

Προγραμματιστική Εργασία

Ημερομηνία Παράδοσης: 6 Ιουλίου 2007

1 Τί πρέπει να κάνετε για την εργασία αυτή

Στην εργασία αυτή θα προσομοιώσουμε τον αλγόριθμο που χρησιμοποιείται στις Πανελλαδικές Εξετάσεις για τον καθορισμό των επιτυχόντων και αποτυχόντων. Πιο συγκεκριμένα πρόκειται για ένα απλοποιημένο μοντέλο των εξετάσεων αυτών στο οποίο κάθε φοιτητής γράφει σε έξι μαθήματα, και έχει δικαίωμα να δηλώσει c προτιμήσεις μεταξύ των d διαθέσιμων τμημάτων Πανεπιστημίων ($c \leq d$). Σε ότι αφορά τον υπολογισμό του βαθμού κάθε φοιτητή θα θεωρήσουμε δύο συστήματα υπολογισμού της βαθμολογίας, ένα απλό και ένα σύνθετο. Σκοπός σας είναι να γράψετε κώδικα που να υπολογίζει αν κάποιος μαθητής πέτυχε στις εξετάσεις και σε ποιο τμήμα για κάθε ένα από τα δύο συστήματα υπολογισμού της βαθμολογίας του. Ο αλγόριθμός σας θα πρέπει να είναι όσο το δυνατόν πιο αποδοτικός (μετράμε την απόδοσή του με βάση την ασυμπτωτική του πολυπλοκότητα) ως προς τον αριθμό των μαθητών n , τον αριθμό των τμημάτων d και τον αριθμό των επιλογών κάθε μαθητή c .

2 Δομή Student

Ο μαθητής είναι μία δομή (struct) της C ως κάτωθι:

```
struct Student
{
    unsigned int id;
    unsigned int nc;
    unsigned int AL;
    unsigned int EG[N_EXAMS];
    unsigned int choices[N_CHOICES_MAX];
};
```

Τα πεδία της δομής έχουν ως εξής:

id Είναι ο κωδικός αριθμός του μαθητή. Πρόκειται για μη-μηδενικό ακέραιο, που είναι μοναδικός για κάθε μαθητή.

nc Ο αριθμός των τμημάτων c που έχει επιλέξει ο μαθητής. Στο μοντέλο μας, ο αριθμός αυτός είναι ίδιος για όλους τους μαθητές. Η τιμή του nc είναι το πολύ $N_CHOICES_MAX$ η οποία είναι σταθερά με τιμή 50.

AL Ο βαθμός απολυτηρίου Λυκείου του μαθητή. Πρόκειται για ακέραιο μεταξύ 1000 και 2000.

EG[N_EXAMS] Πίνακας με τις βαθμολογίες του μαθητή στα 6 μαθήματα που έδωσε εξετάσεις. Η N_EXAMS είναι σταθερά με τιμή 6. Οι 6 τιμές του πίνακα είναι ακέραιοι μεταξύ 0 και 2000.

choices[N_CHOICES_MAX] Πίνακας με τους κωδικούς των τμημάτων στα οποία επιθυμεί να πετύχει ο μαθητής. Μόνο οι nc πρώτες τιμές του πίνακα χρησιμοποιούνται. Η πρώτη τιμή του πίνακα είναι η πρώτη επιλογή του μαθητή, η δεύτερη τιμή του πίνακα αντιστοιχεί στην δεύτερη κατά σειρά επιλογή του μαθητή, κοκ.

3 Δομή Department

το τμήμα είναι μία δομή (struct) της C ως κάτωθι:

```
struct Department
{
    unsigned int id;
    unsigned int num_accepted;
    unsigned int topic1, topic2;
    static const unsigned int cf1 = 13;
    static const unsigned int cf2 = 7;
};
```

Τα πεδία της δομής έχουν ως εξής:

id Είναι ο κωδικός αριθμός του τμήματος. Πρόκειται για μη-μηδενικό ακέραιο, που είναι μοναδικός για κάθε τμήμα.

num_accepted Ο αριθμός των μαθητών που δέχεται το τμήμα. μαθητής. Η τιμή του num_accepted είναι το μη-μηδενικός ακέραιος.

topic1 Η τιμή της μεταβλητής αυτής είναι μεταξύ 1 και 6 (συμπεριλαμβανομένων). Αντιστοιχεί στον αύξοντα αριθμό του μαθήματος που έχει τη μεγαλύτερη βαρύτητα για το συγκεκριμένο τμήμα.

topic2 Η τιμή της μεταβλητής αυτής είναι μεταξύ 1 και 6 (συμπεριλαμβανομένων). Αντιστοιχεί στον αύξοντα αριθμό του μαθήματος που έχει τη δεύτερη μεγαλύτερη βαρύτητα για το συγκεκριμένο τμήμα.

cf1 Η βαρύτητα του μαθήματος topic1.

cf2 Η βαρύτητα του μαθήματος topic2.

