
352. ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Ασκήσεις

Εαρινό Εξάμηνο 2024

Γιάννης Λιβιεράτος
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Σχολή Θετικών Επιστημών, Τμήμα Μαθηματικών

Το αρχείο θα τροποποιείται και θα ενημερώνεται με λύσεις των προτεινόμενων ασκήσεων και νέες ασκήσεις ανά τακτά χρονικά διαστήματα.

Contents

1	Εισαγωγικά	3
2	Ασυμπτωτικός Συμβολισμός	3
3	Πίνακες	3
4	Αλγόριθμοι	4
4.1	Αλγόριθμοι Αναζήτησης	4
4.2	Αλγόριθμοι Ταξινόμησης	4

1 Εισαγωγικά

Ερώτηση 1.1. Η χρήση κατάλληλων δομών δεδομένων μπορεί να:

1. επηρεάσει την μνήμη που δεσμεύεται από ένα πρόγραμμα,
2. βελτιώσει την ταχύτητα εκτέλεσης ενός αλγορίθμου,
3. το είδος των αλγορίθμων που μπορούν να χρησιμοποιηθούν για την επίλυση ενός προβλήματος,
4. επιβάλει την γλώσσα προγραμματισμού που πρέπει να χρησιμοποιηθεί για την υλοποίηση ενός αλγορίθμου.

Εξηγήστε γιατί ισχύει ή όχι κάθε ένα από τα παραπάνω.

Ερώτηση 1.2. Η ψευδογλώσσες έχουν ένα προφανές μειονέκτημα (που φαίνεται και στην ίδια την λέξη). Τι πλεονεκτήματα έχουν έναντι της φυσικής γλώσσας και των γλωσσών προγραμματισμού; Ποια χαρακτηριστικά τους καθιστούν ιδιαίτερα δύσχρηστες τις γλώσσες μηχανής και τις συμβολικές (Assembly) γλώσσες για μάθημα θεωρητικής πληροφορικής;

2 Ασυμπτωτικός Συμβολισμός

Άσκηση 2.1. Έστω $f, g : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ με $f(n) = O(g(n))$. Να δείξετε ότι:

- (i) $2f(n) = O(g(n))$.
- (ii) $f(n) = O(\frac{1}{3} \cdot g(n))$.
- (iii) $cf(n) = O(g(n))$, για κάθε σταθερά $c > 0$.

Άσκηση 2.2. Έστω $f, g, h : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ με $h(n) := \max\{f(n), g(n)\}$, για κάθε $n \in \mathbb{N}$. Να δείξετε ότι:

$$f(n) + g(n) = O(h(n)).$$

Άσκηση 2.3. Έστω $f, g : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ με $f(n) = O(g(n))$ και $g(n) = O(f(n))$. Ισχύει ότι $f(n) = g(n)$, για κάθε $n \in \mathbb{N}$; Αν ναι, να αποδειχθεί, αλλιώς να βρεθεί αντιπαράδειγμα.

Άσκηση 2.4. Να δειχθούν τα ακόλουθα:

- $n \log_{10} n = O(n^2)$.
- $n + 5 = O(n - 7)$. Μπορείτε να βρείτε τη βέλτιστη σταθερά για την οποία ισχύει αυτό;
- $3n^3 + 2n^2 + n - 15 = O(n^3)$. Μπορείτε να βρείτε τη βέλτιστη σταθερά για την οποία ισχύει αυτό;
- $n^{100} = O(2^n)$ και το ανάποδο δεν ισχύει.
- $2^n = O(3^n)$ και το ανάποδο δεν ισχύει.

Άσκηση 2.5. Έστω $f : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$, με:

$$f(n) = \sum_{i=0}^n i := 0 + 1 + \dots + n.$$

Να βρεθεί κατάλληλη $g : \mathbb{N} \rightarrow \mathbb{R}$ και να αποδειχθεί ότι $f(n) = O(g(n))$.

3 Πίνακες

Ερώτηση 3.1. Έστω πίνακες A, B , διαστάσεων $p \times q$ και $q \times r$ αντίστοιχα. Πόσοι πολλαπλασιασμοί επιμέρους στοιχείων θα πρέπει να εκτελεστούν κατά τον πολλαπλασιασμό $A \times B$;

Ερώτηση 3.2. Έστω πίνακες A, B, C διαστάσεων $10 \times 3, 3 \times 5, 5 \times 10$ αντίστοιχα. Για την εύρεση του γινομένου ABC :

- να βρούμε το AB και μετά να πολλαπλασιάσουμε με C ,
- να βρούμε το BC και μετά να πολλαπλασιάσουμε με A ,
- δεν έχει σημασία.

4 Αλγόριθμοι

Ερώτηση 4.1. Πόσες φορές θα εκτελεστεί η εντολή *print(.)* στον παρακάτω ψευδοκώδικα;

Algorithm 1 Repeat

```
1: for  $i = 2, 4, \dots, 2n$  do
2:   for  $j = 3n, 3n - 1, \dots, 2n + 1$  do
3:     for  $k = i, \dots, j$  do
4:       print("Oh dear...")
5:     end for
6:   end for
7: end for
```

Μπορούμε να μιλήσουμε για πολυπλοκότητα στον αλγόριθμο, από την στιγμή που δεν δέχεται κάποια είσοδο;

Ερώτηση 4.2. Πόσες φορές θα εκτελεστεί η εντολή *print(.)* στον παρακάτω ψευδοκώδικα;

Algorithm 2 Repeat?

```
1:  $j = 2$ 
2: for  $i = 1, 2, \dots, n$  do
3:    $j = j + 2$ 
4: end for
5: while  $j < 2n$  do
6:   print("Emptiness...")
7:    $j ++$ 
8: end while
```

Ερώτηση 4.3. Πόσες φορές θα εκτελεστεί η εντολή *print(.)* στον παρακάτω ψευδοκώδικα;

Algorithm 3 BetterUseFor

```
1:  $j = 1$ 
2: while  $j < n$  do
3:   print("...")
4: end while
```

4.1 Αλγόριθμοι Αναζήτησης

Άσκηση 4.1. Να γραφεί αλγόριθμος που να υπολογίζει το δεύτερο μεγαλύτερο στοιχείο σε πίνακα διαστάσεων $n \times m$.

Άσκηση 4.2. Να γραφεί αλγόριθμος που να υπολογίζει το τρίτο μεγαλύτερο στοιχείο σε πίνακα διαστάσεων $p \times q \times r$.

4.2 Αλγόριθμοι Ταξινόμησης

Άσκηση 4.3. Να γραφεί αλγόριθμος που, δοθέντων τριών μονοδιάστατων πινάκων ακέραιων αριθμών με τα στοιχεία τους σε αύξουσα σειρά, να δημιουργεί πίνακα που να περιλαμβάνει τα στοιχεία και των τριών πινάκων, σε αύξουσα σειρά, χωρίς επαναλήψεις και να υπολογισθεί η πολυπλοκότητά του.

Αν έπρεπε να έχουμε εξαρχής δηλωμένη την διάσταση του τελικού πίνακα, τι θα μπορούσαμε να κάνουμε σε περίπτωση που υπάρχουν στοιχεία που επαναλαμβάνονται;

Άσκηση 4.4. Να γραφεί αλγόριθμος ταξινόμησης σε φθίνουσα σειρά πίνακα μη αρνητικών ακέραιων αριθμών. Δίνεται ότι το μήκος του πίνακα είναι n , το μέγιστο στοιχείο M , και ζητείται αλγόριθμος πολυπλοκότητας $O(n + M)$.