Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Αλγόριθμοι και Πολυπλοκότητα
3η Σειρά Προτεινόμενων Ασκήσεων

Άσκηση 1: Γραφηματικές Ακολουθίες

Η *ακολουθία βαθμών* (degree sequence) ενός γραφήματος είναι η ακολουθία των βαθμών των κορυφών του, συνήθως σε φθίνουσα σειρά. Μια ακολουθία $n \geq 1$ φυσικών αριθμών ονομάζεται *γραφηματική* (graphic sequence) αν αποτελεί την ακολουθία βαθμών ενός *απλού* μη κατευθυνόμενου γραφήματος n κορυφών. Η μοναδική γραφηματική ακολουθία με ένα στοιχείο είναι η $d_1 = 0$.

Να δείξετε ότι μια ακολουθία $n > 1$ φυσικών $\boldsymbol{d} = (d_1, \dots, d_n)$, $d_1 \geq \cdots \geq d_n$, είναι γραφηματική αν και μόνο αν η ακολουθία $\boldsymbol{d}'$ που προκύπτει από την $\boldsymbol{d}$ αν αφαιρέσουμε το μεγαλύτερο στοιχείο d_1 και μειώσουμε τα d_1 επόμενα μεγαλύτερα στοιχεία της $\boldsymbol{d}$ κατά 1 είναι επίσης γραφηματική.

Λύση. Έστω $\boldsymbol{d} = (d_1, \dots, d_n)$, $d_1 \geq \cdots \geq d_n$, ακολουθία $n > 1$ φυσικών αριθμών, και έστω $\boldsymbol{d}'$ η ακολουθία που προκύπτει από την $\boldsymbol{d}$ αν αφαιρέσουμε το d_1 και μειώσουμε τα d_1 επόμενα μεγαλύτερα στοιχεία $d_2, \dots, d_{d_1+1}$ της κατά 1. Πρέπει βέβαια $d_1 \leq n-1$, διαφορετικά η $\boldsymbol{d}'$ δεν θα ήταν καλώς ορισμένη.

Θα δείξουμε πρώτα ότι η συνθήκη είναι ικανή. Έστω ότι η ακολουθία $\boldsymbol{d}'$ είναι γραφηματική, και έστω G' απλό γράφημα με $n-1$ κορυφές που έχει ως ακολουθία βαθμών την $\boldsymbol{d}'$. Αν προσθέσουμε στο G' μια νέα κορυφή βαθμού d_1 και την συνδέσουμε με τις d_1 κορυφές του G' που έχουν βαθμό $d_2 - 1, \dots, d_{d_1+1} - 1$, προκύπτει ένα απλό γράφημα G με n κορυφές που έχει ως ακολουθία βαθμών την $\boldsymbol{d}$. Άρα και η $\boldsymbol{d}$ είναι γραφηματική.

Στη συνέχεια αποδεικνύουμε ότι η συνθήκη είναι αναγκαία. Έστω ότι η ακολουθία $\boldsymbol{d}$ είναι γραφηματική, και έστω G απλό γράφημα με n κορυφές που έχει ως ακολουθία βαθμών την $\boldsymbol{d}$. Θεωρούμε κορυφή v του G με βαθμό d_1. Έστω $N(v)$ το σύνολο των γειτόνων της v στο G, και έστω S σύνολο d_1 κορυφών του G με (τους επιθυμητούς) βαθμούς $d_2, \dots, d_{d_1+1}$ αντίστοιχα (αν υπάρχουν πολλά τέτοια σύνολα, επιλέγουμε αυτό με το μέγιστο πλήθος κορυφών του $N(v)$). Αν $N(v) = S$, αφαιρώντας την v από το G προκύπτει απλό γράφημα G' με $n-1$ κορυφές που έχει ως ακολουθία βαθμών την $\boldsymbol{d}'$.

Αν $N(v) \neq S$, θα τροποποιήσουμε το G ώστε να αυξήσουμε το πλήθος των κορυφών του S που ανήκουν στο $N(v)$, δηλ. το $|N(v) \cap S|$, χωρίς να μεταβάλουμε τον βαθμό καμία κορυφής. Για αυτό, θεωρούμε κορυφές $x \in S \setminus N(v)$ και $y \in N(v) \setminus S$. Αφού ο βαθμός της x είναι μεγαλύτερος από τον βαθμό της y (επειδή η y είναι γείτονας της v αλλά δεν ανήκει στο S), υπάρχει κορυφή z που συνδέεται με την x αλλά όχι με την y. Τροποποιούμε το G αντικαθιστώντας την ακμή $\{z, x\}$ με την ακμή $\{v, x\}$ και την ακμή $\{v, y\}$ με την ακμή $\{z, y\}$. Οι βαθμοί των εμπλεκόμενων κορυφών δεν αλλάζουν και το γράφημα που προκύπτει είναι απλό. Επαναλαμβάνοντας αυτή την διαδικασία όσες φορές χρειαστεί, καταλήγουμε σε ένα απλό γράφημα $\hat{G}$ με n κορυφές και ακολουθία βαθμών την $\boldsymbol{d}$, όπου η κορυφή μέγιστου βαθμού v έχει $N(v) = S$. Αφαιρώντας την v από το $\hat{G}$ προκύπτει απλό γράφημα με ακολουθία βαθμών την $\boldsymbol{d}'$. $\square$

