Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Αλγόριθμοι και Πολυπλοκότητα
Διδάσκοντες: Δημήτρης Φωτάκης, Δώρα Σούλιου
3η Σειρά Γραπτών Ασκήσεων - Ημ/νία Παράδοσης 19/12/2016

Άσκηση 1: Εφαρμογές BFS και DFS (DPV 3.18 και 3.21)

(α) Δίνεται ένα δέντρο $T(V, E)$ με ρίζα μια κορυφή $r \in V$ (θεωρούμε ότι το T αναπαρίσταται με λίστα γειτνίασης). Θέλουμε να εφαρμόσουμε έναν αλγόριθμο προεπεξεργασίας γραμμικού χρόνου στο T, ώστε στη συνέχεια να απαντάμε ερωτήματα σχέσης προγόνου-απογόνου, δηλ. ερωτήματα "είναι η κορυφή u πρόγονος της κορυφής v στο T;", σε σταθερό χρόνο (μια κορυφή u είναι *πρόγονος* μια κορυφής v στο T αν η u ανήκει στο μονοπάτι $v - r$). Να διατυπώσετε έναν τέτοιο αλγόριθμο προεξεργασίας και να εξηγήσετε πως η πληροφορία που παρέχει επιτρέπει να απαντηθούν ερωτήματα σχέσης προγόνου-απογόνου σε σταθερό χρόνο.

(β) Έστω $G(V, E)$ ένα απλό κατευθυνόμενο ισχυρά συνεκτικό γράφημα. Να διατυπώσετε έναν αλγόριθμο γραμμικού χρόνου ο οποίος υπολογίζει έναν (απλό κατευθυνόμενο) κύκλο περιττού μήκους στο G (ή διαπιστώνει ότι το G δεν περιέχει κατευθυνόμενους κύκλους περιττού μήκους).

Άσκηση 2: Μια Συνάρτηση Κόστους σε Κατευθυνόμενα Γραφήματα (DPV 3.25)

Θεωρούμε ένα κατευθυνόμενο γράφημα $G(V, E)$ στο οποίο κάθε κορυφή $u \in V$ έχει μία θετική τιμή $p(u) \in \mathbb{N}$. Το κόστος $c(u)$ κάθε κορυφής $u \in V$ είναι η τιμή της φθηνότερης κορυφής που είναι προσπελάσιμη από τη u (συμπεριλαμβανομένης και της ίδιας της u). Να διατυπώσετε αλγόριθμο γραμμικού χρόνου που υπολογίζει το κόστος $c(u)$ για όλες τις κορυφές $u \in V$. Να αιτιολογήσετε την ορθότητα των αλγορίθμων στα (α) και (β).

(α) Να διατυπώσετε αλγόριθμο γραμμικού χρόνου για την περίπτωση που το G είναι ένα Κατευθυνόμενο Ακυκλικό Γράφημα (DAG).

(β) Να γενικεύσετε τον αλγόριθμο του (α) ώστε να εφαρμόζεται σε κάθε κατευθυνόμενο γράφημα G. Μπορείτε να χρησιμοποιήσετε ότι οι ισχυρά συνεκτικές συνιστώσες ενός κατευθυνόμενου γραφήματος G μπορούν να υπολογιστούν σε γραμμικό χρόνο (είναι σημαντικό και ενδιαφέρον να δείτε πως στις ενότητες DPV 3.4 και CLRS 22.5).

Άσκηση 3: Ανάλυση Ασφάλειας (KT 3.11)

Ένα εταιρικό δίκτυο αποτελείται από n υπολογιστές $C_1, \ldots C_n$. Έπειτα από μία προσβολή του δικτύου από ιό, θέλουμε να μελετήσουμε την εξάπλωση του ιού στο δίκτυο. Από την επεξεργασία των log files, έχουμε καταλήξει σε m τριάδες της μορφής (C_i, C_j, t), οι οποίες δίνονται σε αύξουσα χρονική σειρά, που δηλώνουν ότι οι υπολογιστές C_i και C_j επικοινώνησαν μεταξύ τους τη χρονική στιγμή t. Γνωρίζουμε ότι αν δύο υπολογιστές C_i και C_j επικοινώνησαν τη χρονική στιγμή t και (μόνον) ο ένας είχε μολυνθεί από τον ιό κάποια στιγμή $t' \leq t$, τότε μολύνθηκε και ο άλλος τη χρονική στιγμή t. Γνωρίζουμε ακόμη ότι ο ιός εισήλθε στο δίκτυο από τον υπολογιστή C_1, ο οποίος μολύνθηκε τη χρονική στιγμή $t_1 = 0$. Για κάθε υπολογιστή C_j, θέλουμε να υπολογίσουμε τη χρονική στιγμή t_j κατά την οποία μολύνθηκε (ή να διαπιστώσουμε ότι ο C_j δεν έχει μολυνθεί από τον ιό). Να διατυπώσετε έναν αλγόριθμο χρόνου $\Theta(n+m)$ για αυτό το πρόβλημα και να αιτιολογήσετε την ορθότητα και την υπολογιστική του πολυπλοκότητα. *Παράδειγμα:* Αν έχουμε $n = 4$ υπολογιστές και $m = 4$ τριάδες, τις $(C_1, C_2, 2)$, $(C_2, C_3, 6)$, $(C_3, C_4, 6)$ και $(C_1, C_4, 10)$, οι χρόνοι μόλυνσης είναι $t_1 = 0$, $t_2 = 2$, $t_3 = 6$ και $t_4 = 6$ (ο C_4 μολύνεται μέσω του C_3 τη χρονική στιγμή 6).