4 Συστήματα υπολογισμού βαθμολογίας

Όπως ήδη αναφέρθηκε στην Παράγραφο 1, στην εργασία αυτή θεωρούμε δύο συστήματα υπολογισμού της βαθμολογίας. Στο πρώτο σύστημα, το οποίο είναι και το πιο απλό εκ των δύο, ο τελικός βαθμός του φοιτητή υπολογίζεται κατά 40% από το βαθμό του απολυτηρίου Λυκείου, ενώ το υπόλοιπο 60% προκύπτει από τα 6 μαθήματα στα οποία έδωσε εξετάσεις. Η συνάρτηση που κάνει τον υπολογισμό αυτό δίνεται παρακάτω:

```
unsigned int simple_grade(const Student* s)
{
    unsigned int sum = 0;
    for (int i = 0; i < N_EXAMS; ++i) {
        sum += s->EG[i];
    }
    sum += 4 * s->AL;
    return sum;
}
```

Παρατηρήστε ότι ο βαθμός του μαθητή είναι μεταξύ 4000 και 20000, και ότι οι μεταβλητές `topic1`, `topic2`, αλλά και οι σταθερές `cf1` και `cf2` δεν χρησιμοποιούνται. Επίσης παρατηρήστε (και αυτή είναι η σημαντική διαφορά σε σχέση με το σύστημα υπολογισμού βαθμολογίας που θα παρουσιαστεί παρακάτω) ότι ο βαθμός του μαθητή είναι ανεξάρτητος του τμήματος στο οποίο θέλει να εισαχθεί.

Το δεύτερο σύστημα υπολογισμού βαθμολογίας είναι πιο κοντά στην πραγματικότητα. Ο βαθμός του μαθητή εξαρτάται από το τμήμα στο οποίο θέλει να εισαχθεί και υπολογίζεται κατά 80% από το απολυτήριο Λυκείου και κατά 20% από τα δύο μαθήματα που αντιστοιχούν στο τμήμα για το οποίο υπολογίζεται η βαθμολογία (13% από το πιο σημαντικό μάθημα για το τμήμα και 7% από το δεύτερο πιο σημαντικό μάθημα για το τμήμα). Η συνάρτηση που κάνει τον υπολογισμό αυτό δίνεται παρακάτω:

```
unsigned int grade_by_department(const Student* s, const Department* d)
{
    unsigned int sum = 80 * s->AL;
    sum += d->cf1 * s->EG[d->topic1-1] + d->cf2 * s->EG[d->topic2-1];
    return (unsigned int)(sum / 10);
}
```

4.1 Ισοβαθμίες

Ας υποθέσουμε ότι δύο μαθητές με κωδικούς 101 και 225 συγκεντρώνουν τη ίδια βαθμολογία ως προς κάποιο τμήμα (που έχουν μεταξύ των επιλογών τους). Ας υποθέσουμε ότι η βαθμολογία αυτή είναι 16534 και ότι το τμήμα δέχεται 50 μαθητές. Ας υποθέσουμε ότι η κοινή τους βαθμολογία είναι η 50η καλύτερη σε σχέση με τους μαθητές που έχουν επιλέξει το συγκεκριμένο τμήμα. Με άλλα λόγια για το συγκεκριμένο τμήμα έχουμε 50 μαθητές με βαθμολογία τουλάχιστον 16534 έναν μαθητή με την ίδια βαθμολογία με τον 50ο. Στην περίπτωση αυτή πρέπει ο 51ος μαθητής να γίνει δεκτός και αυτός στο εν λόγω τμήμα λόγω «ισοβαθμίας με τον μαθητή που καθορίζει τη βάση».

