Αλγόριθμοι και Πολυπλοκότητα

Αρχοντία Γιαννοπούλου
Όλγα Φουρτουνέλλη

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Αλγόριθμοι σε Γραφήματα IV

Άπληστοι Αλγόριθμοι

Κατευθυνόμενα Γραφήματα

Κατευθυνόμενο γράφημα $G = (V, A)$

- V = κορυφές (nodes) ή κόμβοι (nodes)
- A = διατεταγμένα ζευγάρια κορυφών (edges).
  Η ακμή (u, v) κατευθύνεται **από** την κορυφή u **στην** κορυφή v.

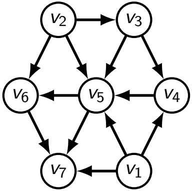
Δοσμένης μίας κορυφής v, καλούμε

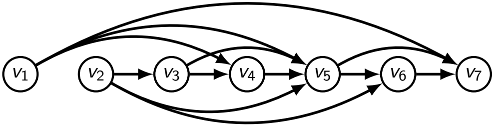
- εσώβαθμο indegree(v) της v το πλήθος των ακμών που έχουν κατεύθυνση προς την v και
- εξώβαθμο outdegree(v) το πλήθος των ακμών που ξεκινάνε από την v.

Τότε

$$\sum_{v \in V} \text{indegree}(v) = \sum_{v \in V} \text{outdegree}(v).$$

Ένα κατευθυνόμενο γράφημα





Αναζήτηση κατευθυνόμενων γραφημάτων

Κατευθυνόμενη προσπελασιμότητα.
Δεδομένης μίας κορυφής s, βρείτε όλες τις κορυφές που είναι προσπελάσιμες από την s.

Πρόβλημα συντομότερης κατευθυνόμενης διαδρομής $s \to t$.
Δεδομένων δύο κορυφών s και t, ποιό είναι το μήκος της συντομότερης διαδρομής μεταξύ της s και της t;

Αναζήτηση κατευθυνόμενων γραφημάτων

Διάσχιση Γραφήματος.
Ο αλγόριθμος BFS επεκτείνεται με ανάλογο τρόπο σε κατευθυνόμενα γραφήματα ακολουθώντας την φορά των ακμών, με την ίδια πολυπλοκότητα $\mathcal{O}(m+n)$, όπου $m =$ πλήθος κατευθυνόμενων ακμών.

Σαρωτής παγκόσμιου ιστού (web crawler).
Ξεκινήστε από την ιστοσελίδα s. Βρείτε όλες τις ιστοσελίδες που είναι προσπελάσιμες από την s, είτε άμεσα είτε έμμεσα.

Ισχυρή συνεκτικότητα

Οι κορυφές u και v είναι αμοιβαία προσπελάσιμες αν υπάρχει μια διαδρομή από την u στην v και μία διαδρομή από την v στην u (όχι κατ΄ ανάγκη μεταξύ τους διακεκριμμένες).

Ισχυρή συνεκτικότητα

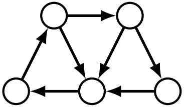
Οι κορυφές u και v είναι αμοιβαία προσπελάσιμες αν υπάρχει μια διαδρομή από την u στην v και μία διαδρομή από την v στην u (όχι κατ΄ ανάγκη μεταξύ τους διακεκριμμένες).

Ένα κατευθυνόμενο γράφημα καλείται ισχυρά συνεκτικό αν κάθε ζευγάρι κορυφών του είναι αμοιβαία προσπελάσιμες.

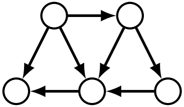
Ισχυρή συνεκτικότητα

Οι κορυφές u και v είναι αμοιβαία προσπελάσιμες αν υπάρχει μια διαδρομή από την u στην v και μία διαδρομή από την v στην u (όχι κατ' ανάγκη μεταξύ τους διακεκριμμένες).

Ένα κατευθυνόμενο γράφημα καλείται ισχυρά συνεκτικό αν κάθε ζευγάρι κορυφών του είναι αμοιβαία προσπελάσιμες.



Ισχυρά συνεκτικό

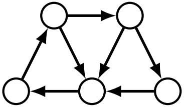


Όχι ισχυρά συνεκτικό

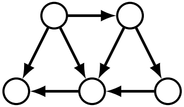
Ισχυρή συνεκτικότητα

Οι κορυφές u και v είναι αμοιβαία προσπελάσιμες αν υπάρχει μια διαδρομή από την u στην v και μία διαδρομή από την v στην u (όχι κατ΄ ανάγκη μεταξύ τους διακεκριμμένες).

Ένα κατευθυνόμενο γράφημα καλείται ισχυρά συνεκτικό αν κάθε ζευγάρι κορυφών του είναι αμοιβαία προσπελάσιμες.



Ισχυρά συνεκτικό



Όχι ισχυρά συνεκτικό

Λήμμα Έστω ένα γράφημα G και μία κορυφή του s. Το G είναι ισχυρά συνεκτικό αν και μόνο αν κάθε κορυφή του είναι προσπελάσιμη από την s και η s είναι προσπελάσιμη από κάθε κορυφή του.

Αλγόριθμος ισχυρής συνεκτικότητας

Θεώρημα. Μπορεί να αποφασιστεί αν το G είναι ένα ισχυρά συνεκτικό γράφημα σε χρόνο $\mathcal{O}(m+n)$.

Αλγόριθμος ισχυρής συνεκτικότητας

Θεώρημα. Μπορεί να αποφασιστεί αν το G είναι ένα ισχυρά συνεκτικό γράφημα σε χρόνο $\mathcal{O}(m+n)$.

- Διάλεξε οποιαδήποτε κορυφή s.

Αλγόριθμος ισχυρής συνεκτικότητας

Θεώρημα. Μπορεί να αποφασιστεί αν το G είναι ένα ισχυρά συνεκτικό γράφημα σε χρόνο $\mathcal{O}(m+n)$.

- Διάλεξε οποιαδήποτε κορυφή s.
- Τρέξε στο G τον αλγόριθμο BFS με αφετηρία την s. (Αν δεν είναι όλες οι κορυφές του G προσπελάσιμες, σταμάτα. Το G δεν είναι ισχυρά συνεκτικό).

Αλγόριθμος ισχυρής συνεκτικότητας

Θεώρημα. Μπορεί να αποφασιστεί αν το G είναι ένα ισχυρά συνεκτικό γράφημα σε χρόνο $\mathcal{O}(m+n)$.

- Διάλεξε οποιαδήποτε κορυφή s.
- Τρέξε στο G τον αλγόριθμο BFS με αφετηρία την s. (Αν δεν είναι όλες οι κορυφές του G προσπελάσιμες, σταμάτα. Το G δεν είναι ισχυρά συνεκτικό).
- Αντίστρεψε την φορά όλων των ακμών στο G και πάρε το γράφημα $G^{\alpha\nu\tau}$.

Αλγόριθμος ισχυρής συνεκτικότητας

Θεώρημα. Μπορεί να αποφασιστεί αν το G είναι ένα ισχυρά συνεκτικό γράφημα σε χρόνο $\mathcal{O}(m+n)$.

- Διάλεξε οποιαδήποτε κορυφή s.
- Τρέξε στο G τον αλγόριθμο BFS με αφετηρία την s. (Αν δεν είναι όλες οι κορυφές του G προσπελάσιμες, σταμάτα. Το G δεν είναι ισχυρά συνεκτικό).
- Αντίστρεψε την φορά όλων των ακμών στο G και πάρε το γράφημα $G^{\alpha\nu\tau}$.
- Τρέξε στο $G^{\alpha\nu\tau}$ τον αλγόριθμο BFS με αφετηρία την s. (Αν δεν είναι όλες οι κορυφές του $G^{\alpha\nu\tau}$ προσπελάσιμες, σταμάτα. Το G δεν είναι ισχυρά συνεκτικό).

Αλγόριθμος ισχυρής συνεκτικότητας

Θεώρημα. Μπορεί να αποφασιστεί αν το G είναι ένα ισχυρά συνεκτικό γράφημα σε χρόνο $\mathcal{O}(m+n)$.

- Διάλεξε οποιαδήποτε κορυφή s.
- Τρέξε στο G τον αλγόριθμο BFS με αφετηρία την s. (Αν δεν είναι όλες οι κορυφές του G προσπελάσιμες, σταμάτα. Το G δεν είναι ισχυρά συνεκτικό).
- Αντίστρεψε την φορά όλων των ακμών στο G και πάρε το γράφημα $G^{\alpha\nu\tau}$.
- Τρέξε στο $G^{\alpha\nu\tau}$ τον αλγόριθμο BFS με αφετηρία την s. (Αν δεν είναι όλες οι κορυφές του $G^{\alpha\nu\tau}$ προσπελάσιμες, σταμάτα. Το G δεν είναι ισχυρά συνεκτικό).
- Επίστρεψε ``Αληθές'' αν και οι δύο κλήσεις του BFS προσπέλασαν όλες τις κορυφές του G και $G^{\alpha\nu\tau}$.

Αλγόριθμος ισχυρής συνεκτικότητας

Θεώρημα. Μπορεί να αποφασιστεί αν το G είναι ένα ισχυρά συνεκτικό γράφημα σε χρόνο $\mathcal{O}(m+n)$.

- Διάλεξε οποιαδήποτε κορυφή s.
- Τρέξε στο G τον αλγόριθμο BFS με αφετηρία την s. (Αν δεν είναι όλες οι κορυφές του G προσπελάσιμες, σταμάτα. Το G δεν είναι ισχυρά συνεκτικό).
- Αντίστρεψε την φορά όλων των ακμών στο G και πάρε το γράφημα $G^{\alpha\nu\tau}$.
- Τρέξε στο $G^{\alpha\nu\tau}$ τον αλγόριθμο BFS με αφετηρία την s. (Αν δεν είναι όλες οι κορυφές του $G^{\alpha\nu\tau}$ προσπελάσιμες, σταμάτα. Το G δεν είναι ισχυρά συνεκτικό).
- Επίστρεψε "Αληθές" αν και οι δύο κλήσεις του BFS προσπέλασαν όλες τις κορυφές του G και $G^{\alpha\nu\tau}$.