Άσκηση 2: Ολική Καταβόθρα

Ολική καταβόθρα σε ένα κατευθυνόμενο γράφημα λέγεται μια κορυφή που δεν έχει εξερχόμενες ακμές και έχει εισερχόμενη ακμή από κάθε άλλη κορυφή. Θεωρούμε γράφημα με $n \geq 2$ κορυφές που αναπαρίσταται με πίνακα γειτνίασης:

1. Να διατυπώσετε αλγόριθμο με χρόνο εκτέλεσης $O(n^2)$ που βρίσκει μια ολική καταβόθρα ή αποφαίνεται ότι δεν υπάρχει.
2. Να διατυπώσετε αλγόριθμο με χρόνο εκτέλεσης $O(n)$ που βρίσκει μια ολική καταβόθρα ή να αποφαίνεται ότι δεν υπάρχει.

Λύση. (1) Μπορούμε να ελέγξουμε για κάθε κορυφή χωριστά αν είναι ολική καταβόθρα ή όχι. Δεδομένου ότι ο έλεγχος για κάθε κορυφή μπορεί να γίνει στο $O(n)$, ο συνολικός χρόνος θα είναι $O(n^2)$.

(2) Το πρόβλημα μπορεί να λυθεί σε χρόνο $O(n)$ αν εκμεταλλευθούμε ότι (α) κάθε γράφημα έχει το πολύ μία ολική καταβόθρα, και (β) αν ένα γράφημα έχει ολική καταβόθρα, τότε κάθε μονοπάτι οδηγεί σε αυτή.

Έστω $v_1, \ldots, v_n$ οι κορυφές του γραφήματος. Η ιδέα του αλγόριθμου είναι η εξής: Ακολουθούμε μια διαδρομή όπου οι κορυφές εμφανίζονται σε αύξουσα σειρά των δεικτών τους. Συγκεκριμένα, ξεκινάμε από την κορυφή v_1, και συνεχίζουμε στην αμέσως μεγαλύτερη γειτονική της (αν υπάρχει). Συνεχίζουμε έτσι μέχρι να καταλήξουμε σε κορυφή v_i που δεν έχει ακμή προς καμία κορυφή v_j με $j > i$. Παρατηρούμε ότι αν το γράφημα έχει ολική καταβόθρα αυτή είναι η κορυφή v_i (γιατί; να αποδείξετε αυτόν τον ισχυρισμό). Τόσο ο εντοπισμός της κορυφής v_i όσο και ο έλεγχος για το αν είναι ολική καταβόθρα υλοποιούνται σε χρόνο $O(n)$. □

Άσκηση 3: Εξισορρόπηση Δέντρου

Έστω $T(V, E)$ δέντρο με θετικά βάρη $w : V \mapsto \mathbb{N}^*$ στις κορυφές του. Θέλουμε να επιλέξουμε μια κορυφή v του T ως ρίζα, ώστε το συνολικό βάρος των κορυφών στο βαρύτερο υποδέντρο της v να είναι το ελάχιστο δυνατό. Πιο συγκεκριμένα, θεωρώντας μια κορυφή v με βαθμό $d(v)$ ως ρίζα του T, διαμερίζουμε τις κορυφές του T στην ρίζα v και στις κορυφές των $d(v)$ υποδέντρων $T_{v_1}, \ldots, T_{v_{d(v)}}$ με ρίζες τους γείτονες $v_1, \ldots, v_{d(v)}$ της v στο T. Το συνολικό βάρος των κορυφών σε κάθε υποδέντρο είναι $w(T_{v_1}), \ldots, w(T_{v_{d(v)}})$. Θεωρούμε ότι η καταλληλότερη ρίζα για το T είναι η κορυφή v που ελαχιστοποιεί την ποσότητα $b(v) = \max_{1 \leq i \leq d(v)}\{w(T_{v_i})\}$, δηλ. το συνολικό βάρος των κορυφών στο βαρύτερο υποδέντρο της v.

Να διατυπώσετε έναν όσο το δυνατόν πιο αποδοτικό αλγόριθμο που δεδομένου ενός δέντρου $T(V, E)$ και του βάρους $w(u)$ κάθε κορυφής $u \in V$, υπολογίζει μια κορυφή v με ελάχιστο $b(v)$.