Γενικά μιλώντας, κάθε μαθητής που έχει την ίδια βαθμολογία με τον τελευταίο μαθητή που γίνεται δεκτός σε κάποιο τμήμα γίνεται και αυτός δεκτός στο εν λόγω τμήμα καθ' υπέρβαση του προκαθορισμένου αριθμού εισακτέων για το τμήμα.

5 Προδιαγραφές

Το πρόγραμμά σας θα πρέπει να είναι γραμμένο σε C ή C++. Θα πρέπει να συμμορφώνεται με τις απαιτήσεις που περιγράφονται παρακάτω στην Παράγραφο 6. Μπορείτε να θεωρήσετε ότι τα αρχεία δεδομένων δεν περιέχουν λάθη στην καταγραφή των δεδομένων οπότε δε χρειάζεται τα προγράμματά σας να ελέγχουν για λάθη στα δεδομένα. Τα προγράμματά σας θα πρέπει να μεταγλωττίζονται στους υπολογιστές των αιθουσών H205 ή Λ205 με τους μεταγλωττιστές `gcc/g++` της C/C++ που υπάρχουν εγκατεστημένοι στους υπολογιστές αυτούς.

Η εργασία αυτή γίνεται ατομικά. Στην κείμενο που θα παραδώσετε, καθώς επίσης και σε κάθε αρχείο με κώδικα που θα παραδώσετε (βλέπε Παράγραφο 8) σημειώστε το ονοματεπώνυμό σας. Αναφέρετε με σαφήνεια όλες τις πηγές σας (βιβλιογραφικές, διαδικτυακές, κτλ.), και ειδικότερα αν πρόκειται για κώδικα.

Μεταξύ των αρχείων κώδικα που θα παραδώσετε πρέπει να υπάρχει ένα αρχείο `simple_grades.c` ή `simple_grades.cpp` (αναλόγως του αν ο κώδικας είναι γραμμένος σε C ή C++) που θα λύνει το πρόβλημα κατανομής μαθητών σε τμήματα με βάση το απλό σύστημα βαθμολόγησης. Επίσης θα πρέπει να υπάρχει ένα αρχείο `grades.c` ή `grades.cpp` (αναλόγως του αν ο κώδικας είναι γραμμένος σε

C ή C++) που θα λύνει το πρόβλημα κατανομής μαθητών σε τμήματα με βάση το σύνθετο σύστημα βαθμολόγησης.

6 Επικοινωνία με το χρήστη (user interface)

Τα προγράμματά σας θα πρέπει να παίρνουν τρεις παραμέτρους (command line arguments). Οι δύο πρώτοι παράμετροι θα πρέπει να είναι αρχεία εισόδου, το μεν πρώτο για τα τμήματα το δε δεύτερο για τους μαθητές. Τα προγράμματά σας θα πρέπει να εκτυπώνουν στην οθόνη (standard output) τα αποτελέσματά σας. Η μορφή των αρχείων δεδομένων, αλλά η μορφή που πρέπει να έχει η έξοδος, εξηγούνται παρακάτω. Η τρίτη παράμετρος είναι προαιρετική και πρόκειται για ένα boolean bit. Αν η τιμή της τρίτης παραμέτρου είναι 1, τότε εκτυπώνονται στην οθόνη και οι αποτυχόντες μαθητές (βλέπε παρακάτω παράγραφο που εξηγεί το output), ενώ αν είναι η τιμή της παραμέτρου είναι 0 τότε δεν εκτυπώνονται οι αποτυχόντες μαθητές. Αν η παράμετρος αυτή δεν δίνεται, το πρόγραμμα πρέπει να συμπεφέρεται σαν να είχαμε δώσει την τιμή 0.

6.1 Αρχεία εισόδου (input)

Τα αρχεία δεδομένων είναι δύο ειδών: αρχείο εισόδου τμημάτων και αρχείο εισόδου μαθητών. Θεωρείστε ότι δεν υπάρχουν λάθη στα δύο αρχεία αυτά και ότι οι πληροφορίες των δύο αρχείων είναι συμβατές μεταξύ τους.