Ο αλγόριθμος είναι σωστός λόγω του προηγούμενου λήμματος.

Συντομότερη Διαδρομή

Δεδομένα:

- Κατευθυνόμενο γράφημα $G = (V, A)$.
- Αφετηρία s, προορισμός t.
- $\ell_e > 0$ πραγματικός αριθμός = μήκος της ακμής e.

Συντομότερη Διαδρομή

Δεδομένα:

- Κατευθυνόμενο γράφημα $G = (V, A)$.
- Αφετηρία s, προορισμός t.
- $\ell_e > 0$ πραγματικός αριθμός $=$ μήκος της ακμής e.

Κόστος διαδρομής: άθροισμα μηκών των ακμών της διαδρομής.

Συντομότερη Διαδρομή

Δεδομένα:

- Κατευθυνόμενο γράφημα $G = (V, A)$.
- Αφετηρία s, προορισμός t.
- $\ell_e > 0$ πραγματικός αριθμός $=$ μήκος της ακμής e.

Κόστος διαδρομής: άθροισμα μηκών των ακμών της διαδρομής.

Πρόβλημα συντομότερης διαδρομής: Να βρούμε τη συντομότερη σε μήκος κατευθυνόμενη διαδρομή από την κορυφή s στην κορυφή t.

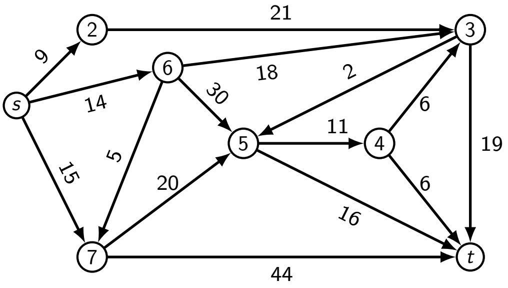
Συντομότερη Διαδρομή

Δεδομένα:

- Κατευθυνόμενο γράφημα $G = (V, A)$.
- Αφετηρία s, προορισμός t.
- $\ell_e > 0$ πραγματικός αριθμός = μήκος της ακμής e.

Κόστος διαδρομής: άθροισμα μηκών των ακμών της διαδρομής.

Πρόβλημα συντομότερης διαδρομής: Να βρούμε τη συντομότερη σε μήκος κατευθυνόμενη διαδρομή από την κορυφή s στην κορυφή t.



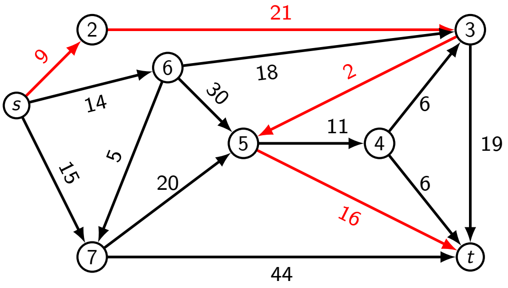
Συντομότερη Διαδρομή

Δεδομένα:

- Κατευθυνόμενο γράφημα $G = (V, A)$.
- Αφετηρία s, προορισμός t.
- $\ell_e > 0$ πραγματικός αριθμός = μήκος της ακμής e.

Κόστος διαδρομής: άθροισμα μηκών των ακμών της διαδρομής.

Πρόβλημα συντομότερης διαδρομής: Να βρούμε τη συντομότερη σε μήκος κατευθυνόμενη διαδρομή από την κορυφή s στην κορυφή t.



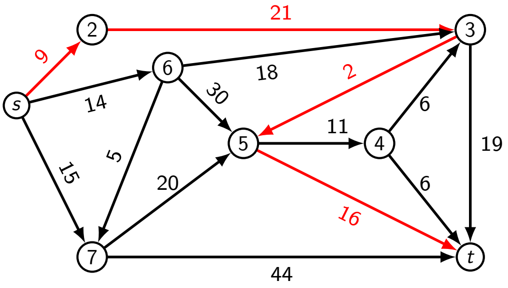
Συντομότερη Διαδρομή

Δεδομένα:

- Κατευθυνόμενο γράφημα $G = (V, A)$.
- Αφετηρία s, προορισμός t.
- $\ell_e > 0$ πραγματικός αριθμός = μήκος της ακμής e.

Κόστος διαδρομής: άθροισμα μηκών των ακμών της διαδρομής.

Πρόβλημα συντομότερης διαδρομής: Να βρούμε τη συντομότερη σε μήκος κατευθυνόμενη διαδρομή από την κορυφή s στην κορυφή t.



Μήκος διαδρομής
$(s \to 2 \to 3 \to 5 \to t) =$
$9 + 21 + 2 + 16 = 48$

Αλγόριθμος Dijkstra

- Διατήρησε ένα σύνολο S εξερευνημένων κορυφών για το οποίο έχουμε προσδιορίσει την απόσταση $d(u)$ της συντομότερης διαδρομής από την s στην $u \in S$.

Αλγόριθμος Dijkstra

- Διατήρησε ένα σύνολο S εξερευνημένων κορυφών για το οποίο έχουμε προσδιορίσει την απόσταση $d(u)$ της συντομότερης διαδρομής από την s στην $u \in S$.

- Αρχικοποίησε $S = \{s\}$, $d(s) = 0$.

Αλγόριθμος Dijkstra

- Διατήρησε ένα σύνολο S εξερευνημένων κορυφών για το οποίο έχουμε προσδιορίσει την απόσταση $d(u)$ της συντομότερης διαδρομής από την s στην $u \in S$.

- Αρχικοποίησε $S = \{s\}$, $d(s) = 0$.

- Επαναληπτικά, διάλεξε την κορυφή v που δεν έχει εξερευνηθεί και ελαχιστοποιεί την τρέχουσα ελάχιστη απόσταση $\pi(v) = d(u) + \ell_{uv}$ από τις διαδρομές που περιέχουν την συντομότερη διαδρομή προς u στο S, ακολουθούμενη από ακμή $e = (u, v)$.

Αλγόριθμος Dijkstra

- Διατήρησε ένα σύνολο S εξερευνημένων κορυφών για το οποίο έχουμε προσδιορίσει την απόσταση $d(u)$ της συντομότερης διαδρομής από την s στην $u \in S$.

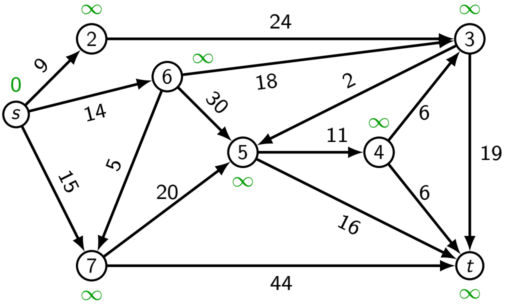
- Αρχικοποίησε $S = \{s\}$, $d(s) = 0$.

- Επαναληπτικά, διάλεξε την κορυφή v που δεν έχει εξερευνηθεί και ελαχιστοποιεί την τρέχουσα ελάχιστη απόσταση $\pi(v) = d(u) + \ell_{uv}$ από τις διαδρομές που περιέχουν την συντομότερη διαδρομή προς u στο S, ακολουθούμενη από ακμή $e = (u, v)$.

- Θέσε $d(v) = \pi(v)$ και πρόσθεσε την v στο S.