Λύση. Η επιλογή μιας κορυφής v με ελάχιστο $b(v)$ μπορεί να γίνει σε χρόνο $O(n)$. Επιλέγουμε αυθαίρετα μία κορυφή v ως υποψήφια ρίζα του δέντρου. Στο T με ρίζα την v, υπολογίζουμε σε γραμμικό χρόνο (εφαρμόζοντας postorder) το συνολικό βάρος των κορυφών σε κάθε υποδέντρο του T, και αποθηκεύουμε την αντίστοιχη τιμή στην ρίζα κάθε υποδέντρου. Έτσι για κάθε κορυφή u, έχουμε υπολογίσει και αποθηκεύσει το συνολικό βάρος $w(T_u)$ των κορυφών που είναι απόγονοί της u στο T με ρίζα την v (θεωρούμε ότι το $w(T_u)$ περιλαμβάνει το $w(u)$).

Έστω $d(v)$ ο βαθμός της v, έστω $v_1, \ldots, u_{d(v)}$ οι γειτονές της v στο T, και έστω $T_{v_1}, \ldots, T_{v_{d(v)}}$ τα αντίστοιχα υποδέντρα. Χωρίς βλάβη της γενικότητας, θεωρούμε ότι το T_{v_1} είναι το βαρύτερο υποδέντρο της v και ότι $b(v) = w(T_{v_1})$. Παρατηρούμε ότι για κάθε γείτονα v_i της v, $b(v_i) \geq w(v) + \sum_{j \neq i} w(T_{v_j})$. Επομένως κάθε γείτονας v_i της v, εκτός της v_1, έχει $b(v_i) \geq b(v)$. Συνεπώς η κορυφή που ψάχνουμε είτε είναι η v είτε βρίσκεται στο βαρύτερο υποδέντρο T_{v_1} της v. Πιο συγκεκριμένα, αν $w(T_{v_1}) \leq w(v) + \sum_{j=2}^{d(v)} w(T_{v_j})$, η κορυφή v έχει ελάχιστο $b(v)$ και είναι η κορυφή που ψάχνουμε. Διαφορετικά, η κορυφή που ψάχνουμε βρίσκεται στο βαρύτερο υποδέντρο T_{v_1} της v (γιατί; να αποδείξετε αυτόν τον ισχυρισμό). Σε αυτή την περίπτωση, θέτουμε την v_1 στην θέση της v (δηλ. στην θέση της υποψήφιας ρίζας του T), υπολογίζουμε το $b(v_1)$ ως μέγιστο του $w(v) + \sum_{j=2}^{d(v)} w(T_{v_j})$, που είναι το βάρος του υποδέντρου της v_1 με ρίζα την v, και των βαρών των υποδέντρων της v_1 με ρίζες τις άλλες γειτονικές της κορυφές, και επαναλαμβάνουμε την παραπάνω διαδικασία.

Συνολικά έχουμε το πολύ $|V|$ επαναλήψεις. Ο χρόνος για κάθε επανάληψη ποικίλει, ανάλογα με το πλήθος των γειτόνων της υποψήφια ρίζας. Όμως σε κάθε επανάληψη, η νέα ρίζα v_1 βρίσκεται σε κάποιο υποδέντρο της τρέχουσας ρίζας v, και έτσι το $b(v_1)$ μπορεί να υπολογιστεί από την υπάρχουσα πληροφορία, χωρίς να απαιτείται να υπολογίσουμε τα βάρη των υποδέντρων από την αρχή. Έτσι η πληροφορία που έχουμε για κάθε κορυφή u αρχικά (δηλ. το βάρος της $w(u)$ και το συνολικό βάρος $w(T_u)$ των απογόνων της u στο T με ρίζα την πρώτη υποψήφια κορυφή) εξετάζεται $\mathrm{O}(1)$ φορές σε όλη την διάρκεια του αλγόριθμου. Επομένως ο χρόνος εκτέλεσης είναι $\mathrm{O}(|V|)$. □

Άσκηση 4: Ενημέρωση Ελάχιστου Συνδετικού Δέντρου (DPV 5.23)

Θεωρούμε μη κατευθυνόμενο γράφημα $G(V, E, w)$ με θετικά βάρη στις ακμές, και ένα Ελάχιστο Συνδετικό Δέντρο $T(V, E')$ του G. Να διατυπώσετε αλγόριθμο *γραμμικού χρόνου* που τροποποιεί κατάλληλα το T ώστε να παραμείνει Ελάχιστο Συνδετικό Δέντρο όταν το βάρος μιας ακμής $e \in E$ αλλάζει από $w(e)$ σε $\hat{w}(e)$. Για την διατύπωση και τον προσδιορισμό του χρόνου εκτέλεσης του αλγορίθμου σας, να υποθέσετε ότι τόσο το G όσο και το T αναπαρίστανται με λίστες γειτνίασης.

Λύση. Διακρίνουμε 4 περιπτώσεις, ανάλογα με το αν η e ανήκει στο T ή όχι, και αν το βάρος της αυξάνεται ή μειώνεται. Το T μπορεί να αλλάξει μόνο αν (i) $e \in T$ και το βάρος της αυξάνεται, και (ii) $e \notin T$ και το βάρος της μειώνεται.

