ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών

Αλγόριθμοι και Πολυπλοκότητα

Ενότητα 3

Αλγόριθμοι Γραφημάτων

Prim-Kruskal

Ν. Μ. Μισυρλής

Τμήμα Πληροφορικής και Τηλεπικοινωνιών,

Άπληστοι (Greedy) Αλγόριθμοι και Γραφήματα

- Δέντρα Επικάλυψης (spanning trees)
- Βέλτιστη Υπακολουθία
- Άπληστη Επιλογή
- Αλγόριθμος του Prim

Γραφήματα

Κατευθυνόμενο Γράφημα: Ένας κατευθυνόμενο γράφημα $G = (V, E)$ είναι ένα διατεταγμένο ζεύγος που αποτελείται από:

- ένα σύνολο V κορυφών
- ένα σύνολο $E \subseteq V \times V$ ακμών

μη Κατευθυνόμενο Γράφημα: Σε έναν μην κατευθυνόμενο γράφημα $G = (V, E)$, το σύνολο των ακμών E αποτελείται από μη διατεταγμένα ζεύγη από κορυφές.

Και στις δυο περιπτώσεις ισχύει:

$$|E| = O(V^2)$$

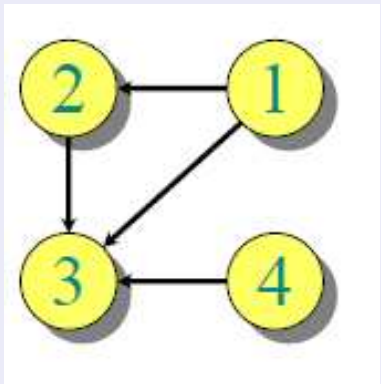
Αν το γράφημα είναι συνδεδεμένο ισχύει:

$$|E| \geq |V| - 1$$
$$\log |E| = \Theta(\log V)$$

Πίνακας Γειτνίασης

Πίνακας Γειτνίασης: Πίνακας γειτνίασης ενός γραφήματος $G = (V, E)$, όπου $V = \{1, 2, \ldots, n\}$, είναι ο πίνακας $A[1 \ldots n, 1 \ldots n]$ με

$$A[i,j] = \begin{cases} 1 & \text{αν } (i,j) \in E, \\ 0 & \text{αν } (i,j) \notin E \end{cases}$$

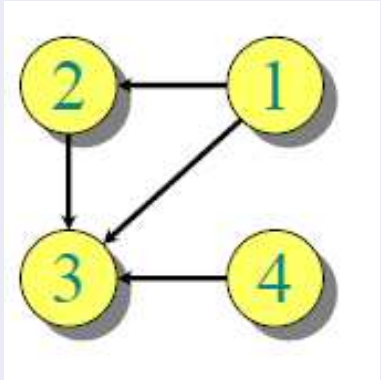


Για πυκνή απεικόνιση απαιτείται $\Theta(V^2)$ χώρος αποθήκευσης

Λίστα Γειτνίασης

Λίστα Γειτνίασης: Λίστα γειτνίασης μιας κορυφής $v \in V$ είναι η λίστα $Adj[v]$ από κορυφές γειτονικές της v.

Παράδειγμα



Για μη κατευθυνόμενα γραφήματα, $|Adj[v]| = degree(v)$.
Ενώ για κατευθυνόμενα γραφήματα, $|Adj[v]| = out - degree(v)$.

Θεώρημα της Χειραψίας:
Για μη κατευθυνόμενα γραφήματα, $\sum_{v \in V} = 2|E| \Rightarrow$ οι λίστες γειτνίασης χρησιμοποιούν $\Theta(V + E)$ αποθηκευτικό χώρο - αραιή απεικόνιση (και για τους δύο τύπους γραφημάτων)

Αλγόριθμοι Γραφημάτων

Πράξεις σε ασύνδετα σύνολα

- Δεδομένου ενός συνόλου στοιχείων θέλουμε να χωρίσουμε τα στοιχεία αυτά σε διαφορετικές μη επικαλυπτόμενες ομάδες.
- Κάθε μία από τις ομάδες αναγνωρίζεται από ένα στοιχείο της (αντιπρόσωπος-representative).
- Μία από τις πολλές εφαρμογές των ασύνδετων συνόλων είναι η εύρεση των συνεκτικών συνιστωσών ενός γραφήματος.