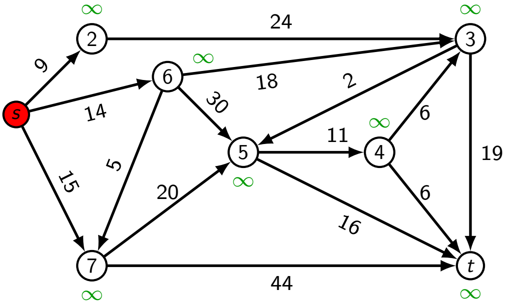
Παράδειγμα

$$S = \{\}$$
$$PQ = \{s, 2, 3, 4, 5, 6, 7, t\}$$



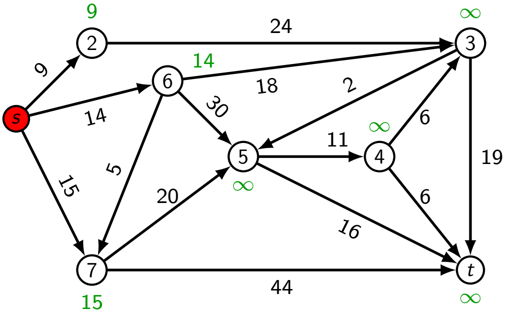
Παράδειγμα

$$S = \{s\}$$
$$PQ = \{2, 3, 4, 5, 6, 7, t\}$$



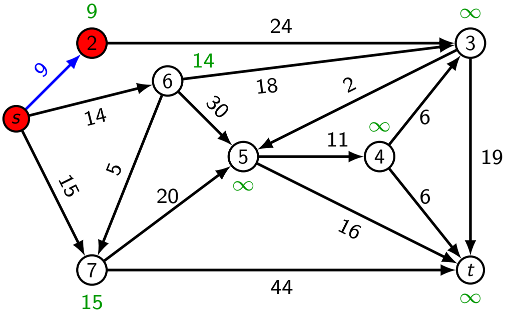
Παράδειγμα

$$\begin{aligned} S &= \{s\} \\ PQ &= \{2, 3, 4, 5, 6, 7, t\} \end{aligned}$$



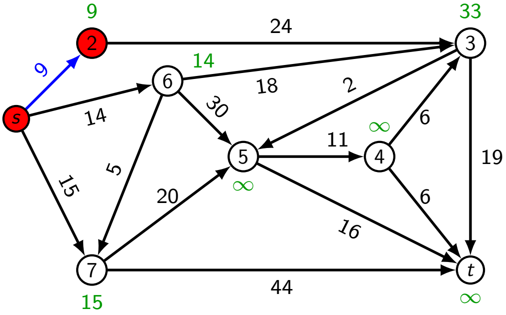
Παράδειγμα

$$\begin{aligned} S &= \{s, 2\} \\ PQ &= \{3, 4, 5, 6, 7, t\} \end{aligned}$$



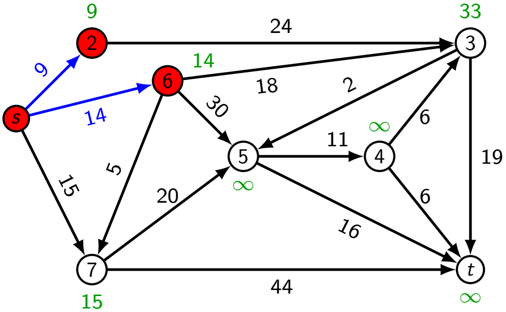
Παράδειγμα

$$S = \{s, 2\}$$
$$PQ = \{3, 4, 5, 6, 7, t\}$$



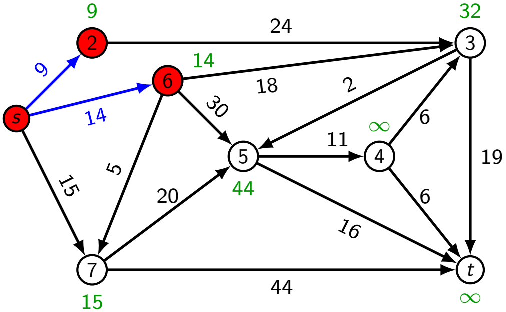
Παράδειγμα

$$S = \{s, 2, 6\}$$
$$PQ = \{3, 4, 5, 7, t\}$$



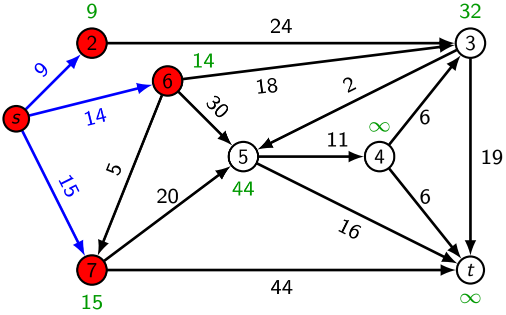
Παράδειγμα

$$S = \{s, 2, 6\}$$
$$PQ = \{3, 4, 5, 7, t\}$$



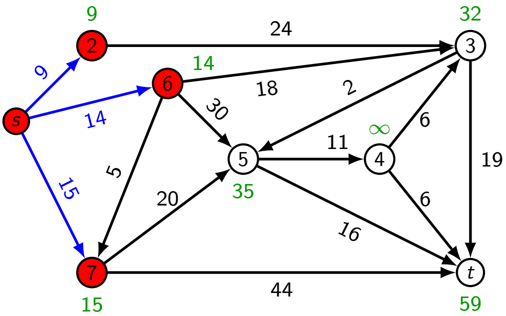
Παράδειγμα

$$S = \{s, 2, 6, 7\}$$
$$PQ = \{3, 4, 5, t\}$$



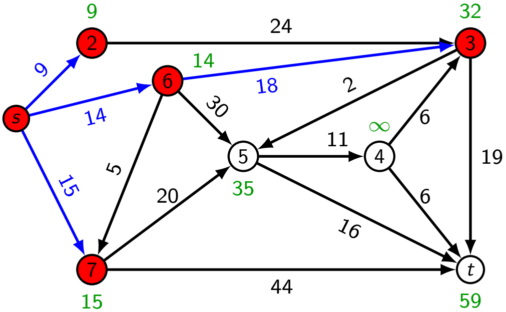
Παράδειγμα

$$\begin{aligned} S &= \{s, 2, 6, 7\} \\ PQ &= \{3, 4, 5, t\} \end{aligned}$$



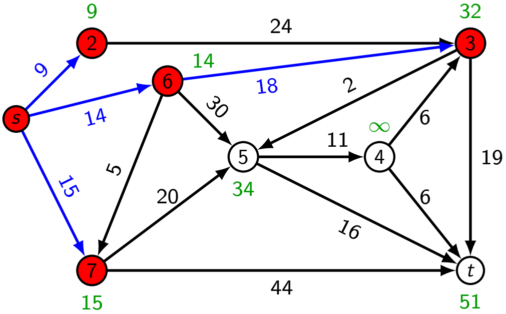
Παράδειγμα

$$
\begin{aligned}
S &= \{s, 2, 6, 7, 3\} \\
PQ &= \{4, 5, t\}
\end{aligned}
$$



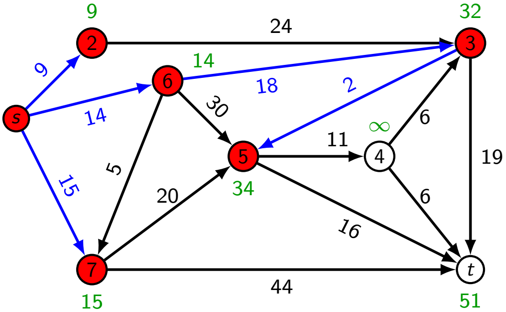
Παράδειγμα

$$S = \{s, 2, 6, 7, 3\}$$
$$PQ = \{4, 5, t\}$$



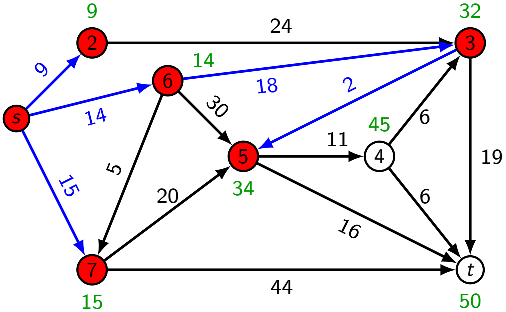
Παράδειγμα

$$\begin{aligned} S &= \{s, 2, 6, 7, 3, 5\} \\ PQ &= \{4, t\} \end{aligned}$$



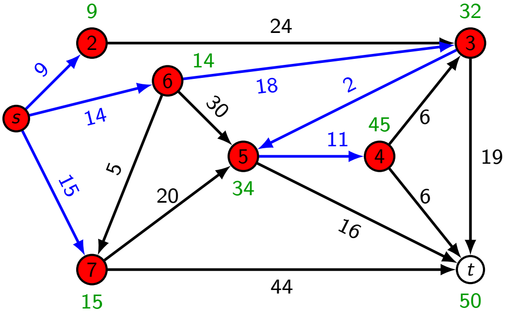
Παράδειγμα

$$
\begin{aligned}
S &= \{s, 2, 6, 7, 3, 5\} \\
PQ &= \{4, t\}
\end{aligned}
$$



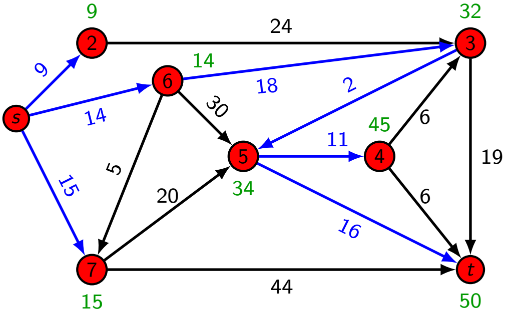
Παράδειγμα

$$\begin{aligned} S &= \{s, 2, 6, 7, 3, 5, 4\} \\ PQ &= \{t\} \end{aligned}$$



Παράδειγμα

$$S = \{s, 2, 6, 7, 3, 5, 4, t\}$$
$$PQ = \{\}$$



Ορθότητα

Αναλλοίωτη συνθήκη: Για κάθε κορυφή $u \in S$, $d(u) =$ μήκος συντομότερης διαδρομής $s \to u$.

Ορθότητα

Αναλλοίωτη συνθήκη: Για κάθε κορυφή $u \in S$, $d(u) =$ μήκος συντομότερης διαδρομής $s \to u$.

Απόδειξη με επαγωγή στο S.

Ορθότητα

Αναλλοίωτη συνθήκη: Για κάθε κορυφή $u \in S$, $d(u) =$ μήκος συντομότερης διαδρομής $s \to u$.

Απόδειξη με επαγωγή στο S.

Βάση: για $|S| = 1$, $S = \{s\}$, $d(s) = 0$ ισχύει τετριμμένα.

Ορθότητα

Αναλλοίωτη συνθήκη: Για κάθε κορυφή $u \in S$, $d(u) =$ μήκος συντομότερης διαδρομής $s \to u$.

Απόδειξη με επαγωγή στο S.

Βάση: για $|S| = 1$, $S = \{s\}$, $d(s) = 0$ ισχύει τετριμμένα.

Ε.Υ.: Υποθέτουμε ότι ισχύει για $|S| = k \geq 1$.

Ορθότητα

Αναλλοίωτη συνθήκη: Για κάθε κορυφή $u \in S$, $d(u) =$ μήκος συντομότερης διαδρομής $s \to u$.

Απόδειξη με επαγωγή στο S.

Βάση: για $|S| = 1$, $S = \{s\}$, $d(s) = 0$ ισχύει τετριμμένα.

Ε.Υ.: Υποθέτουμε ότι ισχύει για $|S| = k \geq 1$.

Έστω v η κορυφή που προστίθεται στο S στο επόμενο βήμα και έστω (u, v) η ακμή μέσω της οποίας έχει αποκτήσει το μικρότερο $\pi(v)$. Με άλλα λόγια, η συντομότερη διαδρομή $s \to u$ μαζί με την ακμή (u, v) μου δίνει ότι $\pi(v) = d(u) + \ell_{uv}$.

Ορθότητα

Αναλλοίωτη συνθήκη: Για κάθε κορυφή $u \in S$, $d(u) =$ μήκος συντομότερης διαδρομής $s \to u$.

Απόδειξη με επαγωγή στο S.

Βάση: για $|S| = 1$, $S = \{s\}$, $d(s) = 0$ ισχύει τετριμμένα.

Ε.Υ.: Υποθέτουμε ότι ισχύει για $|S| = k \geq 1$.