Στην περίπτωση (i), αφαιρούμε την e, "σπάζοντας" έτσι το T σε δύο συνεκτικές συνιστώσες T_1 και T_2 (ο υπολογισμός των T_1 και T_2 γίνεται π.χ. με BFS σε χρόνο $\Theta(n)$). Σε χρόνο $\Theta(m)$, με ένα πέρασμα των ακμών, υπολογίζουμε την ακμή ελάχιστου βάρους e' που διασχίζει την τομή που ορίζεται από τις κορυφές των T_1 και T_2. Αν $w(e') < \hat{w}(e)$, ενημερώνουμε το T προσθέτοντας την e' και αφαιρώντας την e. Διαφορετικά, διατηρούμε το T.

Στην περίπτωση (ii), προσθέτουμε την e στο T, και βρίσκουμε (π.χ. με DFS σε χρόνο $\mathrm{O}(n)$) τον κύκλο C που σχηματίζεται. Σε χρόνο $\mathrm{O}(n)$, ελέγχουμε αν υπάρχει ακμή $e' \in C$ με βάρος μεγαλύτερο του $\hat{w}(e)$. Αν ναι, ενημερώνουμε το T προσθέτοντας την e και αφαιρώντας την e'. Διαφορετικά, διατηρούμε το T. □

Άσκηση 5: Bottleneck Shortest Path Tree (KT 4.19)

Θεωρούμε ένα απλό *μη κατευθυνόμενο* γράφημα $G(V, E, c)$ με χωρητικότητες $c(e)$ στις ακμές του $e \in E$, και αξιολογούμε τα μονοπάτια μεταξύ δύο κορυφών u, v με βάση την χωρητικότητά τους. Η χωρητικότητα ενός $u - v$ μονοπατιού p καθορίζεται από την ελάχιστη χωρητικότητα των ακμών του, και είναι $C(p) = \min_{e \in p}\{c(e)\}$. Το ζητούμενο είναι να υπολογίσουμε ένα (όσο το δυνατόν πιο αραιό) συνδετικό υπογράφημα του G που για κάθε ζευγάρι κορυφών $u, v \in V$, περιέχει ένα $u - v$ μονοπάτι μέγιστης χωρητικότητας.

(α) Να δείξετε ότι υπάρχει συνδετικό *δέντρο* T του G στο οποίο για κάθε ζευγάρι κορυφών $u, v \in V$, το μοναδικό $u - v$ μονοπάτι στο T αποτελεί ένα $u - v$ μονοπάτι μέγιστης χωρητικότητας.

(β) Με βάση το (α), να διατυπώσετε έναν αποδοτικό αλγόριθμο για τον υπολογισμό ενός τέτοιου συνδετικού δέντρου.

Λύση. (α) Έστω T ένα *Μέγιστο Συνδετικό Δέντρο* (ΜΣΔ, Maximum Spanning Tree) του G, δηλ. ένα συνδετικό δέντρο με μέγιστο συνολικό βάρος. Θα δείξουμε ότι για κάθε ζευγάρι κορυφών $u, v \in V$, το μοναδικό $u - v$ μονοπάτι στο T αποτελεί ένα $u - v$ μονοπάτι μέγιστης χωρητικότητας.

Για να καταλήξουμε σε άτοπο, θεωρούμε ζεύγος κορυφών u, v για τις οποίες το $u - v$ μονοπάτι μέγιστης χωρητικότητας p^* έχει χωρητικότητα μεγαλύτερη από την χωρητικότητα του $u - v$ μονοπατιού p στο T. Έστω e μια ακμή ελάχιστης χωρητικότητας του p. Αφού $C(p^*) > C(p) = w(e)$, όλες οι ακμές του p^* έχουν βάρος μεγαλύτερο του $w(e)$. Αφαιρούμε την e από το T, "σπάζοντας" το έτσι σε δύο συνεκτικές συνιστώσες T_1 και T_2, η μία από τις οποίες περιέχει την u και η άλλη την v. Έστω e' η ακμή του p^* που διασχίζει την τομή που ορίζεται από τις κορυφές των T_1 και T_2. Αφού $w(e') > w(e)$, το $T \cup \{e'\} \setminus \{e\}$ είναι ένα συνδετικό δέντρο με συνολικό βάρος μεγαλύτερο από το βάρος του T, άτοπο.

(β) Ένα ΜΣΔ υπολογίζεται αντίστοιχα με ένα ΕΣΔ, αν σε κάθε βήμα επιλέγουμε μια μέγιστου βάρους ακμή η οποία διασχίζει μια τομή που δεν διασχίζεται από κάποια από τις ήδη επιλεγμένες ακμές. Έτσι ένα ΜΣΔ μπορεί να υπολογιστεί με τον αλγόριθμο του Kruskal, αν αρχικά οι ακμές έχουν ταξινομηθεί σε φθίνουσα σειρά βάρους, ή με τον αλγόριθμο του Prim, αν σε κάθε βήμα, επιλέγεται και εντάσσεται στο δέντρο η κορυφή με το μεγαλύτερο κόστος σύνδεσης / ένταξης. $\square$