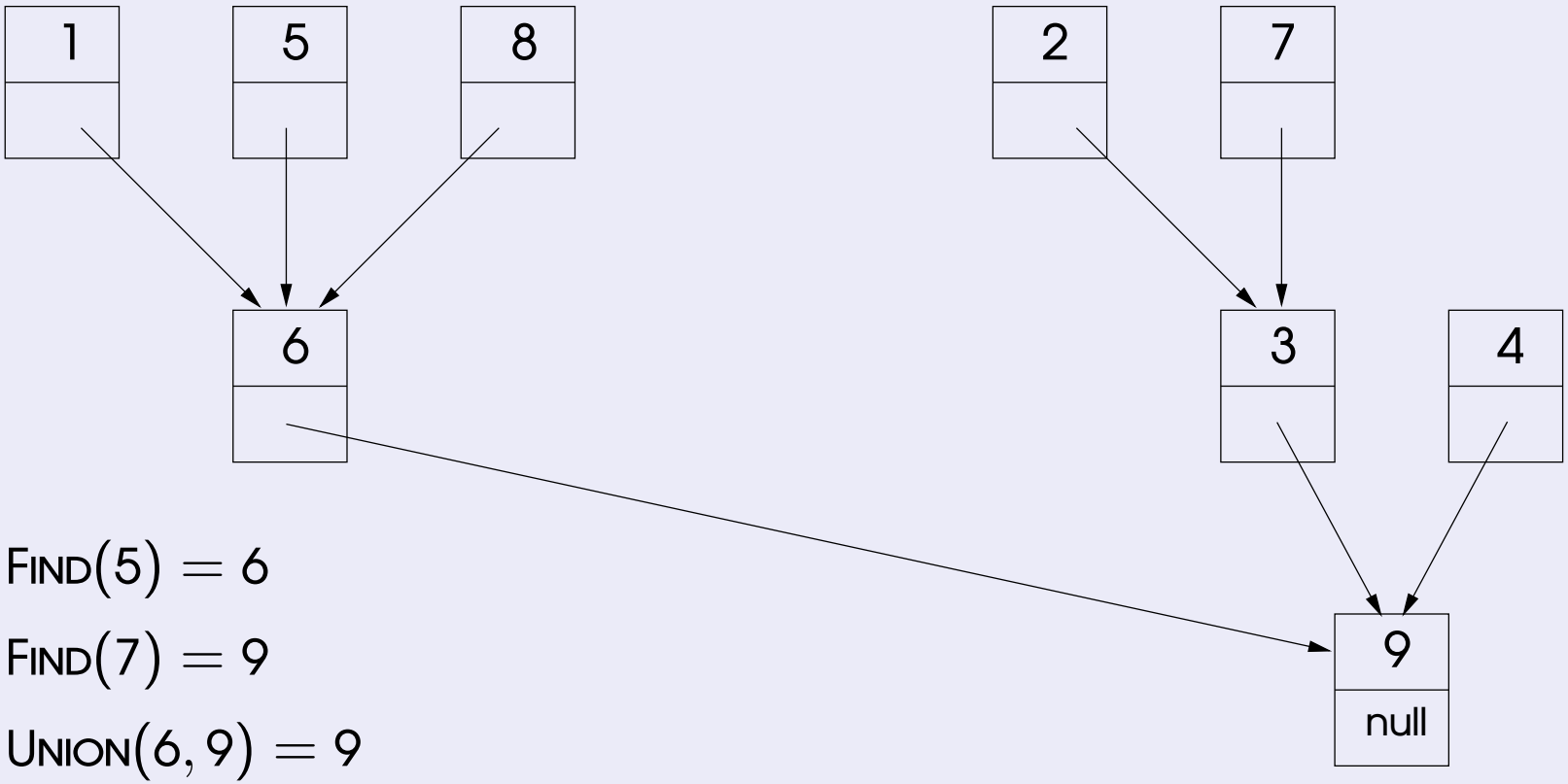
Αλγόριθμοι Γραφημάτων

Πράξεις σε ασύνδετα σύνολα

- Οι βασικές πράξεις είναι οι εξής:
- FIND(x): Δεδομένου ενός στοιχείου x επιστρέφει τον αντιπρόσωπο της ομάδας στην οποία ανήκει το x.
- UNION(x, y): Ενώνει τις δύο ομάδες / σύνολα που έχουν αντιπροσώπους τα x και y αντίστοιχα, σε μία νέα ομάδα. Αντιπρόσωπος της νέας ομάδας μπορεί να είναι οποιοδήποτε στοιχείο των δύο ομάδων αν και συνήθως επιλέγεται ένας εκ των x και y.
- MAKE-SET(x): Δημιουργεί μια νέα ομάδα η οποία περιέχει το στοιχείο x (το οποίο γίνεται και αντιπρόσωπος της ομάδας).

Πράξεις σε ασύνδετα σύνολα

- Ένα παράδειγμα χρήσης των UNION και FIND



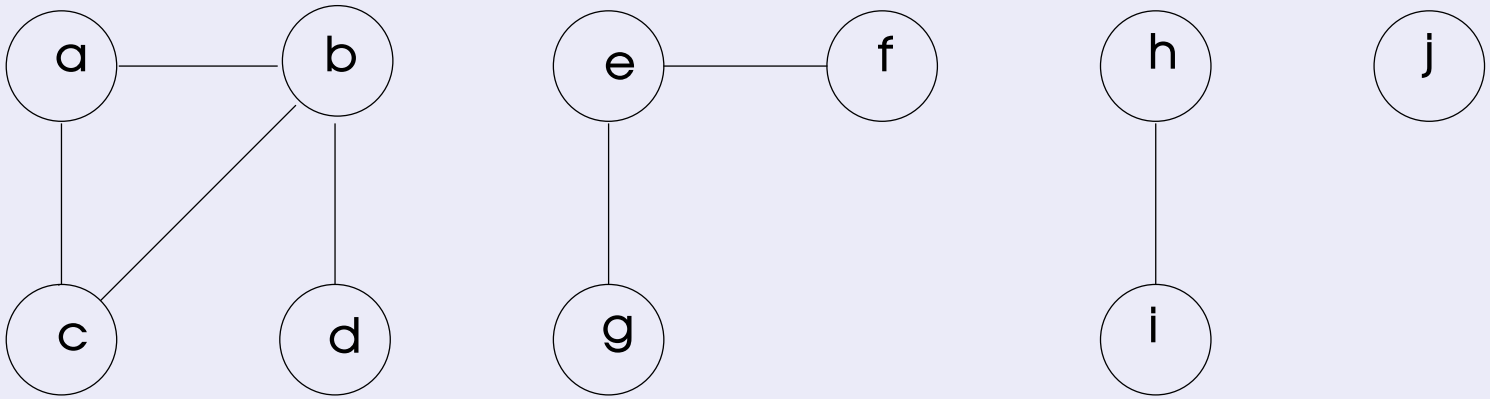
$\text{FIND}(5) = 6$

$\text{FIND}(7) = 9$

$\text{UNION}(6, 9) = 9$

Πράξεις σε ασύνδετα σύνολα

- Ένα γράφημα με τέσσερις συνεκτικές συνιστώσες:
  $\{a, b, c, d\}, \{e, f, g\}, \{h, i\}, \{j\}$.