Το αρχείο εισόδου τμημάτων έχει την εξής μορφή: στην πρώτη γραμμή καταγράφεται ο αριθμός των τμημάτων. Σε κάθε επόμενη γραμμή έχουμε κατά σειρά τον κωδικό αριθμό του τμήματος, των αριθμό των εισακτέων και στη συνέχεια δύο ακεραίους μεταξύ 1 και 6 (συμπεριλαμβανομένων) που καθορίζουν ποια είναι δύο σημαντικά μαθήματα για το συγκεκριμένο τμήμα σε σχέση με το δεύτερο σύστημα υπολογισμού βαθμολογίας. Παρακάτω μπορείτε να δείτε πώς φαίνεται ένα τέτοιο αρχείο τμημάτων για 5 τμήματα:

```
5
1 2 5 1
2 5 4 5
3 4 3 6
4 4 3 3
5 4 2 3
```

Το αρχείο δεδομένων μαθητών έχει λίγο πιο πολύπλοκη μορφή. Στην πρώτη γραμμή έχουμε τον αριθμό των μαθητών και τον αριθμό των επιλογών κάθε μαθητή (ο αριθμός αυτός των επιλογών αντιστοιχεί στη μεταβλητή `nc` κάθε μαθητή). Σε κάθε επόμενη γραμμή έχουμε κατά σειρά τον κωδικό του μαθητή, το βαθμό απολυτηρίου Λυκείου, τους βαθμούς των 6 μαθημάτων των εξετάσεων και τους κωδικούς αριθμούς των τμημάτων επιλογής του. Παρακάτω μπορείτε να δείτε πώς φαίνεται ένα τέτοιο αρχείο μαθητών για 20 μαθητές οι οποίοι δίνουν εξετάσεις για να εισαχθούν στα 5 τμήματα του παραδείγματος που δώσαμε παραπάνω για το αρχείο τμημάτων και οι οποίοι έχουν το δικαίωμα τριών επιλογών.

```
20 3
1 1170 1500 192 1741 1155 1572 1385 2 5 4
2 1446 707 1465 520 788 1554 1690 3 4 1
3 1456 220 1091 781 1137 1919 1736 1 2 5
4 1436 1579 255 164 1882 51 308 2 1 3
```

```

5 1877 861 528 627 1540 214 1542 4 2 3
6 1649 1860 1005 1375 872 1217 1153 4 3 1
7 1961 288 893 649 1906 716 796 1 5 3
8 1866 299 1819 1303 127 1910 1933 4 5 3
9 1282 1926 1159 874 751 1847 57 4 3 1
10 1053 1549 1407 242 607 149 1376 2 4 1
11 1154 323 1665 1102 634 1056 866 2 4 1
12 1586 955 1880 1525 255 29 1000 5 1 4
13 1579 199 1657 1495 1817 1024 1770 5 3 4
14 1881 1172 445 1999 318 1391 1403 5 4 2
15 1800 318 1727 1206 840 1963 1738 3 2 5
16 1844 1473 264 954 1569 995 1138 3 1 2
17 1427 1165 176 1350 1175 471 1490 5 1 3
18 1383 3 1433 107 42 1517 1258 5 3 2
19 1197 616 418 1222 744 321 920 5 1 4
20 1477 283 1267 485 248 1743 206 4 2 3

```

6.2 Τι πρέπει να επιστρέφουν τα προγράμματά σας (output)

Τα προγράμματά σας πρέπει να συμπεριφέρονται όπως τα προγράμματα `simple_grades` και `grades` που σας δίνονται (βλ. Παράγραφο 7). Πιο συγκεκριμένα πρέπει να καταγράφονται για κάθε τμήμα: ο κωδικός του τμήματος, ο προκαθορισμένος αριθμός εισακτέων σε παρένθεση, ο πραγματικός αριθμός εισακτέων[†]. Σε κάθε επόμενη γραμμή καταγράφονται οι μαθητές που πέτυχαν στο συγκεκριμένο τμήμα σε φθίνουσα σειρά βαθμολογίας. Σε περίπτωση που δύο μαθητές έχουν την ίδια βαθμολογία καταγράφεται πρώτα εκείνος με το μικρότερο κωδικό αριθμό. Για κάθε επιτυχόντα μαθητή καταγράφεται ο αύξων αριθμός επιτυχίας του (1 για τον πρώτο, 2 για τον δεύτερο, κοκ.), ο κωδικός του, ο αύξων αριθμός επιλογής του για το συγκεκριμένο τμήμα (αν το τμήμα στο οποίο πέτυχε είναι η 3η επιλογή του πρέπει να γραφεί ο αριθμός 3) και η βαθμολογία του (αν πρόκειται για το δεύτερο σύστημα υπολογισμού βαθμολογίας, η βαθμολογία είναι η βαθμολογία του μαθητή ως προς το εν λόγω τμήμα στο οποίο πέτυχε).