Άσκηση 6: Δεύτερο Ελαφρύτερο Συνδετικό Δέντρο (CLRS 23.1)

Θεωρούμε μη κατευθυνόμενο συνεκτικό γράφημα $G(V, E, w)$ με θετικά βάρη στις ακμές, και υποθέτουμε ότι $|E| \geq |V|$ και ότι όλα τα βάρη των ακμών του G είναι διαφορετικά μεταξύ τους. Έστω $\mathcal{T}$ το σύνολο όλων των συνδετικών δέντρων του G, και έστω $T' \in \mathcal{T}$ ένα Ελάχιστο Συνδετικό Δέντρο (ΕΣΔ) του G. Το Δεύτερο Ελαφρύτερο Συνδετικό Δέντρο (2ο-ΕΣΔ) του G είναι ένα συνδετικό δέντρο $T \in \mathcal{T}$ τέτοιο ώστε $w(T) = \min_{T'' \in \mathcal{T} \setminus \{T'\}} \{w(T'')\}$. Με άλλα λόγια, το 2ο-ΕΣΔ T είναι ένα συνδετικό δέντρο του G που έχει βάρος μεγαλύτερο ή ίσο από το βάρος του ΕΣΔ T' και μικρότερο ή ίσο από το βάρος κάθε άλλου συνδετικού δέντρου.

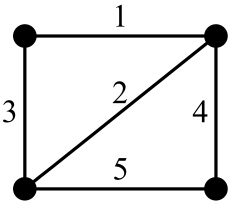
(α) Να δείξετε ότι το ΕΣΔ του G είναι μοναδικό, και ότι κάτι τέτοιο δεν ισχύει απαραίτητα για το 2ο-ΕΣΔ του G.

(β) Έστω T το ΕΣΔ του G και T' κάποιο 2ο-ΕΣΔ. Να δείξετε ότι τα T και T' διαφέρουν κατά μία μόνο ακμή, δηλ. ότι υπάρχουν ακμές $e \in T$ και $e' \notin T$ τέτοιες ώστε $T' = T \cup \{e'\} \setminus \{e\}$.

(γ) Έστω T ένα συνδετικό δέντρο του G και, για οποιοδήποτε ζεύγος κορυφών $u, v \in V$, έστω $e_{uv}^{\max}$ η ακμή μέγιστου βάρους στο μοναδικό $u - v$ μονοπάτι στο T. Να διατυπώσετε αλγόριθμο χρόνου $O(|V|^2)$ που δέχεται ως είσοδο το T και προσδιορίζει την ακμή $e_{uv}^{\max}$ για όλα τα ζεύγη κορυφών $u, v \in V$.

(δ) Να διατυπώσετε έναν αποδοτικό αλγόριθμο που υπολογίζει ένα 2ο-ΕΣΔ του G.

Λύση. (α) Για να αποδείξουμε τη μοναδικότητα του ΕΣΔ θα χρησιμοποιήσουμε απαγωγή σε άτοπο. Υποθέτουμε λοιπόν ότι το ΕΣΔ του G δεν είναι μοναδικό. Άρα υπάρχουν δύο διαφορετικά ΕΣΔ T, T' με βάρος $w(T) = w(T')$. Έστω λοιπόν $U = T \setminus T'$ το σύνολο των ακμών που ανήκουν στο T αλλά όχι στο T' και $U' = T' \setminus T$ το σύνολο των ακμών που ανήκουν στο T' αλλά όχι στο T. Αφού τα T και T' είναι διαφορετικά, τα σύνολα U και U' είναι μη κενά. Θα δείξουμε ότι αυτό συνεπάγεται την ύπαρξη ζευγαριού ακμών $e \in U$ και $e' \in U'$ τέτοιων ώστε (α) $w(e) < w(e')$ και (β) το $T' \cup \{e\} \setminus \{e'\}$ αποτελεί συνδετικό δέντρο του G.



Σχήμα 1. Παράδειγμα γραφήματος με διαφορετικά βάρη στις ακμές. Το βάρος του (μοναδικού) ΕΣΔ είναι 7. Το γράφημα έχει δύο διαφορετικά (δεύτερα καλύτερα) συνδετικά δέντρα βάρους 8.

Έστω λοιπόν e η ακμή ελάχιστου βάρους στο σύνολο $U \cup U'$. Αφού τα βάρη όλων των ακμών είναι διαφορετικά μεταξύ τους, όλες οι άλλες ακμές στο σύνολο $U \cup U'$ έχουν βάρος μεγαλύτερο της e. Ας θεωρήσουμε ότι η e ανήκει στο T (άρα δεν ανήκει στο T'). Αν προσθέσουμε την ακμή e στο T' σχηματίζεται κύκλος. Αυτός πρέπει να περιέχει κάποια ακμή $e' \in U'$, αφού διαφορετικά ο ίδιος κύκλος θα υπήρχε και στο T. Επομένως $w(e) < w(e')$. Άρα το γράφημα $T' \cup \{e\} \setminus \{e'\}$ αποτελεί ένα συνδετικό δέντρο του G με συνολικό βάρος $w(T') + w(e) - w(e') < w(T')$. Αυτό είναι άτοπο, αφού έχουμε υποθέσει ότι το T' είναι ένα ΕΣΔ.