Αλγόριθμοι Γραφημάτων

Πράξεις σε ασύνδετα σύνολα

- Αλγόριθμος Εύρεσης Συνεκτικών Συνιστωσών με χρήση της δομής ασύνδετων συνόλων

CONNECTED-COMPONENTS (G)

1. **for each** $v \in V$
2. MAKE-SET(v)
3. **for each** $(u, v) \in E$
4. $a = Find(u)$
5. $b = Find(v)$
6. **if** $a \neq b$ **then** $Union(a, b)$
7. **end for**

Αλγόριθμοι Γραφημάτων

Πράξεις σε ασύνδετα σύνολα

- Κατασκευή των συνόλων για το γράφημα του σχήματος

Πλευρά	Συλλογή από ασύνδετα σύνολα									
αρχικά σύνολα	$\{a\}$	$\{b\}$	$\{c\}$	$\{d\}$	$\{e\}$	$\{f\}$	$\{g\}$	$\{h\}$	$\{i\}$	$\{j\}$
(b, d)	$\{a\}$	$\{b, d\}$	$\{c\}$		$\{e\}$	$\{f\}$	$\{g\}$	$\{h\}$	$\{i\}$	$\{j\}$
(e, g)	$\{a\}$	$\{b, d\}$	$\{c\}$		$\{e, g\}$	$\{f\}$		$\{h\}$	$\{i\}$	$\{j\}$
(a, c)	$\{a, c\}$	$\{b, d\}$			$\{e, g\}$	$\{f\}$		$\{h\}$	$\{i\}$	$\{j\}$
(h, i)	$\{a, c\}$	$\{b, d\}$			$\{e, g\}$	$\{f\}$		$\{h, i\}$		$\{j\}$
(a, b)	$\{a, b, c, d\}$				$\{e, g\}$	$\{f\}$		$\{h, i\}$		$\{j\}$
(e, f)	$\{a, b, c, d\}$				$\{e, f, g\}$			$\{h, i\}$		$\{j\}$
(b, c)	$\{a, b, c, d\}$				$\{e, f, g\}$			$\{h, i\}$		$\{j\}$

Πράξεις σε ασύνδετα σύνολα

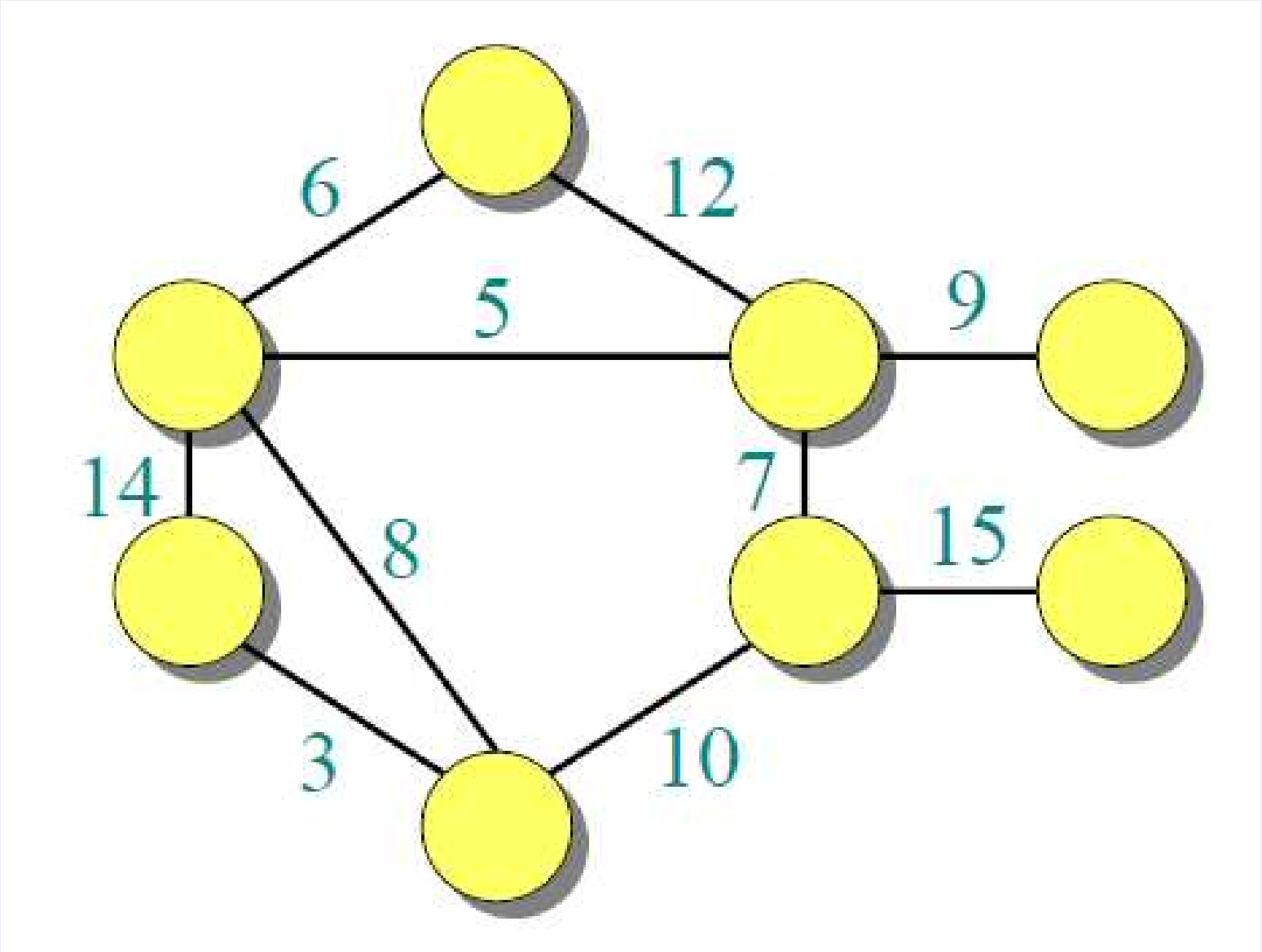
- Έχοντας λοιπόν κατασκευάσει τα σύνολα, μπορούμε να βρούμε εύκολα αν δύο κόμβοι ανήκουν στην ίδια συνεκτική συνιστώσα, ελέγχοντας απλά εάν οι αντιπρόσωποι των ομάδων τους (το αποτέλεσμα της FIND σε κάθε έναν) ταυτίζονται.
- Ο χρόνος εκτέλεσης της πράξης UNION είναι σταθερός.
- Η πράξη FIND, μπορεί να εκτελεσθεί σε γραμμικό χρόνο στην περίπτωση που η αναπαράσταση των στοιχείων ενός συνόλου έχει εκφυλιστεί σε γραμμική λίστα.
- Εάν όμως κατά την ένωση δύο συνόλων επιλέγουμε να ενώνουμε το σύνολο με τα λιγότερα στοιχεία σε αυτό με τα περισσότερα στοιχεία τότε ο χρόνος εκτέλεσης της FIND γίνεται $O(\log n)$ (άσκηση).

