

## Κεφάλαιο 5

1. Τι ονομάζουμε "Χειρότερη περίπτωση ενός αλγορίθμου" ;

Η **χειρότερη περίπτωση** αντιπροσωπεύει τις τιμές εκείνες, που όταν δίνονται ως είσοδος στον αλγόριθμο, οδηγούν στην εκτέλεση μέγιστου αριθμού βασικών πράξεων. (σ.89)

2. Πώς αναγνωρίζουμε συνήθως τη χειρότερη περίπτωση του αλγορίθμου;

Η πλέον συνηθισμένη πρακτική είναι η μέτρηση του αριθμού των βασικών πράξεων που θα πρέπει να εκτελέσει ο αλγόριθμος στη χειρότερη περίπτωση. (σ.89)

3. Πώς εκφράζεται η χειρότερη περίπτωση του αλγορίθμου;

Για να εκφρασθεί αυτή η χειρότερη περίπτωση, χρειάζεται κάποιο μέγεθος σύγκρισης και αναφοράς που να χαρακτηρίζει τον αλγόριθμο. Αφορά στο μέγιστο κόστος εκτέλεσης του αλγορίθμου, κόστος που μετράται σε υπολογιστικούς πόρους. Το κόστος αυτό πολλές φορές κρίνει την επιλογή και το σχεδιασμό ενός αλγορίθμου. (σ.89)

4. Από ποιούς παράγοντες εξαρτάται ο χρόνος εκτέλεσης ενός αλγορίθμου; (σ.91)

- Τύπος ηλεκτρονικού υπολογιστή που θα εκτελέσει το πρόγραμμα του αλγορίθμου,
- Γλώσσα προγραμματισμού που θα χρησιμοποιηθεί,
- Δομή προγράμματος και δομές δεδομένων που χρησιμοποιούνται,
- Χρόνος για πρόσβαση στο δίσκο και στις ενέργειες εισόδου-εξόδου,
- Είδος συστήματος, ενός χρήστη ή πολλών χρηστών.

5. Ποιες προϋποθέσεις πρέπει να ικανοποιούνται για να έχει έννοια κάθε σύγκριση μεταξύ δύο προγραμμάτων αλγορίθμων; (σ.92)

- και τα δύο προγράμματα να έχουν συνταχθεί στην ίδια γλώσσα προγραμματισμού,
- να έχει χρησιμοποιηθεί ο ίδιος μεταφραστής της γλώσσας προγραμματισμού,
- να χρησιμοποιείται η ίδια υπολογιστική πλατφόρμα,
- ακριβώς τα ίδια δεδομένα να αποτελούν είσοδο κατά τον έλεγχο των δύο αλγορίθμων.

6. Πότε ένας αλγόριθμος θεωρείται αποδοτικότερος από έναν άλλο;

Αν οι δύο αλγόριθμοι έχουν το ίδιο αποτέλεσμα, αλλά ο ένας δίνει τα αποτελέσματα σε λιγότερο χρόνο, έχει χρήση λιγότερης μνήμης, τότε είναι αποδοτικότερος από τον άλλο.

7. Ποιες εντολές αποτελούν κρίσιμο σημείο για το χαρακτηρισμό της επίδοσης

Οι βρόχοι επανάληψης αποτελούν το κρίσιμο σημείο για το χαρακτηρισμό της επίδοσης ενός αλγορίθμου. Έτσι αν σε έναν αλγόριθμο ο βρόχος να εκτελεσθεί  $n$  φορές, ο χρόνος εκτέλεσης θα εξαρτάται από το μέγεθος του  $n$ .

8. Πώς υπολογίζεται ο εμπειρικός (εκ των υστέρων) τρόπος μέτρησης της επίδοσης ενός αλγορίθμου;

Ο απλούστερος τρόπος μέτρησης της επίδοσης ενός αλγορίθμου είναι ο εμπειρικός (empirical) ή αλλιώς ο λεγόμενος εκ των υστέρων (a posteriori). Δηλαδή, ο αλγόριθμος υλοποιείται και εφαρμόζεται σε ένα σύνολο δεδομένων, ώστε να υπολογισθεί ο απαιτούμενος χρόνος επεξεργασίας (processing time) και η χωρητικότητα μνήμης (memory space).

9. Ποια τα μειονεκτήματα του εμπειρικού τρόπου μέτρησης της επίδοσης ενός αλγορίθμου;

- με τη μέθοδο αυτή είναι δύσκολο να **προβλεφθεί η συμπεριφορά** του αλγορίθμου για κάποιο **άλλο σύνολο δεδομένων**,
- ο **χρόνος επεξεργασίας** εξαρτάται από το υλικό, τη γλώσσα προγραμματισμού και το μεταφραστή και προ πάντων από τη δεινότητα του προγραμματιστή.

10. Πώς υπολογίζεται ο **θεωρητικός (εκ των προτέρων)** τρόπος μέτρησης της επίδοσης ενός αλγορίθμου;

Ο **θεωρητικός (εκ των προτέρων)** τρόπος εκτίμησης της επίδοσης ενός αλγορίθμου υλοποιείται έχοντας **μία μεταβλητή  $n$** , που εκφράζει το **μέγεθος (size) του προβλήματος**, ώστε η μέτρηση της αποδοτικότητας του αλγορίθμου να ισχύει για οποιοδήποτε σύνολο δεδομένων και ανεξάρτητα από υποκειμενικούς παράγοντες, όπως αυτοί που αναφέρθηκαν.

