ΚΕΦΑΛΑΙΟ 5ο

ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ

5.1. Προσδοκώμενα αποτελέσματα

Όταν θα έχεις ολοκληρώσει τη μελέτη αυτού του κεφαλαίου θα έχεις κατανοήσει τις τεχνικές ανάλυσης των αλγορίθμων. Θα μπορείς να μετράς την επίδοση των αλγορίθμων με βάση την αποδοτικότητά τους. Θα είσαι σε θέση χρησιμοποιώντας τις κατάλληλες μεθόδους να ελέγχεις τη ορθότητά τους. Ή ακόμα λαμβάνοντας υπόψη σου κάποια κριτήρια, να επιλέγεις τον προτιμότερο αλγόριθμο για το πρόβλημα που καλείσαι να αντιμετωπίσεις. Τέλος, θα έχεις κατανοήσει την έννοια της πολυπλοκότητας των αλγορίθμων.

5.2. Επιπλέον παραδείγματα

Παράδειγμα 1

Έστω ότι έχουμε το παρακάτω πρόγραμμα υλοποίησης ενός αλγορίθμου :

```
Αλγόριθμος Ελεγχος_εκτέλεσης
a ← 1
b ← 2
Για  i  από  1 μέχρι 100
    a ← i
    b ← a * i
Τέλος_επανάληψης
Εκτύπωσε a
```

```
Εκτύπωσε b
Τέλος Ελεγχος_εκτέλεσης
```

Να υπολογισθεί η επίδοσή του με βάση τον αριθμό των πράξεων που θα εκτελεσθούν.

Εντολή αλγορίθμου	Αριθμός πράξεων
ανάθεση τιμών στα a και b	2
Βρόχος επανάληψης αρχική τιμή i έλεγχος i αύξηση i ανάθεση τιμών στο a ανάθεση τιμών στο b (2X100)	 1 101 100 100 200
Εκτύπωση a,b	2
ΣΥΝΟΛΟ	***506***

Με δεδομένο ότι ο βρόχος του προγράμματος θα εκτελεσθεί 100 φορές προκύπτει η παραπάνω ανάλυση, η οποία αποτελεί εκτίμηση και του χρόνου εκτέλεσης του προγράμματος αλγορίθμου. Είναι χρήσιμο εδώ να καταγραφεί το μέγεθος του προβλήματος και να εκφρασθεί το σύνολο των κριτηρίων επίδοσης σε σχέση με αυτό.

Παράδειγμα 2

Έστω ότι έχουμε τον παρακάτω αλγόριθμο ανάγνωσης και άμεσης εκτύπωσης των στοιχείων ενός δισδιάστατου πίνακα Α :

```
Αλγόριθμος Ανάγνωση_Εκτύπωση_Πίνακα
Δεδομένα  // n //
Για i από 1 μέχρι  n
   Για j από 1 μέχρι  n
      Διάβασε  A[i, j]
      Εκτύπωσε  A[i, j]
   Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος Ανάγνωση_Εκτύπωση_Πίνακα
```

Να υπολογισθεί ο χρόνος εκτέλεσης του αλγορίθμου αυτού και να σχολιασθεί η βαρύτητα των πράξεων επανάληψης σε σχέση με την απόφαση για την πολυπλοκότητα των αλγορίθμων.

Από ότι παρατηρούμε υπάρχουν δύο βρόχοι επανάληψης (ένας βρόχος για κάθε διάσταση του πίνακα). Για κάθε στιγμή επανάληψης μέσα στο εσωτερικό των δύο βρόχων γίνονται δύο απλές πράξεις (ανάγνωση και εκτύπωση) μοναδιαίου κόστους η καθεμία. Επομένως η πολυπλοκότητα του παραπάνω αλγορίθμου θα εκφράζεται με n * n * 2, δηλαδή ο αλγόριθμος είναι τετραγωνικός. Είναι φανερό ότι οι βρόχοι επανάληψης είναι εκείνοι που καθορίζουν την επιβάρυνση στο κόστος εκτέλεσης του αλγορίθμου.

Παράδειγμα 3. Ανάλυση αλγορίθμου ταξινόμησης.