Ελαφρύτατο Συνδετικό Δέντρο (Minimum Spanning Tree)

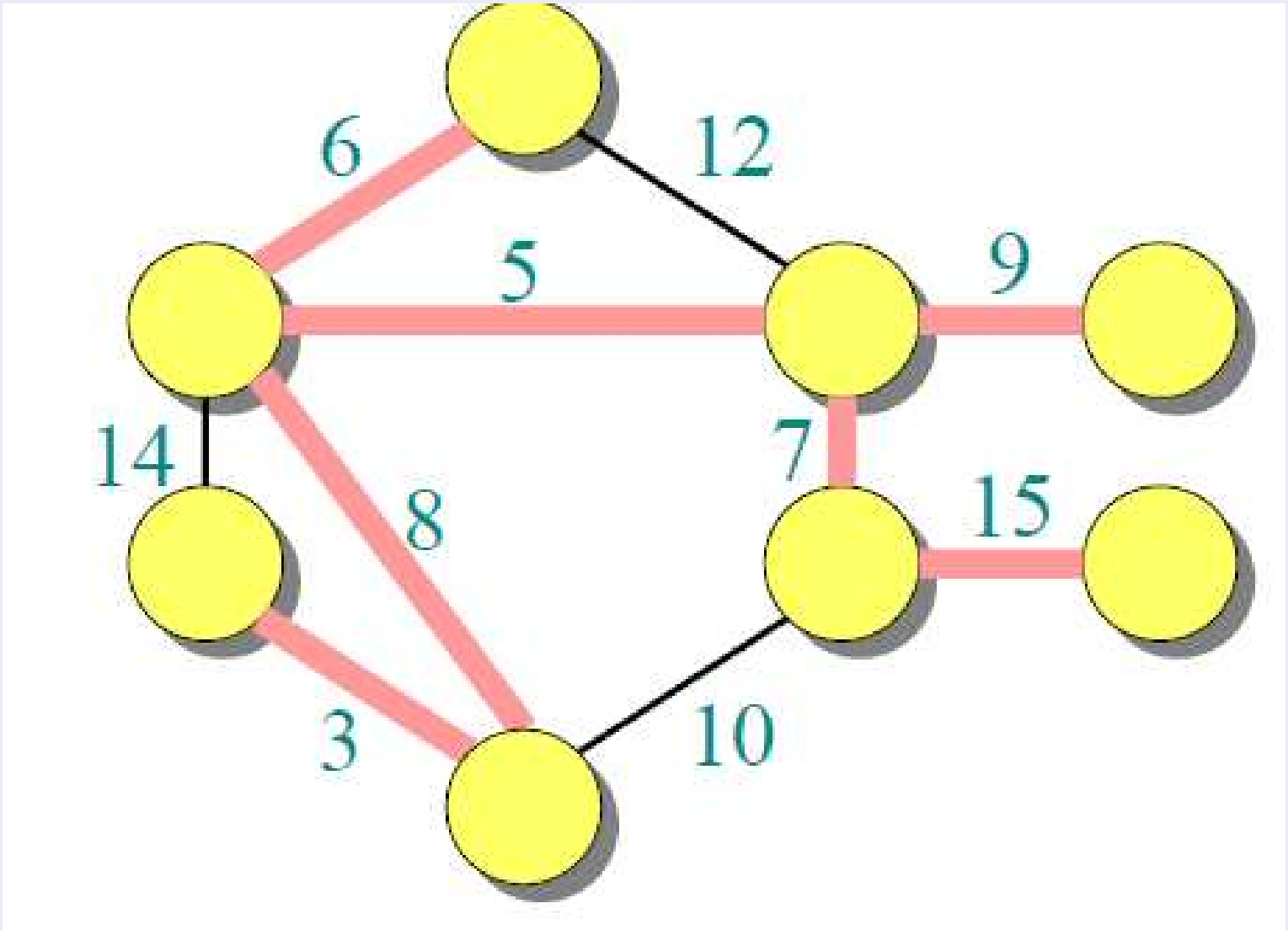
- **Είσοδος:** ένα συνδεδεμένο ακατεύθυντο γράφημα $G = (V, E)$ με συνάρτηση βάρους $w : E \to \mathbb{R}$.
- **΄Εξοδος:** ένα ελαφρύτατο συνδετικό δέντρο T είναι ένα δέντρο, το οποίο συνδέει όλες τις κορυφές, ελάχιστου βάρους:

