# MINIMUM SPANNING TREE AS A PATH-FINDING PROBLEM

Κοντού Άννα

ΑΛΜΑ

Νοέμβριος 2021

# Ελάχιστο επικαλύπτον δένδρο (MST)

**Επικαλύπτον δένδρο** (*spanning tree*) είναι το δένδρο το οποίο περιέχει όλους τους κόμβους του γραφήματος. **Ελάχιστο επικαλύπτον δένδρο** (*minimum spanning tree*) λέγεται το δένδρο που περιέχει τις ελάχιστες ακμές προς όλους τους κόμβους.

# Ελάχιστο επικαλύπτον δένδρο (MST)

- Έστω $G = (V, E)$ ένα συνεκτικό μη κατευθυνόμενο γράφημα, όπου $|V(G) = n|$
- Κάθε ακμή $e = (i, j) \in E(G)$ έχει κόστος $C^0_{ij} = C^0_{ji}$ Διαφορετικά αν $e = (i, j) \notin E(G)$ τότε $C^0_{ij} = C^0_{ji} = \infty$
- Το πρόβλημα του **ελάχιστου επικαλύπτοντος δένδρου (MST)** είναι η εύρεση ενός δένδρου $T = (V, E')$, με $E' \subseteq E$ που να συνδέει όλες τις κορυφές του $G$ έτσι ώστε να ελαχιστοποιεί το συνολικό κόστος των ακμών.

# Το πρόβλημα εύρεσης μονοπατιού (*Path – Finding Problem*)

Το πρόβλημα της **εύρεσης μονοπατιού** είναι ο υπολογισμός του κόστους $C_{ij}^{k}$ για κάθε $1 \leq i, j, k \leq n$ του συντομότερου (χαμηλού κόστους ) μονοπατιού από το $i$ στο $j$ που διέρχεται μόνο από κορυφές του συνόλου $\{1, 2, ..., k\}$ , όπου το κόστος ενός μονοπατιού ορίζεται ως το υψηλότερο κόστος οποιασδήποτε ακμής του μονοπατιού.

# Το πρόβλημα εύρεσης μονοπατιού ($Path - Finding\ Problem$)

Για οποιεσδήποτε κορυφές $i$ και $j$ , το συντομότερο μονοπάτι από το $i$ στο $j$ χωρίς ενδιάμεση κορυφή υψηλότερη από $k$

# Το πρόβλημα εύρεσης μονοπατιού
# (*Path – Finding Problem*)

Για οποιεσδήποτε κορυφές $i$ και $j$ , το συντομότερο μονοπάτι από το $i$ στο $j$ χωρίς ενδιάμεση κορυφή υψηλότερη από $k$

- είτε δεν θα διέρχεται από το $k$ , τότε $C_{ij}^{k} = C_{ij}^{k-1}$

# Το πρόβλημα εύρεσης μονοπατιού (*Path – Finding Problem*)

Για οποιεσδήποτε κορυφές $i$ και $j$ , το συντομότερο μονοπάτι από το $i$ στο $j$ χωρίς ενδιάμεση κορυφή υψηλότερη από $k$

- είτε δεν θα διέρχεται από το $k$ , τότε $C_{ij}^{k} = C_{ij}^{k-1}$
- είτε θα διέρχεται από το $k$ , τότε το κόστος της συντομότερης διαδρομής από το $i$ στο $j$ θα είναι είτε το κόστος της συντομότερης διαδρομής από το $i$ έως το $k$ ή το κόστος της συντομότερης διαδρομής από το $k$ στο $j$, όποιο είναι μεγαλύτερο.

# Το πρόβλημα εύρεσης μονοπατιού (*Path – Finding Problem*)

Για οποιεσδήποτε κορυφές $i$ και $j$ , το συντομότερο μονοπάτι από το $i$ στο $j$ χωρίς ενδιάμεση κορυφή υψηλότερη από $k$

- είτε δεν θα διέρχεται από το $k$ , τότε $C_{ij}^{k} = C_{ij}^{k-1}$
- είτε θα διέρχεται από το $k$ , τότε το κόστος της συντομότερης διαδρομής από το $i$ στο $j$ θα είναι είτε το κόστος της συντομότερης διαδρομής από το $i$ έως το $k$ ή το κόστος της συντομότερης διαδρομής από το $k$ στο $j$, όποιο είναι μεγαλύτερο.