11. Τι ονομάζεται τάξη της πολυπλοκότητας ενός αλγορίθμου;

Αν η πολυπλοκότητα ενός αλγορίθμου είναι  $f(n)$ , τότε λέγεται ότι είναι τάξης  $O(g(n))$ , αν υπάρχουν δύο θετικοί ακέραιοι  $c$  και  $n_0$ , έτσι ώστε για κάθε  $n \geq n_0$  να ισχύει:  $|f(n)| \leq c |g(n)|$ .

**Ο συμβολισμός  $O$  δίνει ένα άνω φράγμα για την πολυπλοκότητα ενός αλγορίθμου**, δηλαδή δίνει ένα μέτρο, που ποτέ δεν πρόκειται να ξεπεράσει ο αλγόριθμος προς τα επάνω.

12. Τι είναι πολυπλοκότητα;

Ο Ορισμός της πολυπλοκότητας διδάσκεται από το βιβλίο της Β' ΓΕΛ. "Η πολυπλοκότητα ενός αλγορίθμου δίνει ένα μέτρο της χρονικής καθυστέρησης του αλγορίθμου για την επίλυση ενός προβλήματος" ή ισοδύναμα "Η πολυπλοκότητα ενός αλγορίθμου δίνει ένα μέτρο της ταχύτητας εκτέλεσης του αλγορίθμου".

13. Δώστε παράδειγμα αλγορίθμου χρονική πολυπλοκότητα στην κατηγορία:  $O(1)$ . Πόσες φορές εκτελούνται οι εντολές;

Κάθε εντολή του προγράμματος εκτελείται μία φορά ή το πολύ μερικές μόνο φορές. Στην περίπτωση αυτή λέγεται ότι ο αλγόριθμος είναι **σταθερής** πολυπλοκότητας.

14. Δώστε παράδειγμα αλγορίθμου χρονική πολυπλοκότητα στην κατηγορία:  **$O(\log n)$** . Πόσες φορές εκτελούνται οι εντολές;

Ο αλγόριθμος είναι **λογαριθμικής** πολυπλοκότητας. Παράδειγμα η δυαδική αναζήτηση σε ταξινομημένο πίνακα. Αν το πλήθος των δεδομένων είναι  $n$  τότε ο μέγιστος αριθμός βημάτων έχει  $\log n$ . Συνήθως, οι λογάριθμοι που χρησιμοποιούνται στο βιβλίο αυτό είναι δυαδικοί.

15. Δώστε παράδειγμα αλγορίθμου χρονική πολυπλοκότητα στην κατηγορία:  **$O(n)$** . Πόσες φορές εκτελούνται οι εντολές;

Η πολυπλοκότητα λέγεται **γραμμική**. Παράδειγμα η σάρωση ενός πίνακα  $n$  στοιχείων. Θα εκτελεστούν  $n$  βήματα. Αυτή είναι η καλύτερη επίδοση για έναν αλγόριθμο που πρέπει να εξετάσει ή να δώσει στην έξοδο  $n$  στοιχεία.

16. Δώστε παράδειγμα αλγορίθμου χρονική πολυπλοκότητα στην κατηγορία:  **$O(n \log n)$** .

Στην κατηγορία αυτή ανήκει μία πολύ σπουδαία οικογένεια αλγορίθμων ταξινόμησης ή quickSort. Διαβάζεται όπως ακριβώς γράφεται ( $n \log n$ ), δηλαδή χωρίς να χρησιμοποιείται κάποιο επίθετο (όπως για παράδειγμα **γραμμολογαριθμική**).

17. Δώστε παράδειγμα αλγορίθμου χρονική πολυπλοκότητα στην κατηγορία:  **$O(n^2)$** .

**Η ταξινόμηση φυσαλίδας (ευθείας ανταλλαγής) έχει απόδοση  $O(n^2)$ . Αν έχουμε  $n$  στοιχεία τα βήματα που θα εκτελέσει θα είναι τάξης  $n^2$** . Πρέπει να χρησιμοποιείται μόνο για προβλήματα μικρού μεγέθους.

18. Δώστε παράδειγμα αλγορίθμου χρονική πολυπλοκότητα στην κατηγορία:  **$O(n^3)$** .

**Αν έχουμε τρεις εμφωλευμένους βρόχους από 1 μέχρι  $n$  τότε η απόδοση είναι  $O(n^3)$  δηλαδή Κυβική πολυπλοκότητα**. Και αυτοί οι αλγόριθμοι πρέπει να χρησιμοποιούνται μόνο για προβλήματα μικρού μεγέθους.

19. Πότε χρησιμοποιούνται αλγόριθμοι **εκθετικής πολυπλοκότητας**.

Σπάνια στην πράξη χρησιμοποιούνται αλγόριθμοι **εκθετικής πολυπλοκότητας  $O(2^n)$**  και μόνο όταν το  $n$  είναι πολύ μικρό

Πολυπλοκότητες μερικών αλγορίθμων

| Αλγόριθμος                    | Πολυπλοκότητα |
|-------------------------------|---------------|
| Ώθηση-απώθηση σε στοιβά       | $O(1)$        |
| Εισαγωγή-εξαγωγή σε ουρά      | $O(1)$        |
| Fibonacci επαναληπτική        | $O(n)$        |
| Ταξινόμηση ευθείας ανταλλαγής | $O(n^2)$      |
| Σειριακή αναζήτηση            | $O(n)$        |
| Δυαδική αναζήτηση             | $O(\log n)$   |