Εργασία στο μάθημα Δομές Δεδομένων και Αλγόριθμοι

Δρ. Γκόγκος Χρήστος
Τμήμα Μηχανικών Πληροφορικής ΤΕ ΤΕΙ Ηπείρου

Οκτώβριος 2014

1 Εισαγωγή

Η αποδοτική δημιουργία προγραμμάτων εξετάσεων είναι ένα σημαντικό και επαναλαμβανόμενο πρόβλημα το οποίο καλούνται να αντιμετωπίσουν τα εκπαιδευτικά ιδρύματα σε όλο τον κόσμο. Μια απλοποιημένη μορφή του προβλήματος έχει προταθεί από τους Carter κ.ά. [CLL96] οι οποίοι διέθεσαν δημόσια 13 στιγμιότυπα προβλημάτων που εν συνεχεία χρησιμοποιήθηκαν σε πληθώρα επιστημονικών εργασιών χρονοπρογραμματισμού. Στα πλαίσια της παρούσας εργασίας ζητείται να κατασκευάσετε μια εφαρμογή που θα είναι σε θέση να διαβάσει 5 στιγμιότυπα προβλημάτων και να αποθηκεύσει τα δεδομένα σε κατάλληλες δομές που θα επιτρέπουν την εκτέλεση συγκεκριμένων ενεργειών σε αυτές.

2 Περιγραφή προβλήματος

Το πρόβλημα αφορά εξετάσεις, σπουδαστές και περιόδους εξέτασης. Κάθε εξέταση έχει μια λίστα από εγγεγραμμένους σπουδαστές και κάθε σπουδαστής μπορεί να είναι εγγεγραμμένος σε μια ή περισσότερες εξετάσεις. Κάθε εξέταση μπορεί να τοποθετηθεί σε μια περίοδο εξέτασης από έναν αριθμό διαθέσιμων περιόδων εξέτασης (π.χ. 20, 35 κλπ). Το πρόβλημα επιζητεί την ανάθεση εξετάσεων σε περιόδους έτσι ώστε να μην υπάρχουν **συγκρούσεις**, δηλαδή να μην υπάρχει σπουδαστής που θα έπρεπε να συμμετέχει σε εξετάσεις σε περισσότερα του ενός μαθήματα στην ίδια περίοδο. Καθώς είναι ενδεχόμενο να υπάρχουν πολλά εναλλακτικά προγράμματα που να ικανοποιούν τον ανωτέρω περιορισμό, προτιμότερο θεωρείται μεταξύ τους εκείνο το πρόγραμμα που δεν έχει μικρό αριθμό κενών περιόδων μεταξύ διαδοχικών εξετάσεων για κάθε σπουδαστή. Ειδικότερα, ορίζονται τιμές ποινής που είναι 16, 8, 4, 2 ή 1 στις περιπτώσεις που ένας σπουδαστής πρέπει να γράψει δύο εξετάσεις που απέχουν 1, 2, 3, 4 ή 5 περιόδους αντίστοιχα. Η συνολική ποινή από όλους τους σπουδαστές αποτελεί το κόστος της λύσης.

3 Δεδομένα προβλήματος

Τα δεδομένα του προβλήματος μπορείτε να τα κατεβάσετε από τη σελίδα `ftp://ftp.mie.utoronto.ca/pub/carter/testprob/all_file.zip`. Η εργασία αφορά τα αρχεία μαθημάτων και σπουδαστών που παρουσιάζονται στον Πίνακα 1.

Πίνακας 1: Δεδομένα εργασίας

Πρόβλημα	Περιγραφή	Αρχείο Μαθημάτων	Αρχείο Σπουδαστών	Εξετάσεις	Σπουδαστές	Εγγραφές	Πυκνότητα	Περίοδοι
CAR91	Carleton University	car-s-91.crs	car-s-91.stu	682	16925	56242	0,13	35
KFU93	King Fahd University	kfu-s-93.crs	kfu-s-93.stu	461	5349	25113	0,06	20
LSE91	London School of Economics	lse-f-91.crs	lse-f-91.stu	381	2726	10918	0,06	18
TRE92	Trent University	tre-s-92.crs	tre-s-92.stu	261	4360	14901	0,18	23
UTE92	University of Toronto, Engineering	ute-s-92.crs	ute-s-92.stu	184	2750	11793	0,08	10

Τα αρχεία που αφορούν τα μαθήματα (κατάληξη .crs) έχουν δύο στήλες με την πρώτη να είναι ο αριθμός του μαθήματος και την δεύτερη να περιέχει το πλήθος των σπουδαστών που είναι

εγγεγραμμένοι στο αντίστοιχο μάθημα.
Τα αρχεία που αφορούν τους σπουδαστές (κατάληξη .stu) έχουν για κάθε σπουδαστή μια γραμμή που περιέχει τους αριθμούς των μαθημάτων στα οποία είναι εγγεγραμμένος χωρισμένους μεταξύ τους με κενά. Η πρώτη γραμμή αντιστοιχεί στον πρώτο σπουδαστή, η δεύτερη γραμμή στον δεύτερο σπουδαστή κ.ο.κ.

