

## **K08 Δομές Δεδομένων και Τεχνικές Προγραμματισμού**

**Διδάσκων: Μανόλης Κουμπάρκης**

**Εαρινό Εξάμηνο 2016-2017.**

**Άσκηση 3 (ανακοινώθηκε στις 24 Απριλίου 2017, προθεσμία παράδοσης: 2 Ιουνίου 2017, 12 τα μεσάνυχτα).**

**15% του συνολικού βαθμού στο μάθημα.**

1. Σχεδιάστε το B-δένδρο που προκύπτει αν εισάγουμε τα κλειδιά 4, 40, 23, 50, 11, 34, 62, 78, 66, 22, 90, 59, 25, 72, 64, 77, 39, 12 με αυτή τη σειρά σε ένα αρχικά κενό B-δένδρο τάξεως 7. Μετά σχεδιάστε το B-δένδρο που θα προκύψει αν διαγράψουμε όλα τα στοιχεία της ρίζας από το μικρότερο προς το μεγαλύτερο.

**(10+5=15 μονάδες)**

2. Θεωρήστε τον κώδικα της Ενότητας 10 που υλοποιεί την δομή του B-δένδρου. Να προσθέσετε στον κώδικα μια συνάρτηση η οποία παίρνει σαν όρισμα ένα δείκτη στη ρίζα ενός B-δένδρου και ένα φυσικό αριθμό  $k \geq 0$  και επιστρέφει το  $k$ -οστό μικρότερο στοιχείο του B-δένδρου (έχουμε δώσει μια παρόμοια συνάρτηση `select` στον κώδικα της Ενότητας 9 για δυαδικά δένδρα αναζήτησης). Μετά να προσθέσετε στον κώδικα μια συνάρτηση η οποία παίρνει σαν όρισμα ένα δείκτη στη ρίζα ενός B-δένδρου και ένα κλειδί  $a$ , και επιστρέφει το κλειδί που προηγείται του  $a$  στην φυσική διάταξη των κλειδιών του δένδρου. Να προσθέσετε κατάλληλο κώδικα στη `main` που θα ελέγχει την συμπεριφορά των συναρτήσεων που αναπτύξατε.

**(50+30=80 μονάδες)**

3. Θεωρήστε ένα μη κατευθυνόμενο γράφο  $G = (V, E)$  με σύνολο κορυφών  $V = \{0, 1, 2, \dots, 9\}$  και σύνολο ακμών  $E = \{(3, 7), (1, 4), (7, 8), (0, 5), (5, 2), (3, 8), (2, 9), (0, 6), (4, 9), (2, 6), (6, 4)\}$ . Να σχεδιάσετε τον παραπάνω γράφο. Να δώσετε την ακολουθία κορυφών που προκύπτει αν διασχίσουμε το γράφο χρησιμοποιώντας τους αλγόριθμους DFS (με αναδρομική κλήση) και BFS από τις διαφάνειες της Ενότητας 13. Να

υποθέσετε ότι ο αλγόριθμος ξεκινάει από τον κόμβο 0 και η σειρά των ακμών στις λίστες γειτνίασης είναι αριθμητική.

**(5+5+5=15 μονάδες)**

4. Θεωρήστε τον κώδικα της Ενότητας 13 για κατευθυνόμενους γράφους με χρήση λιστών γειτνίασης. Να τροποποιήσετε τον αλγόριθμο DFS ώστε να εκτυπώνει όλες τις ακμές (edges) που διασχίζει και να τις ταξινομεί σε tree, back, forward ή cross edges.

**(50 μονάδες)**

5. Θεωρήστε τον κατευθυνόμενο γράφο  $G = (V, E)$  με σύνολο κορυφών  $V = \{0, 1, 2, \dots, 12\}$  και σύνολο ακμών  $E = \{(2, 3), (0, 6), (0, 1), (2, 0), (11, 12), (9, 12), (9, 10), (9, 11), (3, 5), (8, 7), (5, 4), (0, 5), (6, 4), (6, 9), (7, 6)\}$ .

