόχου είναι $\sqrt{n} \cdot [\sqrt{n}\cdot(2\sqrt{n}\text{-}1)] \in \Theta(n\sqrt{n})$

Σε αυτό τον χρόνο θα πρέπει να προσθέσουμε τον χρόνο εκτέλεσης του δεύτερου βρόχου. Υποθέτουμε, χωρίς βλάβη της γενικότητας, ότι ο n είναι πολλαπλάσιο του 3. Για να βρούμε τον χρόνο εκτέλεσης του else, παρατηρούμε ότι ο χρόνος εκτέλεσης του εσωτερικού βρόχου εξαρτάται από την τιμή i, η οποία καθορίζεται από τον εξωτερικό βρόχο. Ο πιο κάτω πίνακας δείχνει τον χρόνο εκτέλεσης του εσωτερικού βρόχου, $T_{εσ}(i)$, σαν συνάρτηση του i:

i	1	2	3	4	5	6	…	n
$T_{εσ}(i)$	0	0	3	0	0	6	…	n

Συνεπώς, ο χρόνος εκτέλεσης του εξωτερικού βρόχου δίνεται από το πιο κάτω άθροισμα:

$$3+6+9+...+n = 3\cdot(1+2+3+...+n/3)$$
$$= 3\sum_{i=1}^{n/3} i = 3\frac{(n/3)(n/3+1)}{2} \in O(n^2)$$

Συμπεραίνουμε ότι ο συνολικός χρόνος εκτέλεσης της διεργασίας είναι της τάξης $\Theta(n^2)$.

Άσκηση 5

Έστω $T(n)$ ο χρόνος εκτέλεσης της recursive(n). Τότε η τιμή του $T(n)$ δίνεται από την πιο κάτω αναδρομική εξίσωση:

$$T(0) = 1$$
$$T(1) = 1$$
$$T(n) = T(n-1) + n^2$$

(Το n^2 προκύπτει από την ανάλυση που έγινε στην Άσκηση 4(Γ).) Θα λύσουμε την αναδρομική εξίσωση με τη μέθοδο της αντικατάστασης:

$$\begin{aligned} T(n) &= T(n-1) + n^2 \\ &= T(n-2) + (n-1)^2 + n^2 \\ &= \dots \\ &= T(n-i) + (n-i+1)^2 + \dots + (n-2)^2 + (n-1)^2 + n^2 \end{aligned}$$

Θέτουμε $k = n - 1$ και έχουμε ότι:

$$\begin{aligned} T(n) &= T(n-k) + (n-k+1)^2 + \dots + (n-2)^2 + (n-1)^2 + n^2 \\ &= T(1) + 2^2 + \dots + n^2 \\ &= \frac{n(n+1)(2n+1)}{6} \in \Theta(n^3) \end{aligned}$$