4 Ζητούμενα

Να κατασκευάσετε ένα πρόγραμμα που θα ελέγχεται από ένα μενού με τις ακόλουθες επιλογές:

1. Επιλογή προβλήματος
2. Στατιστικά προβλήματος
3. Γράφος εξετάσεων
4. Κόστος λύσης

Η λειτουργικότητα που ζητείται για κάθε μια επιλογή περιγράφεται στις ακόλουθες παραγράφους.

4.1 Επιλογή προβλήματος

Μέσω της συγκεκριμένης επιλογής το πρόγραμμα θα επιτρέπει να φορτωθούν στην μνήμη του υπολογιστή τα δεδομένα του προβλήματος που ο χρήστης θα επιλέγει κάθε φορά. Στην συνέχεια να υπολογίζει τα βασικά στοιχεία του προβλήματος όπως τον αριθμό εξετάσεων, τον αριθμό σπουδαστών, τον αριθμό εγγραφών και την πυκνότητα συγκρούσεων. Για τον υπολογισμό της πυκνότητας θα πρέπει να κατασκευαστεί ο πίνακας συγκρούσεων. Ο πίνακας συγκρούσεων είναι ένας δισδιάστατος πίνακας στον οποίο κάθε στοιχείο $c_{ij} = 1$ αν η εξέταση i βρίσκεται σε σύγκρουση με την εξέταση j ενώ ισχύει ότι $c_{ij} = 0$ σε άλλη περίπτωση. Η πυκνότητα συγκρούσεων μπορεί να υπολογιστεί διαιρώντας τον αριθμό των στοιχείων του πίνακα συχνοτήτων που έχουν την τιμή 1 με το συνολικό πλήθος των στοιχείων του πίνακα.

4.2 Στατιστικά προβλήματος

Για το πρόβλημα που έχει επιλεχθεί το πρόγραμμα να υπολογίζει και να εμφανίζει τα ακόλουθα:

- Μέσο όρο δηλωθέντων μαθημάτων ανά σπουδαστή.
- Μεγαλύτερο αριθμό δηλωθέντων μαθημάτων από ένα σπουδαστή.
- Πλήθος σπουδαστών ανά αριθμό δηλωθέντων μαθημάτων (π.χ. 20 σπουδαστές με 1 μάθημα, 30 σπουδαστές με 2 μαθήματα κ.ο.κ.).
- Μέσο όρο εγγεγραμμένων σπουδαστών ανά μάθημα.
- Μεγαλύτερο αριθμό σπουδαστών που είναι εγγεγραμμένοι σε ένα μάθημα.
- Πλήθος μαθημάτων ανά αριθμό εγγεγραμμένων σπουδαστών (π.χ. 10 μαθήματα με 1 σπουδαστή, 13 μαθήματα με 2 σπουδαστές κ.ο.κ.)

4.3 Γράφος εξετάσεων

Να κατασκευαστεί ένας γράφος (graph) με μια κορυφή για κάθε εξέταση. Σε περίπτωση που δύο εξετάσεις έχουν κοινούς σπουδαστές να τοποθετείται ακμή ανάμεσα στους κόμβους των εξετάσεων με βάρος ίσο με τον αριθμό των κοινών σπουδαστών.
Για το πρόβλημα που έχει επιλεχθεί το πρόγραμμα να υπολογίζει και να εμφανίζει τα ακόλουθα:

- Τον μικρότερο βαθμό (degree), τον μεγαλύτερο βαθμό και το μέσο όρο των βαθμών από όλους τους κόμβους.
- Τη διάμετρο του γράφου δηλαδή το μακρύτερο συντομότερο μονοπάτι μεταξύ δύο οποιοδήποτε κορυφών).

4.4 Κόστος λύσης

Για το πρόβλημα που έχει επιλεγεί να φορτώνεται μια λύση και να εμφανίζεται το κόστος στο οποίο αντιστοιχεί σύμφωνα με την περιγραφή που υπάρχει για τον υπολογισμό του κόστους στην παράγραφο 2.

5 Παρατηρήσεις

- Η υλοποίηση του κώδικα θα πρέπει να γίνει κατά προτίμηση στην γλώσσα προγραμματισμού C++. Εναλλακτικά μπορεί να χρησιμοποιηθεί η Java ή η Python.
- Επιτρέπεται η χρήση επιπλέον βιβλιοθηκών.
- Η εργασία μπορεί να γίνει σε ομάδες το πολύ των 2 ατόμων.
- Η εργασία θα συμμετέχει στον τελικό βαθμό του μαθήματος σε ποσοστό 30%.
- Επιπλέον πληροφορίες για το πρόβλημα του χρονοπρογραμματισμού εξετάσεων μπορούν να ληφθούν από τις ακόλουθες αναφορές [BP02], [QBM+09].

5.1 Σχετικές ιστοσελίδες

- `http://www.cs.nott.ac.uk/~rxq/data.htm`
- `http://nemertes.lis.upatras.gr/jspui/handle/10889/6721`

Αναφορές

[BP02] Edmund Kieran Burke and Sanja Petrovic. Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2):266–280, 2002.

[CLL96] Michael W Carter, Gilbert Laporte, and Sau Yan Lee. Examination timetabling: Algorithmic strategies and applications. *Journal of the Operational Research Society*, pages 373–383, 1996.

[QBM+09] Rong Qu, Edmund K Burke, Barry McCollum, Liam TG Merlot, and Sau Y Lee. A survey of search methodologies and automated system development for examination timetabling. *Journal of scheduling*, 12(1):55–89, 2009.