Να σχεδιάσετε τον δοσμένο γράφο και να δώσετε μια τοπολογική ταξινόμηση του (topological sort).

**(5+10=15 μονάδες)**

6. Θεωρήστε τον κατευθυνόμενο γράφο  $G = (V, E)$  με σύνολο κορυφών  $V = \{0, 1, 2, \dots, 12\}$  και σύνολο ακμών  $E = \{(0, 1), (0, 5), (0, 6), (2, 0), (2, 3), (3, 5), (3, 2), (4, 2), (4, 3), (4, 11), (5, 4), (6, 4), (6, 9), (7, 6), (7, 8), (8, 7), (8, 9), (9, 10), (9, 11), (10, 12), (11, 12), (12, 9)\}$ .

Να σχεδιάσετε τον δοσμένο γράφο και να δώσετε τις ισχυρά συνεκτικές συνιστώσες του (strongly connected components).

**(5+10=15 μονάδες)**

7. Να υλοποιήσετε τον αλγόριθμο υπολογισμού των ισχυρά συνεκτικών συνιστωσών ενός κατευθυνόμενου γράφου που δώσαμε στις διαφάνειες της Ενότητας 13 (Graphs).

**(50 μονάδες)**

8. Να υλοποιήσετε τον αλγόριθμο των Prim-Jarnik για τον υπολογισμό του ελάχιστου δέντρου επικάλυψης ενός δοσμένου μη-κατευθυνόμενου γράφου.

**(50 μονάδες)**

9. Να υπολογίσετε την υπολογιστική πολυπλοκότητα χειρίστης περίπτωσης του αλγόριθμου σας για το προηγούμενο ερώτημα (δηλ. θα υπολογίσετε  $O(\dots)$  με

παραμέτρους  $e$  τον αριθμό των ακμών και  $n$  τον αριθμό των κορυφών του γράφου).

**(15 μονάδες)**

10. Θεωρήστε τον κώδικα της Ενότητας 13 για γράφους με χρήση λιστών γειτνίασης. Να προσθέσετε μια συνάρτηση (κατά προτίμηση αναδρομική) η οποία παίρνει σαν είσοδο ένα γράφο και δύο κορυφές και επιστρέφει ένα απλό μονοπάτι που συνδέει αυτές τις κορυφές ή μας πληροφορεί ότι τέτοιο μονοπάτι δεν υπάρχει. Ένα μονοπάτι λέγεται **απλό** εάν όλες οι κορυφές και οι ακμές από τις οποίες αποτελείται είναι διαφορετικές μεταξύ τους. Γράψτε μια κατάλληλη *main* για την επίδειξη των δυνατοτήτων του προγράμματος σας.

**(50 μονάδες)**

11. Ένας τρόπος να ορίσουμε ένα συγκοινωνιακό δίκτυο (π.χ. το Μετρό της Αθήνας) είναι χρησιμοποιώντας ένα σύνολο ακολουθιών από κορυφές, όπου κάθε ακολουθία ορίζει ένα μονοπάτι που συνδέει τις κορυφές. Για παράδειγμα, η ακολουθία 0-9-3-2 ορίζει τις ακμές 0-9, 9-3 και 3-2. Γράψτε ένα πρόγραμμα που θα φτιάχνει ένα γράφο συγκοινωνιών διαβάζοντας ένα αρχείο εισόδου που περιέχει μία ακολουθία σε κάθε γραμμή και χρησιμοποιεί συμβολικά ονόματα π.χ., Akropoli-Syntagma-Panepistmio. Δοκιμάστε το πρόγραμμά σας με είσοδο ένα αρχείο που κωδικοποιεί το Μετρό της Αθήνας. Μετά προσθέστε μια συνάρτηση η οποία παίρνει σαν είσοδο δύο κορυφές του γράφου (σταθμούς του Μετρό) και μας πληροφορεί πως θα πάμε από την πρώτη κορυφή στη δεύτερη. Γράψτε μια κατάλληλη *main* για την επίδειξη των δυνατοτήτων του προγράμματος σας.