Έστω ότι έχουμε την παρακάτω απλή μορφή για τον αλγόριθμο ταξινόμησης με ευθεία ανταλλαγή (bubblesort) :

```
Αλγόριθμος Ευθεία_Ανταλλαγή
Δεδομένα  // A //
Για i από 1 μέχρι n-1
   Για j από 1 μέχρι n-1
      Αν A[j+1] < A[j]  τότε
         Αντιμετάθεσε A[j+1], A[j]
      Τέλος_αν
   Τέλος_επανάληψης
Τέλος_επανάληψης
Αποτελέσματα // A //
Τέλος Ευθεία_Ανταλλαγή
```

Έστω ότι έχουμε τον πίνακα Α με τα παρακάτω στοιχεία :

1	2	3	4
7	4	3	8

Να παρακολουθήσετε την πορεία του αλγορίθμου με καταγραφή της επίδοσής του που θα εκφράζεται από τον αριθμό των πράξεων που πρέπει να εκτελεσθούν.

Επίδοση Αλγορίθμου

Για να εκφρασθεί η απόδοση του αλγορίθμου χρειάζεται να μετρηθεί ο αριθμός των συγκρίσεων και ο αριθμός των ανταλλαγών που θα πραγματοποιηθούν για την ταξινόμηση. Στη συνέχεια παρουσιάζονται αναλυτικά αυτοί οι υπολογισμοί :

Αριθμός Συγκρίσεων

i ← 1 (1ο Πέρασμα)

1	2	3	4	
7	4	3	8	Σύγκριση 1ου και 2ου
4	7	3	8	Σύγκριση 2ου και 3ου
4	3	7	8	Σύγκριση 3ου και 4ου

i ← 2 (2ο Πέρασμα)

1	2	3	4	
4	3	7	8	Σύγκριση 1ου και 2ου
3	4	7	8	Σύγκριση 2ου και 3ου
3	4	7	8	Σύγκριση 3ου και 4ου

i ← 3 (3ο Πέρασμα)

1	2	3	4	
3	4	7	8	Σύγκριση 1ου και 2ου
3	4	7	8	Σύγκριση 2ου και 3ου
3	4	7	8	Σύγκριση 3ου και 4ου

Όπως φαίνεται από την παρακολούθηση της πορείας του αλγορίθμου η εκτέλεση του περιλαμβάνει 3 συνεχόμενα «περάσματα» πάνω από τα στοιχεία του πίνακα και σε κάθε πέρασμα συγκρίνονται ζευγάρια στοιχείων. Από ότι φαίνεται και παραπάνω γίνονται συνολικά 9 πράξεις σύγκρισης, αφού σε κάθε πέρασμα γίνονται 3 συγκρίσεις και όπως αναφέρθηκε υπάρχουν 3 περάσματα. Ο πίνακας του παραδείγματος μας έχει n=4 θέσεις. Αν γενικεύσουμε την καταγραφή των πράξεων της σύγκρισης παρατηρούμε ότι θα χρειαστούν **(n-1)*(n-1) συγκρίσεις** για έναν πίνακα n θέσεων.

Αριθμός Ανταλλαγών

Έστω τώρα θέλουμε να καταγράψουμε τον αριθμό των ανταλλαγών που γίνονται για τις ανάγκες της ταξινόμησης. Για να υπάρξει κριτήριο για την τυποποίηση της επίδοσης του αλγορίθμου θα πρέπει να μετρηθεί ο αριθμός των ανταλλαγών που γίνονται στη χειρότερη περίπτωση. Επομένως η χειρότερη περίπτωση για ένα πίνακα 4 θέσεων (που είναι και το μέγεθος του προβλήματός μας) είναι τα στοιχεία του να βρίσκονται ταξινομημένα κατά φθίνουσα τάξη έτσι ώστε όλα να πρέπει να αλλάξουν θέση. Έστω λοιπόν η πιο απλή περίπτωση αυτής της κατάστασης :

i ← 1 (1ο Πέρασμα)

4	3	2	1	1 ανταλλαγή
3	4	2	1	1 ανταλλαγή
3	2	4	1	1 ανταλλαγή

i ← 2 (2ο Πέρασμα)