Έστω v η κορυφή που προστίθεται στο S στο επόμενο βήμα και έστω (u, v) η ακμή μέσω της οποίας έχει αποκτήσει το μικρότερο $\pi(v)$. Με άλλα λόγια, η συντομότερη διαδρομή $s \to u$ μαζί με την ακμή (u, v) μου δίνει ότι $\pi(v) = d(u) + \ell_{uv}$.

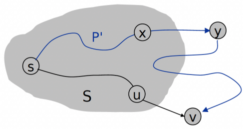
Έστω οποιαδήποτε διαδρομή $s \to v$. Θα δείξουμε ότι δεν είναι συντομότερη από $\pi(v)$.

Ορθότητα

Έστω P μία διαδρομή $s \to v$ και έστω (x, y) η πρώτη ακμή της που καταλήγει εκτός του S. Καλούμε P' την υποδιαδρομή της $s \to x$.

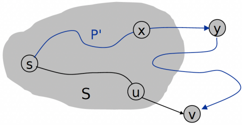
Ορθότητα

Έστω P μία διαδρομή $s \to v$ και έστω (x, y) η πρώτη ακμή της που καταλήγει εκτός του S. Καλούμε P' την υποδιαδρομή της $s \to x$.



Ορθότητα

Έστω P μία διαδρομή $s \to v$ και έστω (x, y) η πρώτη ακμή της που καταλήγει εκτός του S. Καλούμε P' την υποδιαδρομή της $s \to x$.

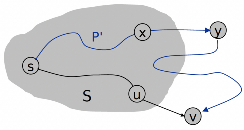


Τότε

$$\ell(P) \geq$$

Ορθότητα

Έστω P μία διαδρομή $s \rightarrow v$ και έστω (x, y) η πρώτη ακμή της που καταλήγει εκτός του S. Καλούμε P' την υποδιαδρομή της $s \rightarrow x$.



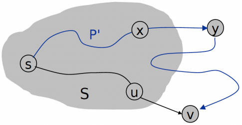
Τότε

$$\ell(P) \geq \ell(P') + \ell_{xy} \qquad .$$

Γιατί η $P' + (x, y)$ είναι υποδιαδρομή της P.

Ορθότητα

Έστω P μία διαδρομή $s \to v$ και έστω (x, y) η πρώτη ακμή της που καταλήγει εκτός του S. Καλούμε P' την υποδιαδρομή της $s \to x$.



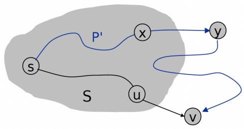
Τότε

$$\ell(P) \geq \ell(P') + \ell_{xy} \geq d(x) + \ell_{xy} \qquad .$$

Γιατί η $P' + (x, y)$ είναι υποδιαδρομή της P.
Γιατί $d(x)$ είναι το μήκος της συντομότερης διαδρομής από την s στην x.

Ορθότητα

Έστω P μία διαδρομή $s \to v$ και έστω (x, y) η πρώτη ακμή της που καταλήγει εκτός του S. Καλούμε P' την υποδιαδρομή της $s \to x$.



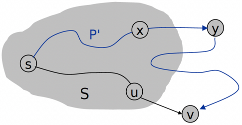
Τότε

$$\ell(P) \geq \ell(P') + \ell_{xy} \geq d(x) + \ell_{xy} \geq \pi(y) \qquad .$$

Γιατί η $P' + (x, y)$ είναι υποδιαδρομή της P.
Γιατί $d(x)$ είναι το μήκος της συντομότερης διαδρομής από την s στην x.
Από τον ορισμό του $\pi(y)$.

Ορθότητα

Έστω P μία διαδρομή $s \to v$ και έστω (x, y) η πρώτη ακμή της που καταλήγει εκτός του S. Καλούμε P' την υποδιαδρομή της $s \to x$.



Τότε

$$\ell(P) \geq \ell(P') + \ell_{xy} \geq d(x) + \ell_{xy} \geq \pi(y) \geq \pi(v).$$

Γιατί η $P' + (x, y)$ είναι υποδιαδρομή της P.
Γιατί $d(x)$ είναι το μήκος της συντομότερης διαδρομής από την s στην x.
Από τον ορισμό του $\pi(y)$.
Γιατί ο αλγόριθμος επιλέγει την κορυφή με την μικρότερη τιμή $\pi()$ και επέλεξε την v.

Πολυπλοκότητα

- Για κάθε ανεξερεύνητη κορυφή v, διατηρούμε $\pi(v) = \min_{u \in S}\{d(u) + \ell_{uv}\}$, όπου $\ell_{uv} = \infty$ αν δεν υπάρχει η ακμή (u, v).

Πολυπλοκότητα

- Για κάθε ανεξερεύνητη κορυφή v, διατηρούμε $\pi(v) = \min_{u \in S}\{d(u) + \ell_{uv}\}$, όπου $\ell_{uv} = \infty$ αν δεν υπάρχει η ακμή (u, v).

- Επόμενη κορυφή προς εξερεύνηση = κορυφή με ελάχιστο $\pi(v)$. Όταν η v εντάσσεται στο S, για κάθε προσκείμενη ακμή $e = (v, w)$, ενημερώνουμε την τιμή $\pi(w) := min\{\pi(w), \pi(v) + \ell_e\}$. Συνολικά $\mathcal{O}(m)$ βήματα.

Πολυπλοκότητα

- Για κάθε ανεξερεύνητη κορυφή v, διατηρούμε $\pi(v) = \min_{u \in S}\{d(u) + \ell_{uv}\}$, όπου $\ell_{uv} = \infty$ αν δεν υπάρχει η ακμή (u, v).

- Επόμενη κορυφή προς εξερεύνηση = κορυφή με ελάχιστο $\pi(v)$. Όταν η v εντάσσεται στο S, για κάθε προσκείμενη ακμή $e = (v, w)$, ενημερώνουμε την τιμή $\pi(w) := min\{\pi(w), \pi(v) + \ell_e\}$. Συνολικά $\mathcal{O}(m)$ βήματα.

- **Αποδοτική υλοποίηση**: Διατηρήστε ουρά προτεραιότητας PQ από ανεξερεύνητες κορυφές με προτεραιότητα $\pi(v)$.

Πολυπλοκότητα

- Για κάθε ανεξερεύνητη κορυφή v, διατηρούμε $\pi(v) = \min_{u \in S}\{d(u) + \ell_{uv}\}$, όπου $\ell_{uv} = \infty$ αν δεν υπάρχει η ακμή (u, v).
- Επόμενη κορυφή προς εξερεύνηση = κορυφή με ελάχιστο $\pi(v)$. Όταν η v εντάσσεται στο S, για κάθε προσκείμενη ακμή $e = (v, w)$, ενημερώνουμε την τιμή $\pi(w) := min\{\pi(w), \pi(v) + \ell_e\}$. Συνολικά $\mathcal{O}(m)$ βήματα.
- **Αποδοτική υλοποίηση**: Διατηρήστε ουρά προτεραιότητας PQ από ανεξερεύνητες κορυφές με προτεραιότητα $\pi(v)$.
- Σύνολο: $\mathcal{O}(m \log n)$ βήματα.

Ελάχιστο Επικαλύπτον Δέντρο

Ελάχιστον Επικαλύπτον/Συνδετικό/Γεννητικό Δέντρο (ΕΕΔ)

Ελάχιστο Επικαλύπτον Δέντρο

Ελάχιστον Επικαλύπτον/Συνδετικό/Γεννητικό Δέντρο (ΕΕΔ)

Δεδομένα

- μη-κατευθυνόμενο συνεκτικό γράφημα $G = (V, E)$
- πραγματικά κόστη ακμών c_e

Ελάχιστο Επικαλύπτον Δέντρο

Ελάχιστον Επικαλύπτον/Συνδετικό/Γεννητικό Δέντρο (ΕΕΔ)

Δεδομένα

- μη-κατευθυνόμενο συνεκτικό γράφημα $G = (V, E)$
- πραγματικά κόστη ακμών c_e

Ένα επικαλύπτον δέντρο είναι ένα υποσύνολο των ακμών $T \subseteq E$ τ.ώ. το T να είναι δέντρο, επικαλύπτον (επικαλύπτει / περιέχει όλες τις κορυφές του γραφήματος V).

Ελάχιστο Επικαλύπτον Δέντρο

Ελάχιστον Επικαλύπτον/Συνδετικό/Γεννητικό Δέντρο (ΕΕΔ)

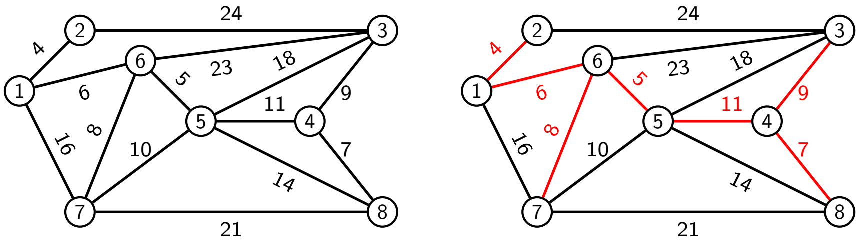
Δεδομένα

- μη-κατευθυνόμενο συνεκτικό γράφημα $G = (V, E)$
- πραγματικά κόστη ακμών c_e

Ένα επικαλύπτον δέντρο είναι ένα υποσύνολο των ακμών $T \subseteq E$ τ.ώ. το T να είναι δέντρο, επικαλύπτον (επικαλύπτει / περιέχει όλες τις κορυφές του γραφήματος V).

Πρόβλημα: Εύρεση ενός ελάχιστου επικαλύπτοντος δέντρου (του οποίου το άθροισμα των κοστών των ακμών $\sum_{e \in T} c_e$ είναι ελάχιστο).

Παράδειγμα



Θεώρημα του Cayley. Υπάρχουν n^{n-2} επικαλύπτοντα δέντρα στο πλήρες γράφημα (κλίκα) K_n.

Θεώρημα του Cayley. Υπάρχουν n^{n-2} επικαλύπτοντα δέντρα στο πλήρες γράφημα (κλίκα) K_n.