Άσκηση 4: Το Σύνολο των Συνδετικών Δέντρων (ΚΤ 4.27 και ΚΤ 4.28)

Θεωρούμε ένα συνεκτικό μη κατευθυνόμενο γράφημα $G(V, E)$ με n κορυφές και m ακμές.

(α) Έστω T_1 και T_2 δύο διαφορετικά συνδετικά δέντρα του G. Να δείξετε ότι για κάθε ακμή $e \in T_1 \setminus T_2$, υπάρχει ακμή $e' \in T_2 \setminus T_1$, τέτοια ώστε το $(T_1 \setminus \{e\}) \cup \{e'\}$ είναι συνδετικό δέντρο. Να διατυπώσετε αποδοτικό αλγόριθμο που με δεδομένα τα T_1, T_2 και e, υπολογίζει μια τέτοια ακμή e'.

(β) Σχηματίζουμε γράφημα H που κάθε κορυφή του αντιστοιχεί σε ένα διαφορετικό συνδετικό δέντρο του G. Δύο συνδετικά δέντρα T_1 και T_2 του G (κορυφές του H) συνδέονται με ακμή στο H αν διαφέρουν κατά μία μόνο ακμή, δηλ. αν $|T_1 \setminus T_2| = |T_2 \setminus T_1| = 1$. Να δείξετε ότι το H είναι συνεκτικό και ότι η απόσταση (στο H) μεταξύ δύο συνδετικών δέντρων T_1 και T_2 του G είναι ίση με $|T_1 \setminus T_2|$. Να εξηγήσετε πως θα χρησιμοποιήσουμε τον αλγόριθμο του (α) για να υπολογίσουμε ένα συντομότερο μονοπάτι (στο H) μεταξύ των T_1 και T_2.

(γ) Θεωρούμε μια διαμέριση των ακμών του G σε δύο σύνολα E_1 και E_2. Να διατυπώσετε αποδοτικό αλγόριθμο που με είσοδο το $G(V, E)$, τα σύνολα E_1 και E_2, και έναν φυσικό k, $1 \le k \le n-1$, υπολογίζει ένα συνδετικό δέντρο του G με ακριβώς k ακμές από το σύνολο E_1. Αν δεν υπάρχει τέτοιο δέντρο, ο αλγόριθμός σας θα πρέπει να το διαπιστώνει.

Άσκηση 5: Μοναδικότητα Ελάχιστου Συνδετικού Δέντρου

Θεωρούμε ένα συνεκτικό μη κατευθυνόμενο γράφημα $G(V, E, w)$ με βάρη στις ακμές. Είναι γνωστό (π.χ. δείτε το 6.α, στην 3η σειρά προτεινόμενων ασκήσεων) ότι αν όλες οι ακμές του G έχουν διαφορετικά βάρη, τότε το Ελάχιστο Συνδετικό Δέντρο (ΕΣΔ) του G είναι μοναδικό.

(α) Να δείξετε ότι το αντίστροφο δεν ισχύει. Δηλαδή, να δώσετε παράδειγμα γραφήματος με μοναδικό ΕΣΔ, το οποίο έχει κάποιες ακμές με ίδιο βάρος.

(β) Να δείξετε ότι αν για κάθε τομή $(S, V \setminus S)$ του $G(V, E, w)$, η ακμή ελάχιστου βάρους που διασχίζει την $(S, V \setminus S)$ είναι μοναδική, τότε το G έχει μοναδικό ΕΣΔ. Όπως και στο (α), να δείξετε ότι το αντίστροφο δεν ισχύει.

(γ) Να διατυπώσετε μια ικανή και αναγκαία συνθήκη για τη μοναδικότητα του ΕΣΔ σε ένα συνεκτικό μη κατευθυνόμενο γράφημα $G(V, E, w)$ (και να αποδείξετε ότι η συνθήκη που διατυπώσατε είναι πράγματι ικανή και αναγκαία).

(δ) Να διατυπώσετε αλγόριθμο με χρονική πολυπλοκότητα $\mathrm{O}(|V|^2 + |E| \log |E|)$ που ελέγχει κατά πόσο ένα συνεκτικό μη κατευθυνόμενο γράφημα $G(V, E, w)$ έχει μοναδικό ΕΣΔ. Να αιτιολογήσετε αναλυτικά την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας. Θα υπάρχει επιπλέον βαθμολογία (bonus) για απαντήσεις με χρονική πολυπλοκότητα $\mathrm{O}(|E| \log |E|)$.