**(50 μονάδες)**

12. Θεωρήστε τη μέθοδο του δυναμικού κατακερματισμού που παρουσιάζεται στις διαφάνειες της Ενότητας 11 και στο κεφάλαιο 14.5 του βιβλίου του Sedgwick. Να δώσετε τα περιεχόμενα του πίνακα κατακερματισμού που προκύπτει όταν εισάγουμε στοιχεία με κλειδιά E A S Y Q U T I O N με αυτή τη σειρά σε ένα αρχικά άδειο πίνακα. Το αρχικό μέγεθος του πίνακα είναι  $M = 4$ . Το μέγεθος διπλασιάζεται όταν ο πίνακας γεμίσει κατά το ήμισυ, και οι συγκρούσεις αντιμετωπίζονται με γραμμική διερεύνηση (linear probing). Χρησιμοποιείτε τη συνάρτηση κατακερματισμού  $11k \bmod M$  για τη

μετατροπή του  $k$ -οστού γράμματος του αλφαβήτου στην αντίστοιχη θέση του πίνακα.

**(10 μονάδες)**

13. Θεωρήστε τον κώδικα για τη μέθοδο του δυναμικού κατακερματισμού που παρουσιάζεται στις διαφάνειες της Ενότητας 11 και στο Πρόγραμμα 14.7 του βιβλίου του Sedgewick. Προσθέστε στο πρόγραμμα αυτό μια συνάρτηση `delete` η οποία διαγράφει ένα στοιχείο από τον πίνακα και τον υποδιπλασιάζει όταν φτάσει να είναι κατά τα 7/8 άδειος.

**(50 μονάδες)**

### **Παρατηρήσεις:**

- Οι λύσεις σας στα ερωτήματα που απαιτούν υλοποίηση πρέπει να είναι οργανωμένες σε `modules` της C όπως έχουμε συζητήσει στο μάθημα.
- Τα προγράμματα σας θα πρέπει να είναι όσο πιο καλά οργανωμένα γίνεται, σύμφωνα με όσα έχετε μάθει στο μάθημα «Εισαγωγή στον Προγραμματισμό».
- Τα προγράμματα σας πρέπει να «τρέχουν» στους υπολογιστές του Τμήματος με το λειτουργικό `linux` αφού μεταφραστούν με τον μεταγλωττιστή `gcc`.

**Πως να παραδώσετε τις λύσεις σας:** Οι λύσεις θα πρέπει να σταλούν στο e-mail `ddproj@di.uoa.gr` μέχρι την προθεσμία παράδοσης και να είναι οργανωμένες ως εξής. Θα στείλετε ακριβώς ένα συμπιεσμένο αρχείο. Η λύση κάθε προβλήματος που απαιτεί υλοποίηση θα είναι σε χωριστό `directory` το οποίο θα περιλαμβάνει τα `source files` που χρειάζονται, και ένα `Makefile` το οποίο θα μπορεί να χρησιμοποιηθεί για να μεταγλωττιστούν τα αρχεία σας και να παραχθεί το αντίστοιχο `executable`. Θα πρέπει να υποβάλλετε και ένα αρχείο `pdf` στο οποίο θα περιέχονται οι απαντήσεις των θεωρητικών ερωτημάτων και το οποίο, για τα προγραμματιστικά ερωτήματα, θα περιέχει όσο `documentation` χρειάζεται ώστε οι βαθμολογητές να κατανοήσουν πλήρως τη λύση σας και να τη βαθμολογήσουν ανάλογα. Αυτό θα πρέπει να γίνει ανεξάρτητα με το αν είναι τα προγράμματα σας καλά σχολιασμένα, πράγμα που συνιστάται. Τέλος, το αρχείο `pdf` θα πρέπει

να ξεκινά με το όνομα σας γραμμένο με Ελληνικούς χαρακτήρες και τον αριθμό μητρώου σας.

**Καλή Επιτυχία!**