Αρα το πρόβλημα δεν λύνεται με αλγόριθμο ωμή βίας.

Θεώρημα του Cayley. Υπάρχουν n^{n-2} επικαλύπτοντα δέντρα στο πλήρες γράφημα (κλίκα) K_n.

Αρα το πρόβλημα δεν λύνεται με αλγόριθμο ωμή βίας.

Ας θυμηθούμε ότι τα δέντρα με n κορυφές έχουν $n-1$ ακμές.

Εφαρμογές

Το ΕΕΔ είναι ένα θεμελιώδες πρόβλημα με ποικίλες εφαρμογές.

- Σχεδιασμός δικτύων: τηλέφωνο, ηλεκτρικό, ύδρευσης, καλωδιακή, δίκτυα υπολογιστών, οδικά δίκτυα.
- Προσεγγιστικοί αλγόριθμοι για NP-δύσκολα προβλήματα: Πρόβλημα του πλανόδιου πωλητή, δέντρο Steiner
- Διαδρομές μεγίστου εμποδίου (max bottleneck paths)
- LDPC κωδικοί για επιδιόρθωση λαθών
- Καταχώρηση εικόνας με εντροπία Renyi
- Εκμάθηση χαρακτηριστικών που προεξέχουν για αναγνώριση προσώπου σε πραγματικό χρόνο
- Μείωση αποθηκευμένης πληροφορίας στην διάταξη αμινοξέων σε μια πρωτεΐνη
- Μοντελοποίηση της τοπικότητας στην αλληλεπίδραση σωματιδίων σε ροές υγρών με αναταράξεις
- Αυτορυθμιζόμενο πρωτόκολλο για γεφύρωση ethernet για την αποφυγή κύκλων σε δίκτυο
- Ανάλυση συστάδων.

Αλγόριθμοι (άπληστοι)

- **Αλγόριθμος του Kruskal:**

- **Αλγόριθμος Αντίστροφης Διαγραφής:**

- **Αλγόριθμος του Prim:**

Αλγόριθμοι (άπληστοι)

- **Αλγόριθμος του Kruskal:** Ξεκινάμε με κενό $T = \emptyset$. Θεωρούμε τις ακμές σε αύξουσα σειρά με βάση το κόστος. Εισάγουμε μια ακμή e στο T εκτός εάν δημιουργείται κύκλος.
  Τερματισμός: όταν όλες οι κορυφές "καλύπτονται" (ανήκουν στο T).

- **Αλγόριθμος Αντίστροφης Διαγραφής:**

- **Αλγόριθμος του Prim:**

Αλγόριθμοι (άπληστοι)

- **Αλγόριθμος του Kruskal:** Ξεκινάμε με κενό $T = \emptyset$. Θεωρούμε τις ακμές σε αύξουσα σειρά με βάση το κόστος. Εισάγουμε μια ακμή e στο T εκτός εάν δημιουργείται κύκλος.
  Τερματισμός: όταν όλες οι κορυφές "καλύπτονται" (ανήκουν στο T).
- **Αλγόριθμος Αντίστροφης Διαγραφής:** Ξεκινάμε με $T = E$. Θεωρούμε τις ακμές σε φθίνουσα σειρά με βάση το κόστος. Διαγράφουμε την ακμή e από το T εκτός εάν το T γίνεται μη συνεκτικό.
  Τερματίζει όταν απαλειφθούν όλοι οι κύκλοι.
- **Αλγόριθμος του Prim:**

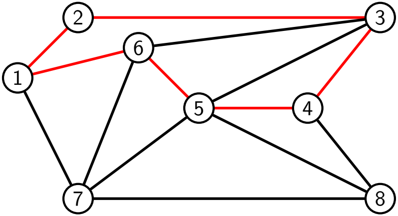
Αλγόριθμοι (άπληστοι)

- **Αλγόριθμος του Kruskal:** Ξεκινάμε με κενό $T = \emptyset$. Θεωρούμε τις ακμές σε αύξουσα σειρά με βάση το κόστος. Εισάγουμε μια ακμή e στο T εκτός εάν δημιουργείται κύκλος.
  Τερματισμός: όταν όλες οι κορυφές "καλύπτονται" (ανήκουν στο T).

- **Αλγόριθμος Αντίστροφης Διαγραφής:** Ξεκινάμε με $T = E$. Θεωρούμε τις ακμές σε φθίνουσα σειρά με βάση το κόστος. Διαγράφουμε την ακμή e από το T εκτός εάν το T γίνεται μη συνεκτικό.
  Τερματίζει όταν απαλειφθούν όλοι οι κύκλοι.

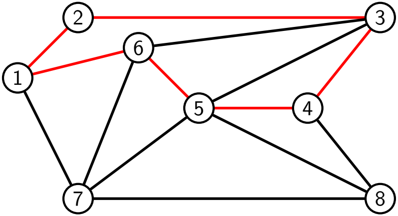
- **Αλγόριθμος του Prim:** Ξεκινάμε με μια αρχική κορυφή s: με απληστία μεγαλώνουμε δέντρο T από την s προς τα έξω. Σε κάθε βήμα, προσθέτουμε στο T την ακμή e με το μικρότερο κόστος που έχει ακριβώς ένα της άκρο στο T (αποφυγή κύκλου).
  Τερματίζει όταν καλύψει όλες τις κορυφές ("επικαλύπτον" δένδρο).

Κύκλοι και αποκοπές



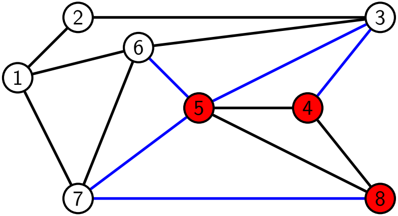
Κύκλος: 1-2, 2-3, 3-4, 4-5, 5-6, 6-1

Κύκλοι και αποκοπές



Κύκλος: 1-2, 2-3, 3-4, 4-5, 5-6, 6-1

Μια **αποκοπή** είναι ένα υποσύνολο κορυφών A. Το αντίστοιχο σύνολο (ακμών) αποκοπής D είναι το υποσύνολο των ακμών με ακριβώς ένα άκρο στο A (το άλλο άκρο στο $V \setminus A$).



$$A = \{4, 5, 8\}$$

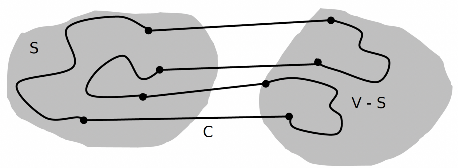
$$D = \{\{3, 4\}, \{3, 5\}, \{5, 6\}, \{5, 7\}, \{7, 8\}\}$$

Τομή κύκλου-αποκοπής

Λήμμα: Ένας κύκλος και μία αποκοπή τέμνονται σε άρτιο πλήθος ακμών.

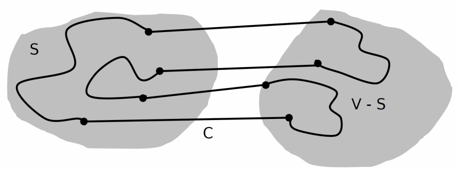
Τομή κύκλου-αποκοπής

Λήμμα: Ένας κύκλος και μία αποκοπή τέμνονται σε άρτιο πλήθος ακμών.



Τομή κύκλου-αποκοπής

Λήμμα: Ένας κύκλος και μία αποκοπή τέμνονται σε άρτιο πλήθος ακμών.



Απόδειξη. Αν ο C βρίσκεται εντός των κορυφών αποκοπής S ή $V - S$, τότε το πλήθος κοινών ακμών $= 0$. Αλλιώς, όσες φορές ο C "μπαίνει" στο S, τόσες θα "βγει".

Ιδιότητες

Ας υποθέσουμε (για απλότητα) ότι όλα τα κόστη ακμών c_e είναι μοναδικά.
Θα αποδείξουμε τα εξής:

Ιδιότητες

Ας υποθέσουμε (για απλότητα) ότι όλα τα κόστη ακμών c_e είναι μοναδικά.
Θα αποδείξουμε τα εξής:

- Ιδιότητα Αποκοπής. Έστω S οποιοδήποτε υποσύνολο κορυφών, και έστω e η ακμή με το ελάχιστο κόστος με ένα μόνο άκρο στο S. Τότε το ΕΕΔ περιέχει την e.
  Εφαρμόζεται στους Prim, Kruskal.

Ιδιότητες

Ας υποθέσουμε (για απλότητα) ότι όλα τα κόστη ακμών c_e είναι μοναδικά. Θα αποδείξουμε τα εξής:

- Ιδιότητα Αποκοπής. Έστω S οποιοδήποτε υποσύνολο κορυφών, και έστω e η ακμή με το ελάχιστο κόστος με ένα μόνο άκρο στο S. Τότε το ΕΕΔ περιέχει την e.
  Εφαρμόζεται στους Prim, Kruskal.

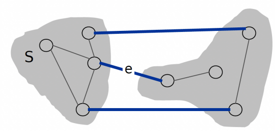
- Ιδιότητα Κύκλου. Έστω C οποιοσδήποτε κύκλος, και έστω f η ακμή με το μέγιστο κόστος που ανήκει στο C. Τότε το ΕΕΔ δεν περιέχει την f.
  Εφαρμόζεται στον Kruskal και Αντίστρ. Διαγραφής.

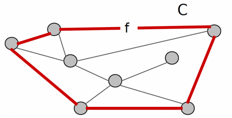
Ιδιότητες

Ας υποθέσουμε (για απλότητα) ότι όλα τα κόστη ακμών c_e είναι μοναδικά. Θα αποδείξουμε τα εξής:

- Ιδιότητα Αποκοπής. Έστω S οποιοδήποτε υποσύνολο κορυφών, και έστω e η ακμή με το ελάχιστο κόστος με ένα μόνο άκρο στο S. Τότε το ΕΕΔ περιέχει την e.
  Εφαρμόζεται στους Prim, Kruskal.