$$w(T) = \sum_{(u,v)\in T} w(u, v).$$

Παράδειγμα



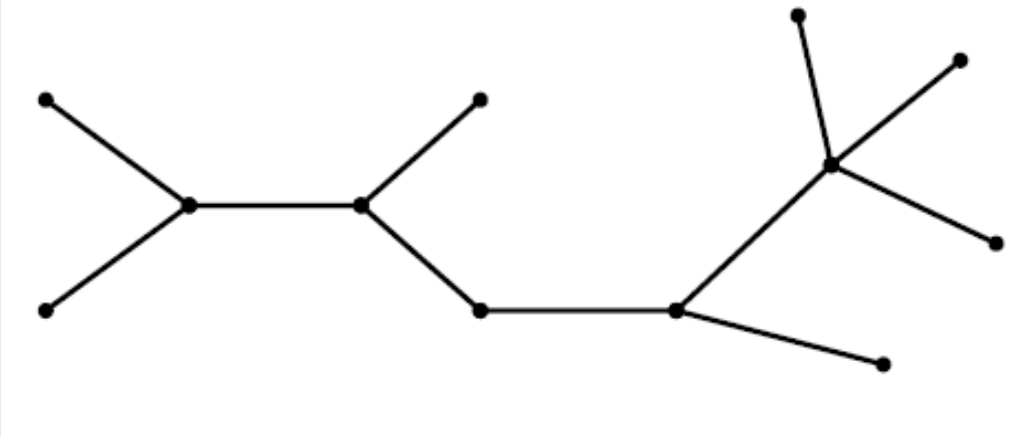
Παράδειγμα



Βέλτιστη Υπακολουθία

T: Ελαφρύτατο Συνδετικό Δέντρο

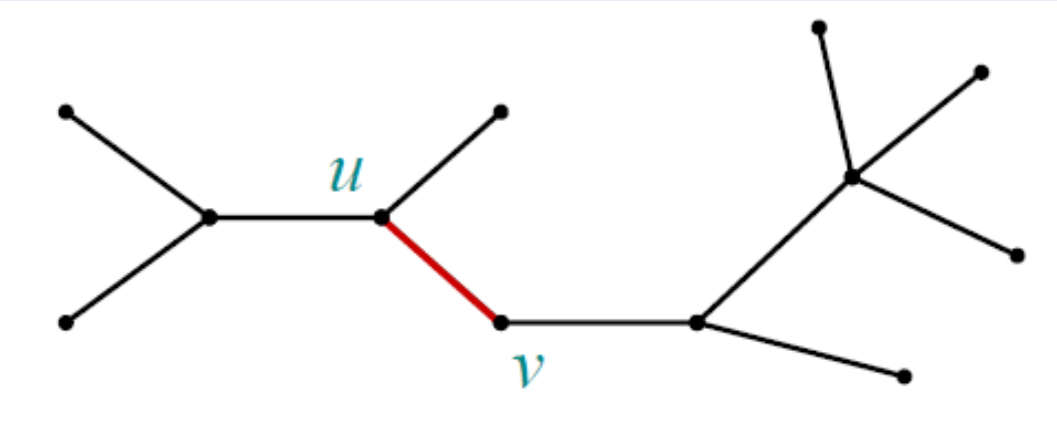
Οι άλλες ακμές του γραφήματος δεν φαίνονται



Βέλτιστη Υπακολουθία

T: Ελαφρύτατο Συνδετικό Δέντρο

Οι άλλες ακμές του γραφήματος δεν φαίνονται

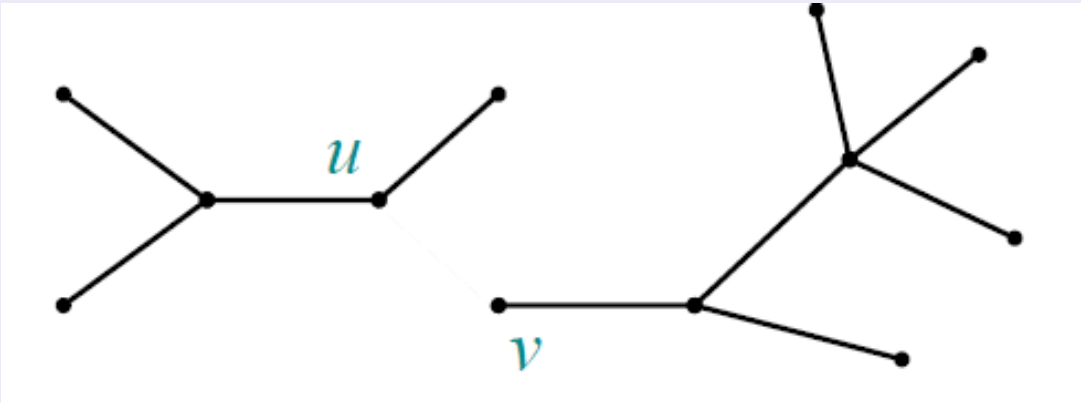


- Αφαίρεση κάποιας ακμής $(u, v) \in T$.