3	2	1	4	1 ανταλλαγή
2	3	1	4	1 ανταλλαγή
2	1	3	4	0 ανταλλαγή

i ← 3 (3ο Πέρασμα)

2	1	3	4	1 ανταλλαγή
1	2	3	4	0 ανταλλαγή
1	2	3	4	0 ανταλλαγή

Επομένως θα γίνουν συνολικά 3+2+1=6 ανταλλαγές για την ταξινόμηση 4 στοιχείων που είναι τοποθετημένα σε αντίθετη της κανονικής τάξης (χειρότερη περίπτωση). Αν γενικεύσουμε το παράδειγμα για πίνακα n θέσεων θα χρειασθούν στη χειρότερη περίπτωση **(n-1)+(n-2)+...+1 ανταλλαγές στοιχείων**.

Η πολυπλοκότητα του παραπάνω αλγορίθμου υπολογίζεται από το άθροισμα του «κόστους» των συγκρίσεων και των ανταλλαγών που είναι

$$(n-1)+(n-2)+\ldots+1+(n-1)\cdot(n-1)=\frac{3}{2}n^2-\frac{5}{2}n+1$$

Είναι φανερό ότι η τάξη της πολυπλοκότητας του αλγορίθμου υπολογίζεται από το μεγαλύτερο όρο του πολυωνύμου, ο οποίος είναι το n^2, και έτσι προκύπτει ότι ο παραπάνω αλγόριθμος έχει τετραγωνική πολυπλοκότητα

Παράδειγμα 4. Εύρεση του i-οστού αριθμού Fibonacci

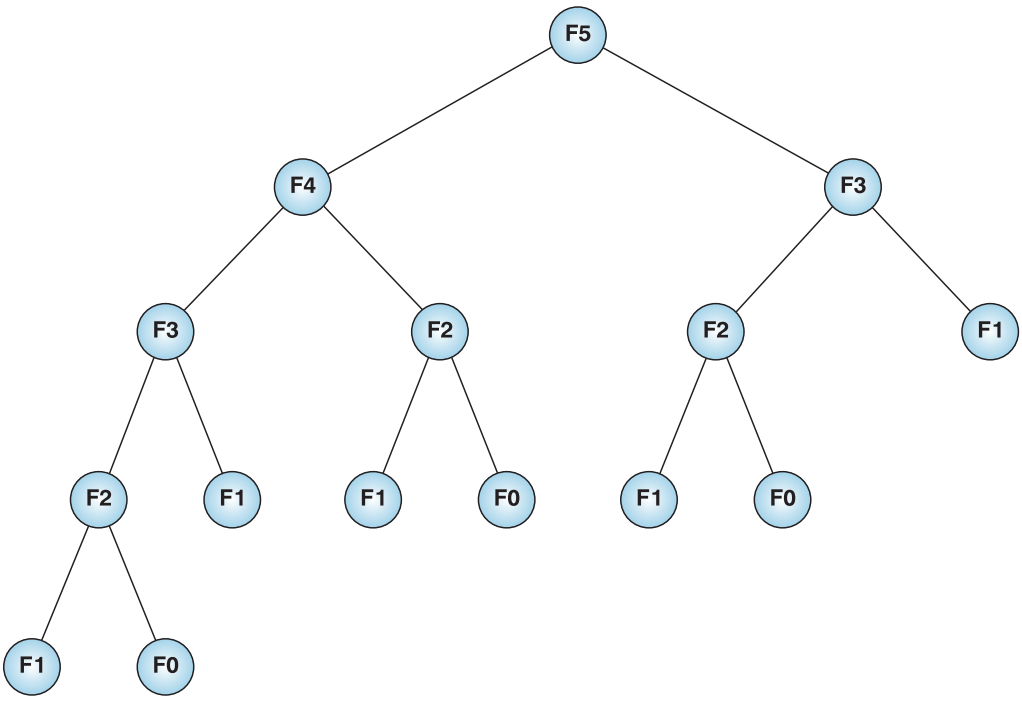
Στο κεφάλαιο του Βιβλίου σου περί αναδρομής συζητήσαμε τα πλεονεκτήματα και μειονεκτήματα των επαναληπτικών και των αναδρομικών συναρτήσεων. Μάλιστα καταλήξαμε στο συμπέρασμα ότι μία επαναληπτική διαδικασία πρέπει να προτιμάται έναντι μίας ισοδύναμης αναδρομικής όταν το βασικό μας κριτήριο είναι ο χρόνος εκτέλεσης/απόκρισης. Στο σημείο αυτό επανερχόμαστε ώστε να εξηγηθεί ότι συχνά μία αναδρομική πρέπει να αποφεύγεται και για ένα άλλο σημαντικό λόγο.