- Ιδιότητα Κύκλου. Έστω C οποιοσδήποτε κύκλος, και έστω f η ακμή με το μέγιστο κόστος που ανήκει στο C. Τότε το ΕΕΔ δεν περιέχει την f.
  Εφαρμόζεται στον Kruskal και Αντίστρ. Διαγραφής.





Ιδιότητα Αποκοπής

Έστω S οποιοδήποτε υποσύνολο κορυφών, και έστω e η ακμή με το ελάχιστο κόστος, με ένα μόνο άκρο στο S. Τότε το ΕΕΔ T^* περιέχει την ε.

Απόδειξη.

Ιδιότητα Αποκοπής

Έστω S οποιοδήποτε υποσύνολο κορυφών, και έστω e η ακμή με το ελάχιστο κόστος, με ένα μόνο άκρο στο S. Τότε το ΕΕΔ T^* περιέχει την ε.

Απόδειξη.

- Υποθέτουμε ότι η e δεν ανήκει στο T^*.

Ιδιότητα Αποκοπής

Έστω S οποιοδήποτε υποσύνολο κορυφών, και έστω e η ακμή με το ελάχιστο κόστος, με ένα μόνο άκρο στο S. Τότε το ΕΕΔ T^* περιέχει την ε.

Απόδειξη.

- Υποθέτουμε ότι η e δεν ανήκει στο T^*.
- Προσθέτοντας την e στο T^* δημιουργείται κύκλος C στο T^*.

Ιδιότητα Αποκοπής

Έστω S οποιοδήποτε υποσύνολο κορυφών, και έστω e η ακμή με το ελάχιστο κόστος, με ένα μόνο άκρο στο S. Τότε το ΕΕΔ T^* περιέχει την ε.

Απόδειξη.

- Υποθέτουμε ότι η e δεν ανήκει στο T^*.
- Προσθέτοντας την e στο T^* δημιουργείται κύκλος C στο T^*.
- Η ακμή e ανήκει στον κύκλο C και στο σύνολο ακμών αποκοπής D που αντιστοιχεί στο S.

Ιδιότητα Αποκοπής

Έστω S οποιοδήποτε υποσύνολο κορυφών, και έστω e η ακμή με το ελάχιστο κόστος, με ένα μόνο άκρο στο S. Τότε το ΕΕΔ T^* περιέχει την ε.

Απόδειξη.

- Υποθέτουμε ότι η e δεν ανήκει στο T^*.
- Προσθέτοντας την e στο T^* δημιουργείται κύκλος C στο T^*.
- Η ακμή e ανήκει στον κύκλο C και στο σύνολο ακμών αποκοπής D που αντιστοιχεί στο S.
- Από το Λήμμα υπάρχει τουλάχιστον μια άλλη ακμή, έστω f, που ανήκει στα C και D.

Ιδιότητα Αποκοπής

Έστω S οποιοδήποτε υποσύνολο κορυφών, και έστω e η ακμή με το ελάχιστο κόστος, με ένα μόνο άκρο στο S. Τότε το ΕΕΔ T^* περιέχει την ε.

Απόδειξη.

- Υποθέτουμε ότι η e δεν ανήκει στο T^*.
- Προσθέτοντας την e στο T^* δημιουργείται κύκλος C στο T^*.
- Η ακμή e ανήκει στον κύκλο C και στο σύνολο ακμών αποκοπής D που αντιστοιχεί στο S.
- Από το Λήμμα υπάρχει τουλάχιστον μια άλλη ακμή, έστω f, που ανήκει στα C και D.
- $T' = T^* \cup \{e\} - \{f\}$ είναι επίσης επικαλύπτον δέντρο.

Ιδιότητα Αποκοπής

Έστω S οποιοδήποτε υποσύνολο κορυφών, και έστω e η ακμή με το ελάχιστο κόστος, με ένα μόνο άκρο στο S. Τότε το ΕΕΔ T^* περιέχει την ε.

Απόδειξη.

- Υποθέτουμε ότι η e δεν ανήκει στο T^*.
- Προσθέτοντας την e στο T^* δημιουργείται κύκλος C στο T^*.
- Η ακμή e ανήκει στον κύκλο C και στο σύνολο ακμών αποκοπής D που αντιστοιχεί στο S.
- Από το Λήμμα υπάρχει τουλάχιστον μια άλλη ακμή, έστω f, που ανήκει στα C και D.
- $T' = T^* \cup \{e\} - \{f\}$ είναι επίσης επικαλύπτον δέντρο.
- Αφού $c_e < c_f$, κόστος$(T') <$ κόστος(T^*).

Ιδιότητα Αποκοπής

Έστω S οποιοδήποτε υποσύνολο κορυφών, και έστω e η ακμή με το ελάχιστο κόστος, με ένα μόνο άκρο στο S. Τότε το ΕΕΔ T^* περιέχει την ε.

Απόδειξη.

- Υποθέτουμε ότι η e δεν ανήκει στο T^*.
- Προσθέτοντας την e στο T^* δημιουργείται κύκλος C στο T^*.
- Η ακμή e ανήκει στον κύκλο C και στο σύνολο ακμών αποκοπής D που αντιστοιχεί στο S.
- Από το Λήμμα υπάρχει τουλάχιστον μια άλλη ακμή, έστω f, που ανήκει στα C και D.
- $T' = T^* \cup \{e\} - \{f\}$ είναι επίσης επικαλύπτον δέντρο.
- Αφού $c_e < c_f$, κόστος(T') < κόστος(T^*).
- Άτοπο.

Ιδιότητα Κύκλου

Έστω C οποιοσδήποτε κύκλος, και έστω f η ακμή με μέγιστο κόστος που ανήκει στον C. Τότε το ΕΕΔ T^* δεν περιέχει την f.

Απόδειξη.

Ιδιότητα Κύκλου

Έστω C οποιοσδήποτε κύκλος, και έστω f η ακμή με μέγιστο κόστος που ανήκει στον C. Τότε το ΕΕΔ T^* δεν περιέχει την f.

Απόδειξη.

- Υποθέτουμε ότι η f ανήκει στο T^*.

Ιδιότητα Κύκλου

Έστω C οποιοσδήποτε κύκλος, και έστω f η ακμή με μέγιστο κόστος που ανήκει στον C. Τότε το ΕΕΔ T^* δεν περιέχει την f.

Απόδειξη.

- Υποθέτουμε ότι η f ανήκει στο T^*.
- Διαγράφοντας την f από το T^* δημιουργείται αποκοπή S στο T^*.

Ιδιότητα Κύκλου

Έστω C οποιοσδήποτε κύκλος, και έστω f η ακμή με μέγιστο κόστος που ανήκει στον C. Τότε το ΕΕΔ T^* δεν περιέχει την f.

Απόδειξη.

- Υποθέτουμε ότι η f ανήκει στο T^*.
- Διαγράφοντας την f από το T^* δημιουργείται αποκοπή S στο T^*.
- Η f ανήκει στον C και στο σύνολο ακμών αποκοπής D που αντιστοιχεί στις κορυφές S.

Ιδιότητα Κύκλου

Έστω C οποιοσδήποτε κύκλος, και έστω f η ακμή με μέγιστο κόστος που ανήκει στον C. Τότε το ΕΕΔ T^* δεν περιέχει την f.

Απόδειξη.

- Υποθέτουμε ότι η f ανήκει στο T^*.
- Διαγράφοντας την f από το T^* δημιουργείται αποκοπή S στο T^*.
- Η f ανήκει στον C και στο σύνολο ακμών αποκοπής D που αντιστοιχεί στις κορυφές S.
- Από το Λήμμα υπάρχει άλλη ακμή e, που ανήκει στα C και D.

Ιδιότητα Κύκλου

Έστω C οποιοσδήποτε κύκλος, και έστω f η ακμή με μέγιστο κόστος που ανήκει στον C. Τότε το ΕΕΔ T^* δεν περιέχει την f.

Απόδειξη.

- Υποθέτουμε ότι η f ανήκει στο T^*.
- Διαγράφοντας την f από το T^* δημιουργείται αποκοπή S στο T^*.
- Η f ανήκει στον C και στο σύνολο ακμών αποκοπής D που αντιστοιχεί στις κορυφές S.
- Από το Λήμμα υπάρχει άλλη ακμή e, που ανήκει στα C και D.
- $T' = T^* \cup \{e\} - \{f\}$ είναι επίσης επικαλύπτον δέντρο.

Ιδιότητα Κύκλου

Έστω C οποιοσδήποτε κύκλος, και έστω f η ακμή με μέγιστο κόστος που ανήκει στον C. Τότε το ΕΕΔ T^* δεν περιέχει την f.

Απόδειξη.

- Υποθέτουμε ότι η f ανήκει στο T^*.
- Διαγράφοντας την f από το T^* δημιουργείται αποκοπή S στο T^*.
- Η f ανήκει στον C και στο σύνολο ακμών αποκοπής D που αντιστοιχεί στις κορυφές S.
- Από το Λήμμα υπάρχει άλλη ακμή e, που ανήκει στα C και D.
- $T' = T^* \cup \{e\} - \{f\}$ είναι επίσης επικαλύπτον δέντρο.
- Αφού $c_e < c_f$, τότε κόστος(T') < κόστος (T^*).

Ιδιότητα Κύκλου

Έστω C οποιοσδήποτε κύκλος, και έστω f η ακμή με μέγιστο κόστος που ανήκει στον C. Τότε το ΕΕΔ T^* δεν περιέχει την f.

Απόδειξη.

- Υποθέτουμε ότι η f ανήκει στο T^*.
- Διαγράφοντας την f από το T^* δημιουργείται αποκοπή S στο T^*.
- Η f ανήκει στον C και στο σύνολο ακμών αποκοπής D που αντιστοιχεί στις κορυφές S.
- Από το Λήμμα υπάρχει άλλη ακμή e, που ανήκει στα C και D.
- $T' = T^* \cup \{e\} - \{f\}$ είναι επίσης επικαλύπτον δέντρο.
- Αφού $c_e < c_f$, τότε κόστος(T') < κόστος (T^*).
- Άτοπο.