Αν σας ζητείται να καταγραφούν και οι αποτυχόντες μαθητές, τότε αυτοί καταγράφονται σε αύξουσα σειρά ως προς τον κωδικό τους αριθμό. Για κάθε μαθητή καταγράφεται ο αύξων αριθμός αποτυχίας[‡], ο κωδικός του αριθμός, οι βαθμολογίες του (απολυτήριο Λυκείου και τα 6 μαθήματα) και οι επιλογές του.

Τέλος, ζητείται να εκτυπωθούν στην οθόνη κάποια στατιστικά στοιχεία, και πιο συγκεκριμένα, ο συνολικός αριθμός των μαθητών, ο αριθμός των τμημάτων, ο συνολικός προκαθορισμένος αριθμός θέσεων στα τμήματα, ο αριθμός των επιτυχόντων μαθητών και ο αριθμός των αποτυχόντων μαθητών. Για το ακριβές `formatting` δείτε τα παραδείγματα εξόδου παρακάτω ή τα προγράμματα που σας δίδονται.

Προκειμένου να γίνουν όλα αυτά πιο σαφή παρατίθενται παρακάτω τα αποτελέσματα εξόδου για τα αρχεία δεδομένων εισόδου της προηγούμενης υποπαραγράφου.

[†]Ο πραγματικός αριθμός εισακτέων μπορεί να είναι διαφορετικός από τον προκαθορισμένο είτε λόγω ισοβαθμιών, είτε επειδή οι αριθμός των μαθητών που επέλεξαν το εν λόγω τμήμα ήταν μικρότερος από τον προκαθορισμένο αριθμό εισακτέων.

[‡]Ο αύξων αριθμός αποτυχίας δεν είναι τίποτε άλλο παρά ένας μετρητής για τους αποτυχόντες μαθητές.

6.2.1 Έξοδος του προγράμματος simple_grades

Department: 1 (2) 2

1	7	1	13092
2	3	1	12708

Department: 2 (5) 5

1	1	1	12225
2	11	1	10262
3	20	2	10140
4	4	1	9983
5	10	1	9542

Department: 3 (4) 4

1	15	1	14992
2	16	1	13769
3	2	1	12508
4	18	2	9892

Department: 4 (4) 4

1	8	1	14855
2	6	1	14078
3	5	1	12820
4	9	1	11742

Department: 5 (4) 4

1	13	1	14278
2	14	1	14252
3	12	1	11988
4	17	1	11535

Failed:

1	19	1197	616	418	1222	744	321	920	5	1	4
---	----	------	-----	-----	------	-----	-----	-----	---	---	---

of students: 20

of departments: 5

of positions: 19

accepted: 19

rejected: 1

6.2.2 Έξοδος του προγράμματος grades

Department: 1 (2) 2

1	7	1	16820
2	3	1	14296

Department: 2 (5) 4

1	4	1	13970
2	1	1	11961
3	11	1	10795
4	10	1	9317

Department: 3 (4) 4

1	15	1	17184
2	16	1	16788
3	17	3	14214
4	2	1	13427

Department: 4 (4) 4

1	8	1	17534
2	5	1	16270
3	6	1	15942
4	20	1	12786

Department: 5 (4) 4

1	14	1	17025
2	12	1	16199
3	13	1	15832
4	18	1	13001

Failed:

1	9	1282	1926	1159	874	751	1847	57	4	3	1
2	19	1197	616	418	1222	744	321	920	5	1	4