Οι επόμενοι δύο αλγόριθμοι υπολογίζουν τον i-οστό αριθμό Fibonacci. Ο πρώτος είναι επαναληπτικός ενώ ο δεύτερος είναι αναδρομικός.

```
Αλγόριθμος Fibonacci
Δεδομένα // n //
Αν n ≤ 1 τότε
    Fib ← n
αλλιώς
    j ← 0
    k ← 1
    Για i από 1 μέχρι n
        j ← j+k
        k ← j-k
    Τέλος_επανάληψης
    Fib ← j
Τέλος_Αν
Αποτελέσματα // Fib //
Τέλος Fibonacci
```

```
Αλγόριθμος Fibonacci
Δεδομένα // n //
Αν n ≤ 1 τότε
    Fib ← n
αλλιώς
    Fib ← Fib(n-1) + Fib(n-2)
Τέλος_Αν
Αποτελέσματα // Fib //
Τέλος Fibonacci
```

Η επαναληπτική διαδικασία, λοιπόν, βασίζεται σε ένα και μόνο βρόχο, άρα η πολυπλοκότητα της μεθόδου είναι τάξης O(n). Η δεύτερη διαδικασία υπολογίζει τις ίδιες τιμές πολλές φορές. Για παράδειγμα, κατά τον υπολογισμό του F_5 γίνονται οι κλήσεις Fib(4) και Fib(3). Όμως κατά τον υπολογισμό του Fib(4) καλείται για δεύτερη φορά η Fib(3). Με το σκεπτικό αυτό η Fib(2) θα κληθεί τρεις φορές, η Fib(1) θα κληθεί πέντε φορές και η Fib(0) θα κληθεί τρεις φορές. Έτσι συμπερασματικά προκύπτει ότι η πολυπλοκότητα της αναδρομικής μεθόδου είναι τάξης $O(F_n)$.



***Σχ. 5.1**. Κλήσεις για τον υπολογισμό του F_5.*

5.3. Δραστηριότητες - ασκήσεις

Στην τάξη

ΔΤ1. Να συζητηθεί η επίδοση των παρακάτω κομματιών αλγορίθμων και να καταγραφεί για κάθε περίπτωση και η αντίστοιχη πολυπλοκότητα. Για να βρείτε την πολυπλοκότητα θα πρέπει να μετρήσετε τον αριθμό των πράξεων στη χειρότερη περίπτωση :

1.
```
Για i από 1 μέχρι  n -1 με βήμα 2
    a ← 2*i
Τέλος_επανάληψης
```

2.
```
Για i από 1 μέχρι  n
    Για j από 1 μέχρι  n
        a ← 2*i +j
    Τέλος_επανάληψης
Τέλος_επανάληψης
```

3.
```
Για  i από 1 μέχρι  n  με βήμα 2
    Για j από 1 μέχρι  n
        a ← 2*i +j
    Τέλος_επανάληψης
Τέλος_επανάληψης
```

ΔΤ2. Εστω ότι ένας πίνακας κρατά τα ποσά που έχουν δώσει οι μαθητές της τάξης σας για την ενίσχυση του παιδικού χωριού SOS της περιοχής σας. Να δώσετε έναν αλγόριθμο για τον υπολογισμό του συνολικού ποσού που θα διατεθεί και να σχολιάσετε την πολυπλοκότητά του.