Βέλτιστη Υπακολουθία

T: Ελαφρύτατο Συνδετικό Δέντρο

Οι άλλες ακμές του γραφήματος δεν φαίνονται

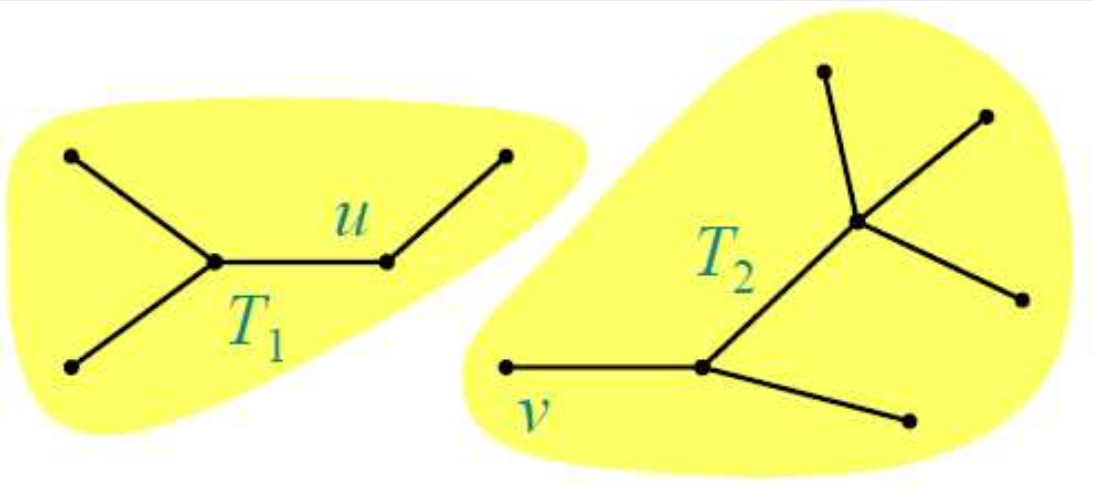


- Αφαίρεση κάποιας ακμής $(u, v) \in T$.

Βέλτιστη Υπακολουθία

Τ: Ελαφρύτατο Συνδετικό Δέντρο

Οι άλλες ακμές του γραφήματος δεν φαίνονται



- Αφαίρεση κάποιας ακμής $(u, v) \in T$.
- Το T χωρίζεται σε δυο υποδέντρα T_1 και T_2.

Theorem

Το υποδέντρο T_1 είναι ένα ελαφρύτατο συνδετικό δέντρο του $G_1 = (V_1, E_1)$ του υπογραφήματος του G παραγόμενου από τις κορυφές του T_1 όπου

$$V_1 = \text{κορυφές του } T_1,$$
$$E_1 = \{(x, y) \in E : x, y \in V_1\}$$

Όμοια και για το T_2

Απόδειξη:

$$w(T) = w(u, v) + w(T_1) + w(T_2)$$

Αν υπήρχε ελαφρύτατο συνδετικό δέντρο T_1' με μικρότερο βάρος από το T_1 για το G_1, τότε το$T' = \{(u, v) \cup T_1' \cup T_2\}$ θα ήταν ένα ελαφρύτατο συνδετικό δέντρο με μικρότερο βάρος από το T για το G. **΄Ατοπο**