(β) Για να καταλήξουμε σε άτοπο, υποθέτουμε ότι το ζητούμενο δεν ισχύει, δηλ. ότι υπάρχει κάποιο 2ο-ΕΣΔ T' που διαφέρει από το T σε δύο τουλάχιστον ακμές του. Έχουμε λοιπόν ότι για τα σύνολα $U = T \setminus T'$ και $U' = T' \setminus T$ ισχύει ότι $|U| \geq 2$ και $|U'| \geq 2$. Θα δείξουμε ότι αυτό συνεπάγεται την ύπαρξη συνδετικού δέντρου T'' του G που είναι διαφορετικό του T και έχει βάρος $w(T'') < w(T')$. Αυτό αντιβαίνει στην υπόθεση ότι το T' είναι ένα 2ο-ΕΣΔ του G.

Η ιδέα είναι να βρούμε ένα ζευγάρι ακμών $e \in U$ και $e' \in U'$ τέτοιων ώστε (α)$w(e) < w(e')$ και (β) το $T'' = T' \cup \{e\} \setminus \{e'\}$ αποτελεί συνδετικό δέντρο του G. Αυτό αρκεί επειδή το συνδετικό δέντρο T'' θα έχει βάρος $w(T'') < w(T')$ και θα είναι διαφορετικό από το T, αφού τα T και T' διαφέρουν κατά δύο τουλάχιστον ακμές.

Έστω λοιπόν e η ακμή ελάχιστου βάρους στο U. Αν προσθέσουμε την e στο T' σχηματίζεται κύκλος, ο οποίος πρέπει να περιέχει κάποια ακμή $e' \notin T$ (άρα $e' \in U'$, διαφορετικά ο ίδιος κύκλος θα υπήρχε και στο T). Άρα το $T'' = T' \cup \{e\} \setminus \{e'\}$ αποτελεί συνδετικό δέντρο του G.

Απομένει να δείξουμε ότι $w(e) < w(e')$. Η προσθήκη της e' στο T δημιουργεί κύκλο, ο οποίος πρέπει να περιέχει κάποια ακμή $\hat{e} \notin T'$ (άρα $\hat{e} \in U$). Επειδή το T είναι ένα ΕΣΔ και τα βάρη όλων των ακμών είναι διαφορετικά μεταξύ τους, πρέπει $w(\hat{e}) < w(e')$ (αν $w(e') < w(\hat{e})$, το συνδετικό δέντρο $T \cup \{e'\} \setminus \{\hat{e}\}$ θα είχε βάρος μικρότερο από το βάρος του T). Αφού η e είναι η ακμή ελάχιστου βάρους του U και $\hat{e} \in U$, ισχύει ότι $w(e) \leq w(\hat{e}) < w(e')$.

(γ) Για κάθε κορυφή $u \in V$, ο υπολογισμός των ακμών $e_{uv}^{\max}$ (και των αντίστοιχων βαρών) για κάθε $v \in V$ μπορεί να γίνει σε χρόνο $\mathrm{O}(|V|)$ με ένα πέρασμα του δέντρου T. Ειδικότερα, θεωρούμε ως ρίζα την κορυφή u, και επισκεπτόμαστε συστηματικά όλες τις κορυφές του T σε αύξουσα απόσταση από την u (για αυτό μπορούμε να χρησιμοποιήσουμε BFS ή DFS με ρίζα την u). Για κάθε κορυφή v που είναι άμεσος απόγονος της u στο T (δηλ. για κάθε v τέτοια ώστε $(u, v) \in T$), η $e_{uv}^{\max}$ είναι η ακμή (u, v). Για κάθε άλλη κορυφή v, έστω x ο άμεσος πρόγονός της v στο T (δηλ. η ακμή $(x, v) \in T$ αποτελεί την πρώτη ακμή του μονοπατιού από την v προς την ρίζα u, έτσι το "πέρασμά" μας φτάνει στην v μέσω της ακμής (x, v)). Τότε η $e_{uv}^{\max}$ είναι η βαρύτερη από τις ακμές (x, v) και $e_{ux}^{\max}$.

Επαναλαμβάνουμε την παραπάνω διαδικασία $|V|$ φορές, κάθε φορά θεωρούμε ως ρίζα μια διαφορετική κορυφή $u \in V$, και υπολογίζουμε τις ακμές $e_{uv}^{\max}$ για όλα τα ζεύγη κορυφών $u, v \in V$ σε χρόνο $\mathrm{O}(|V|^2)$ (δείτε ακόμη ότι μπορούμε να αποφύγουμε κάποιους υπολογισμούς (πόσους και ποιούς;) λόγω συμμετρίας).