ΔΤ3. Να σχολιασθεί και να παρακολουθήσετε βήμα-βήμα τον ακόλουθο επαναληπτικό αλγόριθμο υπολογισμού των αριθμών Fibonacci. Ποιά είναι η πολυπλοκότητα του ακόλουθου αλγόριθμου ;

```
Αλγόριθμος Fibonacci
i ← 1
j ← 0
k ← 0
l ← 1
Επανάλαβε όσο n > 0
    Αν n MOD 2 = 1 τότε m ← j*l
    j ← i*l+j*k+m
    i ← i*k+m
    m ← Ρίζα(l)  'Σχόλιο: Ρίζα είναι η συνάρτηση τετραγωνικής ρίζας'
    h ← 2*k*l+m
```

```
    k ← Ρίζα(k)+m
    n ← n DIV 2
Τέλος_επανάληψης
Fib ← j
Αποτελέσματα Fib
Τέλος Fibonacci
```

ΔΤ4. Σε μία αποθήκη ταινιών κινηματογράφου υπάρχει αρχειοθέτηση των ταινιών με βάση τη χρονιά που παρουσιάσθηκε κάθε ταινία. Ενας σύλλογος ενδιαφέρεται να κάνει ένα αφιέρωμα στις ταινίες της δεκαετίας του 1960 που διαθέτει η αποθήκη. Να προτείνετε ένα αλγόριθμο αναζήτησης των ταινιών αυτών και να σχολιάσετε την επίδοση και την πολυπλοκότητά του.

Στο σπίτι

Στο τετράδιο σας αντιμετωπίστε τα παρακάτω προβλήματα :

ΔΣ1. Να βρείς την πολυπλοκότητα των παρακάτω κομματιών αλγορίθμων :

1.
```
Επανάλαβε όσο i < 100
    a ← 2*i
Τέλος_επανάληψης
```

2.
```
Για i από 1 μέχρι n -1
    a ← 2*i
Τέλος_επανάληψης
```

3.
```
Για i από 1 μέχρι n -1 με βήμα 3
    a ← 2*i
Τέλος_επανάληψης
```

ΔΣ2. Έστω ότι έχουμε τον παρακάτω αλγόριθμο :

```
Αλγόριθμος  Ελεγχος_επίδοσης
x ← 10
c ← 20
Για i από 100 μέχρι 10 με_βήμα -10
    x ← i
    y ← 2*x + i
Τέλος_επανάληψης
Εκτύπωσε x
Εκτύπωσε y
Τέλος Ελεγχος_ επίδοσης
```

Να υπολογισθεί η επίδοσή του με βάση τον αριθμό των πράξεων που θα εκτελεσθούν.

ΔΣ3. Να παρακολουθήσεις την πορεία του αλγορίθμου της Δυαδικής αναζήτησης που δόθηκε στο προηγούμενο κεφάλαιο (Τετράδιο Μαθητή) για έναν πίνακα 10 θέσεων. Να καταγράφεις τον αριθμό των πράξεων που γίνονται σε κάθε βήμα και να καταλήξεις σε ένα συμπέρασμα για την πολυπλοκότητά του.

ΔΣ4. Έστω ότι έχεις τον παρακάτω αλγόριθμο :

```
Αλγόριθμος Ευθεία_Ανταλλαγή2
Δεδομένα  //  A //
Για i από 1 μέχρι  n-1
   Για j από 1 μέχρι  n-1
      Αν A[j+1] < A[j]  τότε
         Αντιμετάθεσε A[j+1], A[j]
      Τέλος_αν
   Τέλος_επανάληψης
   Εκτύπωσε A[j]
Τέλος_επανάληψης
Αποτελέσματα // A //
Τέλος Ευθεία_Ανταλλαγή2
```

Να σχολιάσεις την πολυπλοκότητα του αλγορίθμου. Χρησιμοποίησε το Παράδειγμα 2 και να ελέγξεις αν ο παραπάνω αλγόριθμος είναι ίδιας ή διαφορετικής πολυπλοκότητας από τον αλγόριθμο του Παραδείγματος 2.