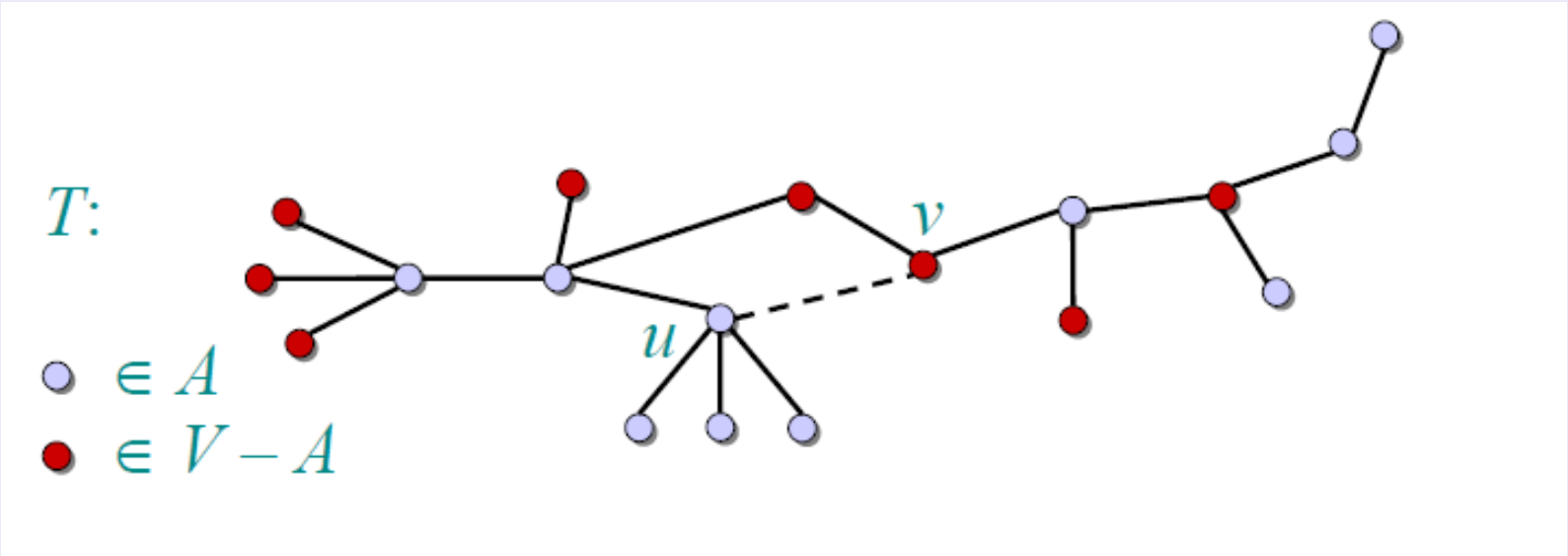
Άπληστοι (greedy) Αλγόριθμοι

- **Ιδιότητα Άπληστης Επιλογής:** Μια τοπικά βέλτιστη επιλογή είναι και καθολικά βέλτιστη.
- **Θεώρημα:** Έστω T το ελαφρύτατο συνδετικό δέντρο του $G = (V, E)$, και $A \subseteq V$. Έστω $(u, v) \in E$ είναι η ακμή με το ελάχιστο βάρος που συνδέει το A με το $V - A$. Τότε,

$$(u, v) \in T.$$

Απόδειξη Θεωρήματος

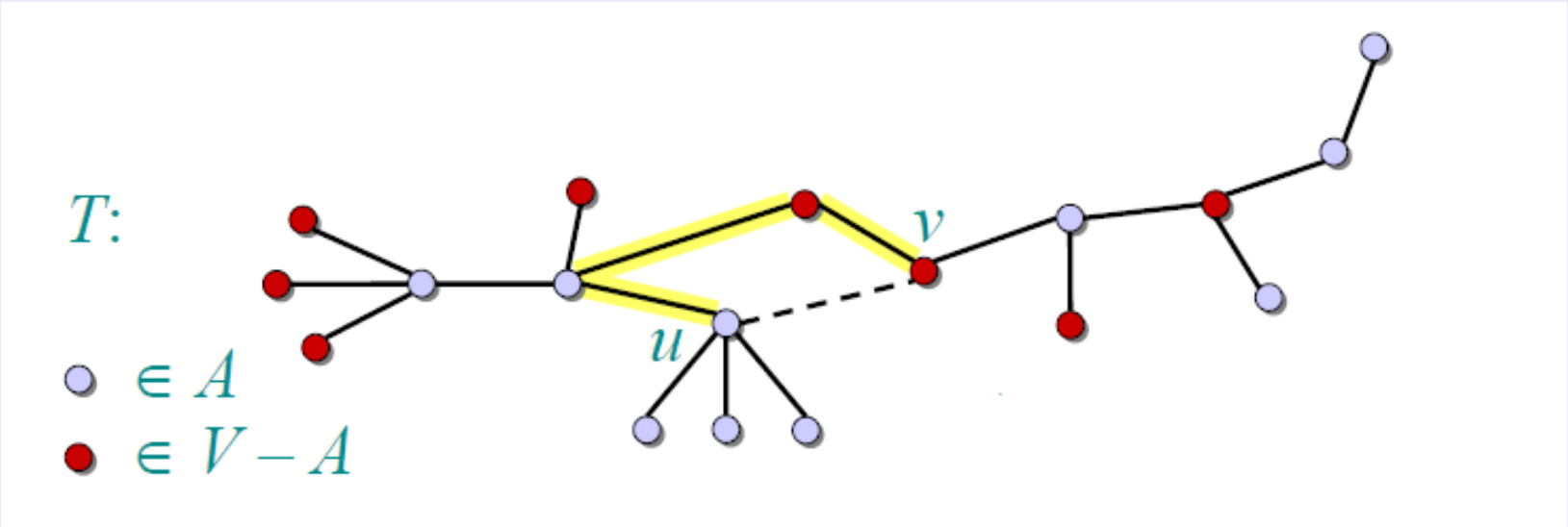
Έστω $(u, v) \notin T$.



Η ακμή (u,v) είναι η ακμή με το ελάχιστο βάρος συνδέουσα το A με το V-A.

Απόδειξη Θεωρήματος

Έστω $(u, v) \notin T$.