Για τον υπολογισμό του $C_{ij}^{k}$ έχουμε την εξής αναδρομική σχέση :

# Το πρόβλημα εύρεσης μονοπατιού (*Path – Finding Problem*)

Για οποιεσδήποτε κορυφές $i$ και $j$ , το συντομότερο μονοπάτι από το $i$ στο $j$ χωρίς ενδιάμεση κορυφή υψηλότερη από $k$

- είτε δεν θα διέρχεται από το $k$ , τότε $C_{ij}^{k} = C_{ij}^{k-1}$
- είτε θα διέρχεται από το $k$ , τότε το κόστος της συντομότερης διαδρομής από το $i$ στο $j$ θα είναι είτε το κόστος της συντομότερης διαδρομής από το $i$ έως το $k$ ή το κόστος της συντομότερης διαδρομής από το $k$ στο $j$, όποιο είναι μεγαλύτερο.

Για τον υπολογισμό του $C_{ij}^{k}$ έχουμε την εξής αναδρομική σχέση :

$$C_{ij}^{k} = min\{C_{ij}^{k-1}, max\{C_{ik}^{k-1}, C_{kj}^{k-1}\}\}$$

# MST as a Path-Finding Problem

**Θεώρημα**

Μία ακμή $e = (i,j) \in E(G)$ βρίσκεται στο μοναδικό *MST* αν και μόνο αν $C_{ij}^{0} = C_{ij}^{n}$.

# MST as a Path-Finding Problem

**Θεώρημα**

Μία ακμή $e = (i, j) \in E(G)$ βρίσκεται στο μοναδικό *MST* αν και μόνο αν $C^0_{ij} = C^n_{ij}$.

**Απόδειξη.**

($\Rightarrow$) Έστω ότι η ακμή $e = (i, j)$ είναι ακμή του *MST* και έστω ότι $C^0_{ij} \neq C^n_{ij}$. Θεωρούμε τη τομή του γραφήματος $G$ που τη διασχίζει η ακμή $e$. Εφόσον $C^0_{ij} \neq C^n_{ij}$ , υπάρχει κάποιο μονοπάτι από το $i$ στο $j$ το οποίο έχει ακμή μέγιστου κόστους με κόστος $C^0_{ij} < C^n_{ij}$. Άρα, κάθε ακμή σε αυτό το μονοπάτι κοστίζει λιγότερο από $C^0_{ij}$. Αυτή η διαδρομή πρέπει να διασχίζει τη τομή τουλάζιστον μία φορά. Αντικαθιστώντας την ακμή $e$ με οποιαδήποτε άλλη ακμή του μονοπατιού που διασχίζει την τομή , μειώνεται το κόστος του δένρου. Άτοπο. $\square$

# MST as a Path-Finding Problem

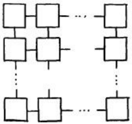
**Θεώρημα**

Έστω $G = (V, E)$ ένα συνεκτικό μη κατευθυνόμενο γράφημα. Μία ακμή $e = (i, j) \in E(G)$ βρίσκεται στο μοναδικό *MST* αν και μόνο αν $C_{ij}^0 = C_{ij}^n$.

**Απόδειξη.**

($\Leftarrow$) Έστω ότι $C_{ij}^0 = C_{ij}^n$ και έστω ότι η ακμή $e = (i, j)$ δεν βρίσκεται στο *MST*. Προσθέτωντας την $e$ στο δένδρο σχηματίζεται ένας κύκλος ο οποίος έχει ακμή μέγιστου κόστους με κόστος μεγαλύτερο από $C_{ij}^0$. Αντικαθιστώντας την ακμή μέγιστου κόστους με την $e$ , προκύπτει ένα δένδρο με μικρότερο κόστος. Άτοπο. $\square$

# Δίκτυο Πλέγματος (Mesh Network)



Ένα ασύρματο δίκτυο πλέγματος (mesh network) είναι ένα δίκτυο που συνδέει μια ομάδα συσκευών μεταξύ τους. Οι συσκευές, που αναφέρονται επίσης ως κόμβοι, συνδέονται με τρόπο που ορισμένοι, αν όχι όλοι οι κόμβοι, έχουν πολλαπλές διαδρομές προς άλλους κόμβους. Με αυτόν τον τρόπο κάθε κόμβος λειτουργεί ως μια πύλη μεταφοράς δεδομένων από και προς τον γειτονικό του.

# Υλοποίηση σε υπολογιστή συνδεδεμένο με πλέγμα

Σε αυτή την ενότητα δίνουμε μια σύντομη περιγραφή ενός αλγορίθμου βημάτων $\mathcal{O}(n)$ για την επίλυση του minimum-cost spanning tree σε έναν υπολογιστή συνδεδεμένο με πλέγμα $n \times n$.

# Υλοποίηση σε υπολογιστή συνδεδεμένο με πλέγμα

Σε αυτή την ενότητα δίνουμε μια σύντομη περιγραφή ενός αλγορίθμου βημάτων $\mathcal{O}(n)$ για την επίλυση του minimum-cost spanning tree σε έναν υπολογιστή συνδεδεμένο με πλέγμα $n \times n$.