Αλγόριθμος του Prim

- Αρχικοποιούμε το $S = \{\text{οποιαδήποτε κορυφή}\}$.

Αλγόριθμος του Prim

- Αρχικοποιούμε το $S = \{\text{οποιαδήποτε κορυφή}\}$.

- Εφαρμόζουμε την ιδιότητα αποκοπής στο S.

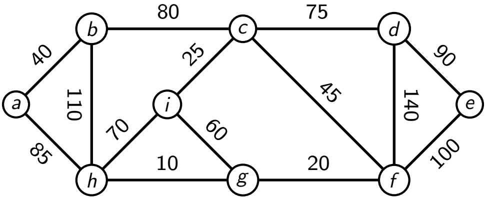
Αλγόριθμος του Prim

- Αρχικοποιούμε το $S = \{\text{οποιαδήποτε κορυφή}\}$.

- Εφαρμόζουμε την ιδιότητα αποκοπής στο S.

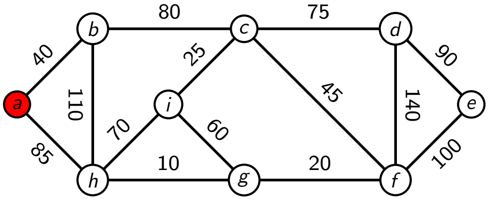
- Προσθέτουμε στο T την ακμή με το ελάχιστο κόστος στο σύνολο ακμών αποκοπής που αντιστοιχεί στο S, και προσθέτουμε μια νέα κορυφή (2ο άκρο ακμής) στο S.

Παράδειγμα



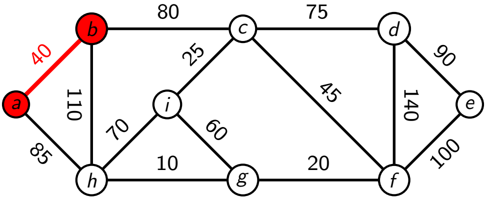
Ξεκινάμε από την κορυφή a.

Παράδειγμα



Ξεκινάμε από την κορυφή a.

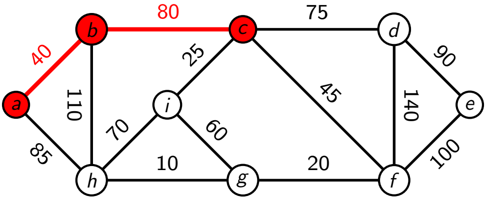
Παράδειγμα



Ξεκινάμε από την κορυφή a.

$T = \{\{a, b\}$

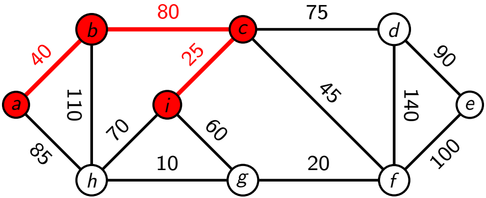
Παράδειγμα



Ξεκινάμε από την κορυφή a.

$$T = \{\{a, b\}, \{b, c\}$$

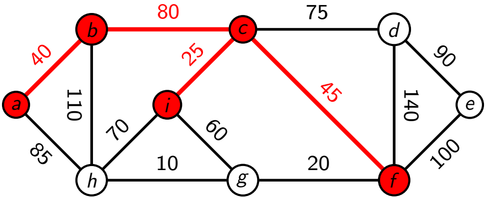
Παράδειγμα



Ξεκινάμε από την κορυφή a.

$T = \{\{a, b\}, \{b, c\}, \{c, i\}$

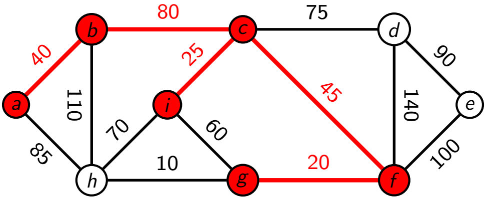
Παράδειγμα



Ξεκινάμε από την κορυφή a.

$$T = \{\{a,b\},\{b,c\},\{c,i\},\{c,f\}\}$$

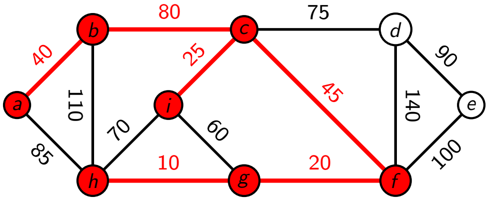
Παράδειγμα



Ξεκινάμε από την κορυφή a.

$T = \{\{a, b\}, \{b, c\}, \{c, i\}, \{c, f\}, \{f, g\}\}$

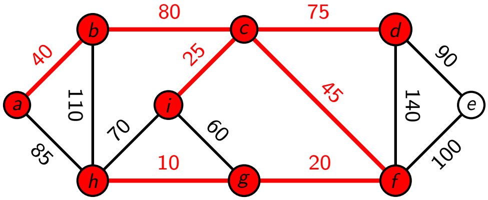
Παράδειγμα



Ξεκινάμε από την κορυφή a.

$$T = \{\{a, b\}, \{b, c\}, \{c, i\}, \{c, f\}, \{f, g\}, \{g, h\}\}$$

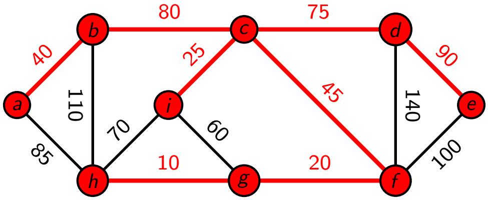
Παράδειγμα



Ξεκινάμε από την κορυφή a.

$$T = \{\{a,b\},\{b,c\},\{c,i\},\{c,f\},\{f,g\},\{g,h\},\{c,d\}\}$$

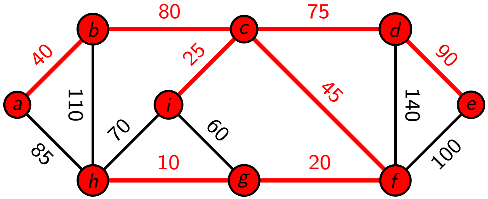
Παράδειγμα



Ξεκινάμε από την κορυφή a.

$$T = \{\{a, b\}, \{b, c\}, \{c, i\}, \{c, f\}, \{f, g\}, \{g, h\}, \{c, d\}, \{d, e\}\}$$

Παράδειγμα



Ξεκινάμε από την κορυφή a.

$$T = \{\{a, b\}, \{b, c\}, \{c, i\}, \{c, f\}, \{f, g\}, \{g, h\}, \{c, d\}, \{d, e\}\}$$

Ψευδοκώδικας

Prim(G, c, s)
Για κάθε $(v \in V)$ $\alpha[v] \leftarrow \infty$ και $\pi(v) \leftarrow 0$
Αρχικοποίησε μια άδεια ουρά προτεραιότητας Q
$\alpha[s] \leftarrow 0$
Για κάθε $(v \in V)$ εισήγαγε την v στην Q
Αρχικοποίησε σύνολο εξερευνημένων κορυφών $S \leftarrow \emptyset$
Ενόσω (Q δεν είναι άδεια)
  $u \leftarrow$ διέγραψε το ελάχιστο στοιχείο της Q
  $S \leftarrow S \cup \{u\}$
  $T \leftarrow T \cup \{u, \pi\{u\}\}$
  Για κάθε (ακμή $e = \{u, v\}$ περιέχει την u)
    Εάν $(v \notin S)$ και $c_e < \alpha[v]$ τότε $\alpha[v] \leftarrow c_e$ και $\pi(v) \leftarrow u$.

Υλοποίηση

Χρησιμοποιούμε ουρά προτεραιότητας όπως ο Dijkstra.

- Διατηρούμε σύνολο κορυφών S που έχουν εξερευνηθεί (και T).

Υλοποίηση

Χρησιμοποιούμε ουρά προτεραιότητας όπως ο Dijkstra.

- Διατηρούμε σύνολο κορυφών S που έχουν εξερευνηθεί (και T).
- Κάθε κορυφή v που δεν έχει εξερευνηθεί βρίσκεται στο Q, με κόστος επισύναψης $\alpha[v] =$ κόστος φθηνότερης ακμής από v σε κορυφή στο S. Το $\alpha[]$ είναι η προτεραιότητα στην ουρά Q.

Υλοποίηση

Χρησιμοποιούμε ουρά προτεραιότητας όπως ο Dijkstra.

- Διατηρούμε σύνολο κορυφών S που έχουν εξερευνηθεί (και T).
- Κάθε κορυφή v που δεν έχει εξερευνηθεί βρίσκεται στο Q, με κόστος επισύναψης $\alpha[v] =$ κόστος φθηνότερης ακμής από v σε κορυφή στο S. Το $\alpha[]$ είναι η προτεραιότητα στην ουρά Q.
- $\mathcal{O}(n^2)$ με πίνακα κορυφών, $\mathcal{O}(m \log n)$ updates με δυαδικό σωρό (λογαριθμικό κόστος ελέγχου $v \in S$ και ανανέωσης σωρού).

Αλγόριθμος του Kruskal

Θεωρούμε ακμές με αυξανόμενη σειρά κόστους, διατηρούμε T.

Αλγόριθμος του Kruskal

Θεωρούμε ακμές με αυξανόμενη σειρά κόστους, διατηρούμε T.

Περίπτωση 1: Αν προσθέτοντας την e στο T δημιουργεί κύκλο, απορρίπτουμε την e σύμφωνα με την ιδιότητα κύκλου.

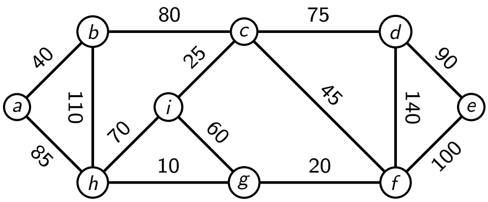
Αλγόριθμος του Kruskal

Θεωρούμε ακμές με αυξανόμενη σειρά κόστους, διατηρούμε T.