(δ) Έστω T το ΕΣΔ του G. Από το (β), γνωρίζουμε ότι ένα 2ο-ΕΣΔ προκύπτει αν αντικαταστήσουμε κάποια ακμή $e \in T$ με κάποια ακμή $e' \notin T$ τέτοια ώστε το $T' = T \cup \{e'\} \setminus \{e\}$ να είναι συνδετικό δέντρο. Για να είναι το T' συνδετικό δέντρο, πρέπει αν η ακμή που προσθέτουμε είναι η $e' = (u, v)$ (δηλ. συνδέει τις κορυφές u και v), η ακμή e που αφαιρούμε να ανήκει στο μονοπάτι μεταξύ u και v στο ΕΣΔ T. Επιπλέον, αφού το βάρος $w(T') = w(T) + w(e') - w(e)$ του T' πρέπει να είναι ελάχιστο, το ζευγάρι ακμών e' και e πρέπει να έχει ελάχιστη διαφορά βάρους $w(e') - w(e)$. Άρα η ακμή που αφαιρούμε πρέπει να είναι η ακμή μέγιστου βάρους $e_{uv}^{\max}$ στο μονοπάτι μεταξύ u και v στο T. Έτσι επιλέγουμε το ζεύγος ακμών $(u, v) \notin T$ και $e_{uv}^{\max}$ με ελάχιστη διαφορά βάρους $w(u, v) - w(e_{uv}^{\max})$. Με βάση τα παραπάνω, το συνδετικό δέντρο $T \cup \{(u, v)\} \setminus \{e_{uv}^{\max}\}$ αποτελεί ένα 2ο-ΕΣΔ.

Η παραπάνω μεθοδολογία υπολογισμού ενός 2ου-ΕΣΔ μπορεί να υλοποιηθεί σε χρόνο $\mathrm{O}(|V|^2)$ όπως αναλύεται παρακάτω. Θεωρούμε ως είσοδο του αλγόριθμου το γράφημα $G(V, E, w)$, το οποίο αναπαρίσταται με λίστα γειτνίασης.

1. Υπολογίζουμε το ΕΣΔ T του G με τον αλγόριθμο του Prim σε χρόνο $\mathrm{O}(|E| + |V| \log |V|) = \mathrm{O}(|V|^2)$.
2. Σύμφωνα με το (γ), υπολογίζουμε την ακμή μέγιστου βάρους $e_{uv}^{\max}$ στο μονοπάτι μεταξύ u και v στο T, για όλα τα ζεύγη κορυφών $u, v \in V$, σε χρόνο $\mathrm{O}(|V|^2)$.
3. Υπολογίζουμε το ζεύγος ακμών $(u, v) \notin T$ και $e_{uv}^{\max}$ με ελάχιστη διαφορά βάρους $w(u, v) - w(e_{uv}^{\max})$. Δεδομένων των ακμών $e_{uv}^{\max}$, αυτό υλοποιείται σε χρόνο $\mathrm{O}(|E|) = \mathrm{O}(|V|^2)$ με έλεγχο της διαφοράς βάρους $w(u, v) - w(e_{uv}^{\max})$ για κάθε ακμή $(u, v) \notin T$.
4. Επιστρέφουμε το συνδετικό δέντρο $T \cup \{(u, v)\} \setminus \{e_{uv}^{\max}\}$, το οποίο είναι ένα 2ο-ΕΣΔ του G. □

Άσκηση 7: Bottleneck Spanning Tree (CLRS 23.3)

Ένα Bottleneck Spanning Tree ενός απλού μη κατευθυνόμενου γραφήματος $G(V, E, w)$ με βάρη στις ακμές είναι ένα συνδετικό δέντρο του G στο οποίο το μέγιστο βάρος ακμής είναι ελάχιστο μεταξύ όλων των συνδετικών δέντρων του G. Δηλαδή το Bottleneck Spanning Tree T ελαχιστοποιεί το bottleneck κόστος $c(T) = \max_{e \in T}\{w(e)\}$.

(α) Να δείξετε ότι κάθε Ελάχιστο Συνδετικό Δέντρο του G αποτελεί ένα Bottleneck Spanning Tree. Με βάση το (α), μπορούμε να υπολογίσουμε ένα Bottleneck Spanning Tree του G σε χρόνο όχι μεγαλύτερο από τον χρόνο που χρειάζεται για τον υπολογισμό ενός Ελάχιστου Συνδετικού Δέντρου. Στη συνέχεια, θα δείξουμε ότι ένα Bottleneck Spanning Tree μπορεί να υπολογιστεί σε γραμμικό χρόνο.

(β) Να διατυπώσετε αλγόριθμο *γραμμικού χρόνου* ο οποίος δέχεται ως είσοδο ένα γράφημα $G(V, E, w)$ και έναν φυσικό B, και αποφασίζει αν το G έχει συνδετικό δέντρο με bottleneck κόστος μικρότερο ή ίσο του B.

(γ) Να διατυπώσετε αλγόριθμο *γραμμικού χρόνου* που δέχεται ως είσοδο ένα γράφημα $G(V, E, w)$, και υπολογίζει ένα Bottleneck Spanning Tree του G.