ΔΣ5. Μπορείς να προτείνεις κάποιο διαφορετικό αλγόριθμο για την ανεύρεση των ταινιών κινηματογράφου από την αποθήκη που περιγράφηκε παραπάνω στις δραστηριότητες για την τάξη (ΔΤ5) ; Πώς θα σχολίαζες την πολυπλοκότητα των δύο διαφορετικών αλγορίθμων ;

5.4. Τεστ αυτοαξιολόγησης

Δίνονται οι παρακάτω ομάδες προτάσεων. Σε κάθε μία από αυτές, να κάνετε τις απαραίτητες διορθώσεις ώστε να ισχύουν οι προτάσεις

1. Η χειρότερη περίπτωση ενός αλγορίθμου αφορά στο ελάχιστο κόστος εκτέλεσης του αλγορίθμου, κόστος που μετράται σε υπολογιστικούς πόρους.
2. Τα δεδομένα συνιστούν το μέγεθος της πολυπλοκότητας ενός αλγορίθμου.
3. Ο απλούστερος τρόπος μέτρησης της επίδοσης ενός αλγορίθμου είναι ο *εμπειρικός* ή αλλιώς ο λεγόμενος *εκ των προτέρων που* υλοποιείται και εφαρμόζεται σε ένα σύνολο δεδομένων ώστε να υπολογισθεί ο απαιτούμενος *χρόνος επεξεργασίας* και η πολυπλοκότητα.

Συμπλήρωσε τα κενά με το σωστή λέξη που λείπει

4 Τα δεδομένα συνιστούν το μέγεθος της _____ ενός αλγορίθμου.

5 Ο συμβολισμός $O(n^2)$ εκφράζει την _____ πολυπλοκότητα.

6 Ο συμβολισμός O(logn) εκφράζει τη _____ πολυπλοκότητα

7 Ως _____ ορίζονται οι αλγόριθμοι που δε δίνουν την καλύτερη λύση, αλλά προτιμώνται για λόγους ταχύτητας.

Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος

8 Αν η πολυπλοκότητα ενός αλγορίθμου είναι f(n), τότε λέγεται ότι ο αλγόριθμος είναι τάξης O(g(n)) αν υπάρχουν δύο θετικοί ακέραιοι c και n0, έτσι ώστε για κάθε n>=n0 να ισχύει: ¦f(n)¦ > c ¦g(n)¦

9 Ο συμβολισμός $O(n^2)$ εκφράζει την Κυβική πολυπλοκότητα και πρέπει να χρησιμοποιείται μόνο για προβλήματα μεγάλου μεγέθους.

10 Ο συμβολισμός O(n) εκφράζει τη γραμμική πολυπλοκότητα η οποία είναι η καλύτερη επίδοση για έναν αλγόριθμο που πρέπει να εξετάσει ή να δώσει στην έξοδο n στοιχεία.

11 Ευριστικοί λέγονται οι αλγόριθμοι που δεν ειναι τυποποιημένοι και δεν ανήκουν στις γνωστές οικογένειες αλγορίθμων.

Διάλεξε όλα όσα χρειάζεται μεταξύ των προτεινόμενων

12. Μία βασική πράξη μπορεί να είναι :

Α) βρόχος επανάληψης

Β) ανάθεση τιμής

Γ) σύγκριση μεταξύ δύο μεταβλητών

Δ) συνθήκη Αν..αλλιώς

Ε) οποιαδήποτε αριθμητική πράξη μεταξύ δύο μεταβλητών

13. Ο χρόνος εκτέλεσης κάθε αλγορίθμου εξαρτάται από ένα σύνολο παραγόντων που περιλαμβάνουν τα εξής:

Α) Τύπος ηλεκτρονικού υπολογιστή που θα εκτελέσει το πρόγραμμα του αλγορίθμου

Β) Χρονική στιγμή εκτέλεσης του αλγορίθμου

Γ) Συνθήκες ανταγωνισμού με άλλους αλγορίθμους

Δ) Γλώσσα προγραμματισμού που θα χρησιμοποιηθεί

Ε) Δομή προγράμματος και δομές δεδομένων που χρησιμοποιεί

Ζ) Αριθμός εντολών του αλγορίθμου

Η) Χρόνος για πρόσβαση στο δίσκο και στις ενέργειες εισόδου-εξόδου

Θ) Είδος συστήματος, ενός χρήστη ή πολλαπλών χρηστών.