Περίπτωση 1: Αν προσθέτοντας την e στο T δημιουργεί κύκλο, απορρίπτουμε την e σύμφωνα με την ιδιότητα κύκλου.

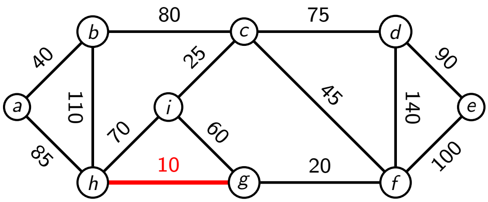
Περίπτωση 2: Αλλιώς, εισάγουμε $e = \{u, v\}$ στο T από την ιδιότητα αποκοπής, όπου $S = \{$κορυφές στη συνεκτική συνιστώσα της $u\}$, e η φθηνότερη στις ακμές αποκοπής.

Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

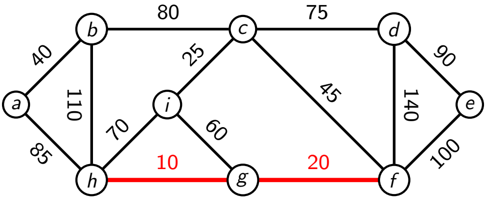
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$T = \{\{g, h\}$

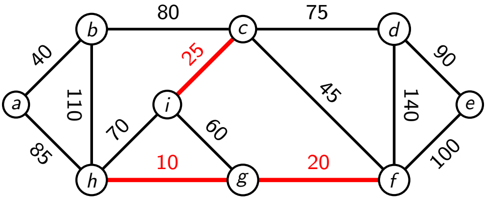
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g, h\}, \{f, g\}\}$$

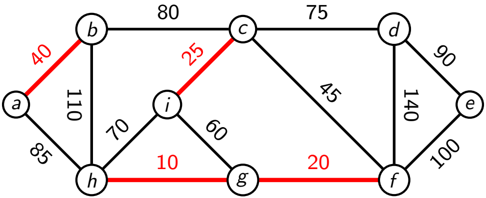
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g, h\}, \{f, g\}, \{c, i\}\}$$

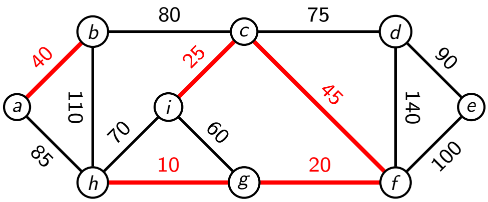
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g,h\},\{f,g\},\{c,i\},\{a,b\}\}$$

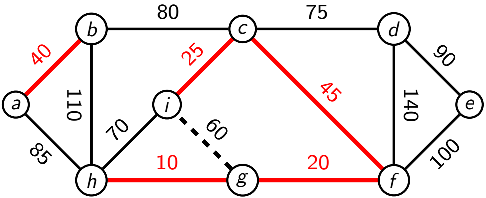
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g,h\},\{f,g\},\{c,i\},\{a,b\},\{c,f\}\}$$

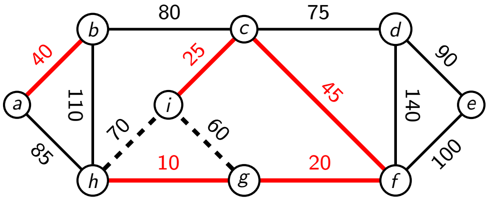
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g,h\},\{f,g\},\{c,i\},\{a,b\},\{c,f\}\}$$

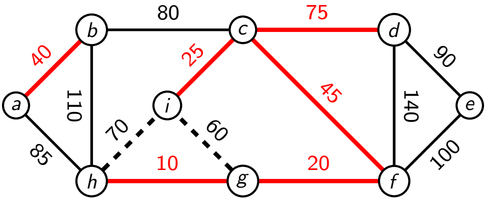
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g, h\}, \{f, g\}, \{c, i\}, \{a, b\}, \{c, f\}\}$$

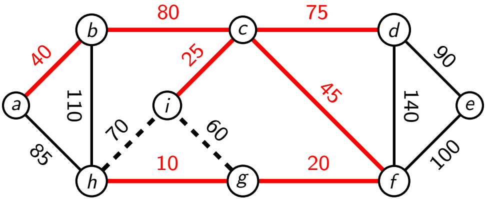
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g,h\},\{f,g\},\{c,i\},\{a,b\},\{c,f\},\{c,d\}\}$$

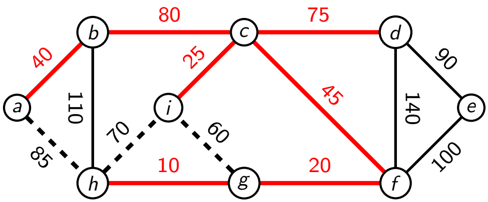
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g,h\},\{f,g\},\{c,i\},\{a,b\},\{c,f\},\{c,d\},\{b,c\}\}$$

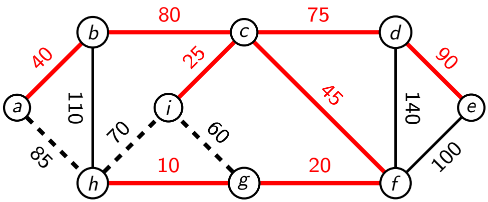
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g,h\},\{f,g\},\{c,i\},\{a,b\},\{c,f\},\{c,d\},\{b,c\}$$

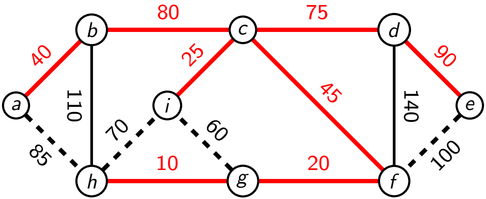
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g,h\},\{f,g\},\{c,i\},\{a,b\},\{c,f\},\{c,d\},\{b,c\},\{d,e\}\}$$

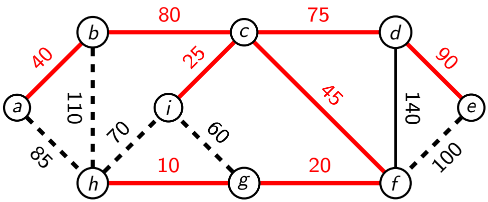
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g,h\},\{f,g\},\{c,i\},\{a,b\},\{c,f\},\{c,d\},\{b,c\},\{d,e\}\}$$

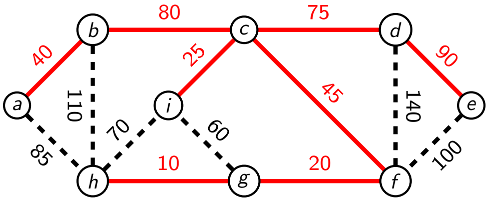
Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g,h\},\{f,g\},\{c,i\},\{a,b\},\{c,f\},\{c,d\},\{b,c\},\{d,e\}\}$$

Παράδειγμα



Ο αλγόριθμος ταξινομεί τις ακμές κατά αύξον κόστος. Ακμή που δεν δημιουργεί κύκλο μπαίνει στο ΕΕΔ αλλιώς απορρίπτεται.

$$T = \{\{g,h\},\{f,g\},\{c,i\},\{a,b\},\{c,f\},\{c,d\},\{b,c\},\{d,e\}\}$$

Ψευδοκώδικας

Kruskal(G, c)
Ταξινόμησε κόστη ακμών ώστε $c_1 \leq c_2 \leq \cdots \leq c_m$.
$T \leftarrow \emptyset$
Για κάθε $(u \in V)$ φτιάξε ένα σύνολο που περιέχει την u
Για $i = 1$ έως m
  $\{u, v\} = e_i$
  Εάν οι u, v περιέχονται σε διαφορετικά σύνολα
    $T \leftarrow T \cup \{e_i\}$
    ένωσε τα σύνολα/συνιστώσες που περιέχουν τις u, v
επίστρεψε T

Υλοποίηση

Χρησιμοποιούμε την δομή δεδομένων union-find.

Υλοποίηση

Χρησιμοποιούμε την δομή δεδομένων union-find.

- Στόχος: σύνολο με τις ακμές στο ΕΕΔ T.

Υλοποίηση

Χρησιμοποιούμε την δομή δεδομένων union-find.

- Στόχος: σύνολο με τις ακμές στο ΕΕΔ T.
- Διατηρούμε ένα σύνολο για κάθε συνεκτική συνιστώσα του T.

Υλοποίηση

Χρησιμοποιούμε την δομή δεδομένων union-find.

- Στόχος: σύνολο με τις ακμές στο ΕΕΔ T.
- Διατηρούμε ένα σύνολο για κάθε συνεκτική συνιστώσα του T.
- $\mathcal{O}(m \log n)$ για ταξινόμηση, $\mathcal{O}(m\alpha(n))$ για εύρεση της ένωσης.

Ντετερμινιστικοί συγκρισιπαγείς (comparison-based) αλγόριθμοι.

- $\mathcal{O}(m \log n)$ [Jarník, Prim, Dijkstra, Kruskal, Boruvka]
- $\mathcal{O}(m \log \log n)$. [Cheriton-Tarjan 1976, Yao 1975]
- $\mathcal{O}(m\beta(m, n))$. [Fredman-Tarjan 1987]
- $\mathcal{O}(m \log \beta(m, n))$. [Gabow-Galil-Spencer-Tarjan 1986]
- $\mathcal{O}(m\alpha(m, n))$. [Chazelle 2000]

Αξιοσημείωτοι.

- $\mathcal{O}(m)$ τυχαιοποιημένος. [Karger-Klein-Tarjan 1995]
- $\mathcal{O}(m)$ επαλήθευση. [Dixon-Rauch-Tarjan 1992]

Ευκλείδιες αποστάσεις.

- 2 διαστάσεις: $\mathcal{O}(n \log n)$ ΕΕΔ ακμών τριγωνοποίησης Delaunay
- k διαστάσεις: $\mathcal{O}(k \cdot n^2)$. Πυκνός αλγόριθμος του Prim

Ιερό δισκοπότηρο: $\mathcal{O}(m)$.