# of students:	20
# of departments:	5
# of positions:	19
# accepted:	18
# rejected:	2

7 Τί σας δίνεται

Στο φάκελλο (directory) `~mkaravel/Public/A33-spring07/project/bin` μπορείτε να βρείτε δύο εκτελέσιμα αρχεία με ονόματα `simple_grades` και `grades` τα οποία αντιστοιχού στη δική μου υλοποίηση της εργασίας σας. Στο φάκελλο `~mkaravel/Public/A33-spring07/project/data` θα βρείτε δείγματα αρχείων δεδομένων, μεταξύ των οποίων τα `depts_example.cin` και `studs_example.cin` που αντιστοιχούν στα δεδομένα του παραδείγματος της Παραγράφου 6.2. Τα προγράμματα που σας δίνονται μπορείτε να το τρέξετε είτε με τα δεδομένα που υπάρχουν στον φάκελλο `data`, είτε με δικά σας δεδομένα και σκοπό έχει να σας βοηθήσει να βρείτε πιθανά λάθη στον κώδικά σας. Προκειμένου να δημιουργήσετε δικά σας αρχεία δεδομένων στο φάκελλο `~mkaravel/Public/A33-spring07/project/generators` θα βρείτε δύο προγράμματα `generate_departments.cpp` και `generate_students.cpp` γραμμένα σε C++ τα οποία μπορείτε να μεταγλωττίσετε και να τα τρέξετε για να φτιάξετε τα δικά σας αρχεία δεδομένων. Είσαστε ελεύθεροι να αλλάξετε τα προγράμματα αυτά κατά βούληση και να γράψετε τους δικούς σας `data generators`.

Τέλος, στο φάκελλο `~mkaravel/Public/A33-spring07/project/include` σας δίνονται τα παρακάτω αρχεία:

`student.h` Στο αρχείο αυτό θα βρείτε τη δομή `Student` καθώς συνατήσεις που διαβάζουν τα δεδομένα της δομής (πλην της μεταβλητής `nc`) από αρχείο, ή τα γράφουν σε αρχείο.

`student.hpp` Ότι και το αρχείο `student.h`, αλλά σε C++.

`department.h` Στο αρχείο αυτό θα βρείτε τη δομή `Department` καθώς συνατήση που διαβάζει τα δεδομένα της δομής από αρχείο.

`department.hpp` Ότι και το αρχείο `student.h`, αλλά σε C++. Στο αρχείο αυτό θα βρείτε τελεστές (operators) ισότητας και ανισότητας για τη δομή `Department`.

`global_consts.h` Το αρχείο αυτό περιέχει τους ορισμούς για τις σταθερές `N_EXAMS` και `N_CHOICES_MAX`.

`grading_functions.h` Το αρχείο αυτό περιέχει τους ορισμούς των δύο συναρτήσεων υπολογισμού βαθμολογίας όπως αυτές παρουσιάστηκαν στην Παράγραφο 4.

`grading_functions.hpp` Ότι και το αρχείο `grading_functions.h`, αλλά σε C++.

8 Τί θα παραδώσετε

Θα πρέπει να παραδώσετε τον κώδικά σας, σε ένα αρχείο της μορφής `.zip` ή `.tar` ή `.tar.gz` ή `.tgz`. Επίσης θα πρέπει να παραδώσετε ένα δακτυλογραφημένο κείμενο 5-10 σελίδων (κατά προτίμηση σε `LATEX`) που

- θα περιγράφει και θα εξηγεί τους αλγορίθμους σας,
- θα περιέχει την ανάλυση της ασυμπτωτικής πολυπλοκότητας των αλγορίθμων σας, και
- θα περιγράφει την υλοποίηση των αλγορίθμων σας.

9 Βαθμολόγηση της εργασίας

Οι εργασίες σας θα βαθμολογηθούν ως εξής:

- από το κείμενο που θα παραδώσετε,
- από το σχεδιασμό του κώδικά σας,
- από το κατά πόσο ο αλγόριθμός σας είναι αποδοτικός ασυμπτωτικά,
- από το κατά πόσο η υλοποίησή σας πετυχαίνει τον ασυμπτωτικό χρόνο που προέκυψε από την ανάλυσή σας,
- από την ορθότητα του αλγορίθμου σας (από το αν, δηλαδή, θα δίνει τα σωστά αποτελέσματα στα δεδομένα στα οποία θα ελεγχθεί).

Υλοποιήσεις που δεν τηρούν τις προδιαγραφές που αναφέρονται στην Παράγραφο 5 δε θα βαθμολογηθούν και θα θεωρηθεί ότι η προγραμματιστική εργασία δεν έχει παραδοθεί.

10 Σχετική βιβλιογραφία

Εκτός των συγγραμμάτων του μαθήματος μπορείτε επίσης να ανατρέξετε στα παρακάτω συγγράμματα:

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliff Stein, Introduction to Algorithms, 2nd Edition, MIT Press, 2001.
- Robert Sedgwick, Algorithms in C++, 3rd Edition, Addison-Wesley, 1998.