Λύση. (α) Με στόχο να καταλήξουμε σε άτοπο, υποθέτουμε ότι υπάρχει Ελάχιστο Συνδετικό Δέντρο (ΕΣΔ) T με bottleneck κόστος μεγαλύτερο από το αντίστοιχο κόστος ενός Bottleneck Spanning Tree (BST) T_b, δηλ. ισχύει ότι $c(T_b) < c(T)$. Συνεπώς υπάρχει ακμή $e^* \in T$ με βάρος $w(e)$ μεγαλύτερο από το βάρος κάθε κάθε ακμής του T_b. Αφαιρούμε την ακμή e^* από το T, "σπάζοντάς" το έτσι σε δύο συνεκτικές συνιστώσες T_1 και T_2. Έστω e μια ακμή του T_b που διασχίζει την τομή που ορίζεται από τις κορυφές των T_1 και T_2 (τέτοια e υπάρχει γιατί το T_b είναι συνδετικό δέντρο). Αφού $w(e^*) > w(e)$, το $T \cup \{e\} \setminus \{e^*\}$ είναι ένα συνδετικό δέντρο με συνολικό βάρος μικρότερο από το βάρος του T, άτοπο.

(β) Ο αλγόριθμος διαγράφει όλες τις ακμές με βάρος μεγαλύτερο του B, και ελέγχει (με BFS) αν το γράφημα που απομένει είναι συνεκτικό. Αν το γράφημα που απομένει είναι συνεκτικό, το δέντρο του BFS αποτελεί ένα συνδετικό δέντρο με bottleneck κόστος μικρότερο ή ίσο του B. Ο χρόνος εκτέλεσης είναι γραμμικός ως προς το πλήθος ακμών.

(γ) Χωρίς βλάβη της γενικότητας, θεωρούμε ότι όλα τα βάρη των ακμών είναι διαφορετικά (αν δεν είναι, επιτυγχάνουμε το ίδιο αποτέλεσμα ορίζοντας έναν "κανόνα προτίμησης" μεταξύ ακμών ίδιου βάρους). Σε χρόνο $\Theta(m)$, υπολογίζουμε την "median" ακμή e_{med}, η οποία έχει βάρος μικρότερο από τις μισές ακμές του γραφήματος και μεγαλύτερο από τις άλλες μισές. Έστω $w(e_{\mathrm{med}}) = B$ το βάρος της "median" ακμής. Με τον αλγόριθμο του (β), ελέγχουμε σε χρόνο $\Theta(m)$ αν το γράφημα έχει bottleneck spanning tree με κόστος μικρότερο ή ίσο του B.

- Αν ναι, εφαρμόζουμε αναδρομικά τον ίδιο αλγόριθμο στο γράφημα που προκύπτει από τη διαγραφή των ακμών με βάρος μεγαλύτερο του B. Το γράφημα αυτό έχει $m/2$ ακμές και είναι συνεκτικό. Το bottleneck spanning tree για το αρχικό γράφημα είναι το bottleneck spanning tree που θα προκύψει ως απάντηση της αναδρομικής κλήσης.
- Αν όχι, συμπτύσσουμε τις συνεκτικές συνιστώσες που υπολόγισε το BFS, και θεωρούμε το γράφημα με κορυφές αυτές τις συνεκτικές συνιστώσες και ακμές όλες τις μεταξύ τους ακμές στο αρχικό γράφημα (αν υπάρχουν πολλαπλές ακμές μεταξύ κάποιων συνεκτικών συνιστωσών, κρατάμε μόνο αυτή με το ελάχιστο βάρος). Το γράφημα αυτό έχει $m/2$ ακμές το πολύ, αφού όλες οι ακμές με βάρος μικρότερο του B έχουν συμπτυχθεί στις συνεκτικές συνιστώσες, και μπορεί να υπολογισθεί σε χρόνο $\Theta(m)$, με ένα πέρασμα των ακμών. Το bottleneck spanning tree για το αρχικό γράφημα αποτελείται από τις ακμές του bottleneck spanning tree για το γράφημα που προκύπτει από την σύμπτυξη και τις ακμές του BFS-δάσους για κάθε συνεκτική συνιστώσα.

Η αναδρομή σταματά όταν το σύνολο ακμών με βάρος μικρότερο ή ίσο του B αποτελεί ένα συνδετικό δέντρο του γραφήματος. Σε αυτή την περίπτωση, το bottleneck spanning tree έχει bottleneck κόστος ίσο με B.

Η ορθότητα αποδεικνύεται με επαγωγή. Για το χρόνο εκτέλεσης, παρατηρούμε ότι όλες οι ενέργειες εκτός της αναδρομικής κλήσης απαιτούν χρόνο $\Theta(m)$. Συνεπώς, ο χρόνος εκτέλεσης συνολικά δίνεται από την αναδρομική σχέση $T(m) = T(m/2) + \Theta(m)$, $T(1) = \Theta(1)$, η οποία έχει λύση $T(m) = \Theta(m)$. □