Υποθέτουμε ότι το διαγώνιο στοιχείο σε κάθε γραμμή πλέγματος μπορεί να μεταδώσει μια τιμή στα άλλα στοιχεία της σειράς σε ένα μόνο βήμα.

# Υλοποίηση σε υπολογιστή συνδεδεμένο με πλέγμα

Ο αλγόριθμος είναι ο εξής :

- Υποθέτουμε ότι το γράφημα εισόδου δίνεται με τη μορφή ενός πίνακα με στοιχεία τα κόστη των ακμών $C^0$, τα οποία εισέρχονται σειρά προς σειρά από την κορυφή του πλέγματος

Ο αλγόριθμος είναι ο εξής :

- Υποθέτουμε ότι το γράφημα εισόδου δίνεται με τη μορφή ενός πίνακα με στοιχεία τα κόστη των ακμών $C^0$, τα οποία εισέρχονται σειρά προς σειρά από την κορυφή του πλέγματος
- Η γραμμή $i$ του πίνακα τροποποιείται καθώς περνά πάνω από τις γραμμές 1 έως $i-1$ και αποθηκεύεται όταν φτάσει στη γραμμή $i$ του πλέγματος.

# Υλοποίηση σε υπολογιστή συνδεδεμένο με πλέγμα

Ο αλγόριθμος είναι ο εξής :

- Υποθέτουμε ότι το γράφημα εισόδου δίνεται με τη μορφή ενός πίνακα με στοιχεία τα κόστη των ακμών $C^0$, τα οποία εισέρχονται σειρά προς σειρά από την κορυφή του πλέγματος
- Η γραμμή $i$ του πίνακα τροποποιείται καθώς περνά πάνω από τις γραμμές 1 έως $i-1$ και αποθηκεύεται όταν φτάσει στη γραμμή $i$ του πλέγματος.
- Όταν η γραμμή $i$ του πίνακα περνά πάνω από τη γραμμή του πλέγματος $k$ , η τιμή $C_{ik}^{k-1}$ μεταφέρεται δεξιά και αριστερά από το διαγώνιο κελί $(k,k)$.
- Κάθε κελί $(k,j), 1 \leq j \leq n$, γνωρίζει τη τιμή του $C_{kj}^{k-1}$ και υπολογίζει

$$C_{ij}^{k} = min\{C_{ij}^{k-1}, max\{C_{ik}^{k-1}, C_{kj}^{k-1}\}\}$$

και περνάει στην επόμενη γραμμή του πλέγματος.

# Υλοποίηση σε υπολογιστή συνδεδεμένο με πλέγμα

- Αφού φτάσει στη γραμμή $i$ του πλέγματος, η γραμμή $i$ του πίνακα παραμένει εκεί μέχρι κάθε γραμμή $l$, $i \leq l \leq n$, του πίνακα περάσει από πάνω της και στη συνέχεια συνεχίζει να διαδίδεται προς τα κάτω περνώντας πάνω από τις υπόλοιπες σειρές του πίνακα.

# Υλοποίηση σε υπολογιστή συνδεδεμένο με πλέγμα

- Αφού φτάσει στη γραμμή $i$ του πλέγματος, η γραμμή $i$ του πίνακα παραμένει εκεί μέχρι κάθε γραμμή $l$, $i \leq l \leq n$, του πίνακα περάσει από πάνω της και στη συνέχεια συνεχίζει να διαδίδεται προς τα κάτω περνώντας πάνω από τις υπόλοιπες σειρές του πίνακα.
- Ο πίνακας εξόδου $C^n$ εξέρχεται σειρά προς σειρά από το κάτω μέρος του πλέγματος.

# Υλοποίηση σε υπολογιστή συνδεδεμένο με πλέγμα

- Αφού φτάσει στη γραμμή $i$ του πλέγματος, η γραμμή $i$ του πίνακα παραμένει εκεί μέχρι κάθε γραμμή $l$, $i \leq l \leq n$, του πίνακα περάσει από πάνω της και στη συνέχεια συνεχίζει να διαδίδεται προς τα κάτω περνώντας πάνω από τις υπόλοιπες σειρές του πίνακα.
- Ο πίνακας εξόδου $C^n$ εξέρχεται σειρά προς σειρά από το κάτω μέρος του πλέγματος.
- Σύμφωνα με το Θεώρημα 1 , ο πίνακας γειτνίασης του minimum-cost spanning tree μπορεί να κατσκευαστεί συγκρίνοντας τους πίνακες εισόδου και εξόδου.

# Τέλος της παρουσίασης

Ευχαριστώ πολύ για την προσοχή σας!