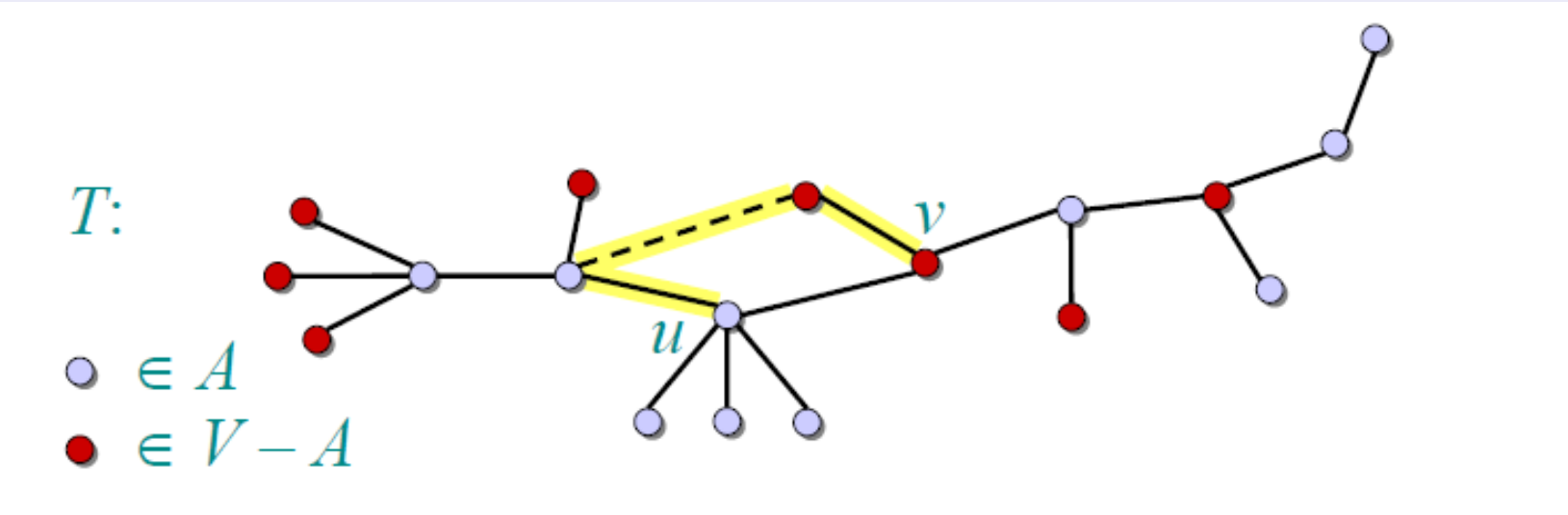


Η ακμή (u,v) είναι η ακμή με το ελάχιστο βάρος συνδέουσα το A με το V-A.

- Θεωρήστε το μοναδικό απλό μονοπάτι από το u στο v στο T.

Απόδειξη Θεωρήματος

Έστω $(u, v) \notin T$.

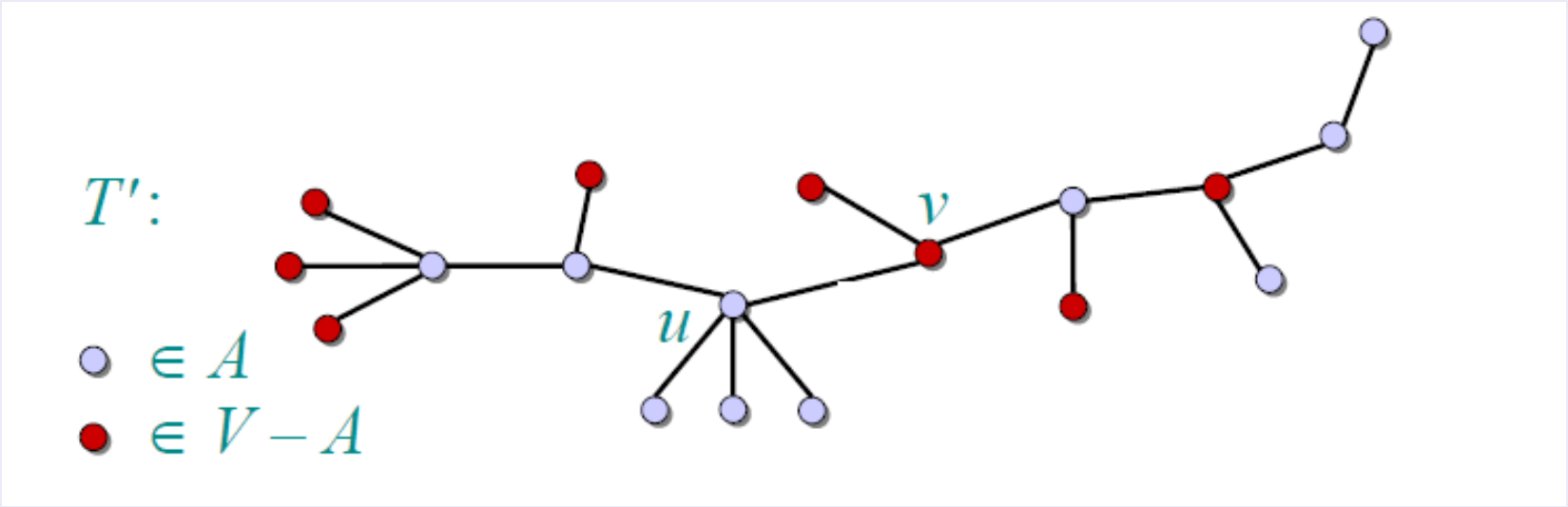


Η ακμή (u,v) είναι η ακμή με το ελάχιστο βάρος συνδέουσα το A με το V-A.

- Θεωρήστε το μοναδικό απλό μονοπάτι από το u στο v στο T.
- Αντικαταστήστε την πρώτη ακμή στο μονοπάτι που συνδέει μια κορυφή του A με μια κορυφή του $V - A$, με την (u, v).

Απόδειξη Θεωρήματος

Έστω $(u, v) \notin T$.



Η ακμή (u,v) είναι η ακμή με το ελάχιστο βάρος συνδέουσα το A με το V-A.

- Θεωρήστε το μοναδικό απλό μονοπάτι από το u στο v στο T.
- Αντικαταστήστε την (u, v) με την πρώτη ακμή στο μονοπάτι που συνδέει μια κορυφή του A με μια κορυφή του $V - A$.
- Προκύπτει ένα ελαφρύτατο συνδετικό δέντρο με μικρότερο βάρος από το T. **΄Ατοπο**

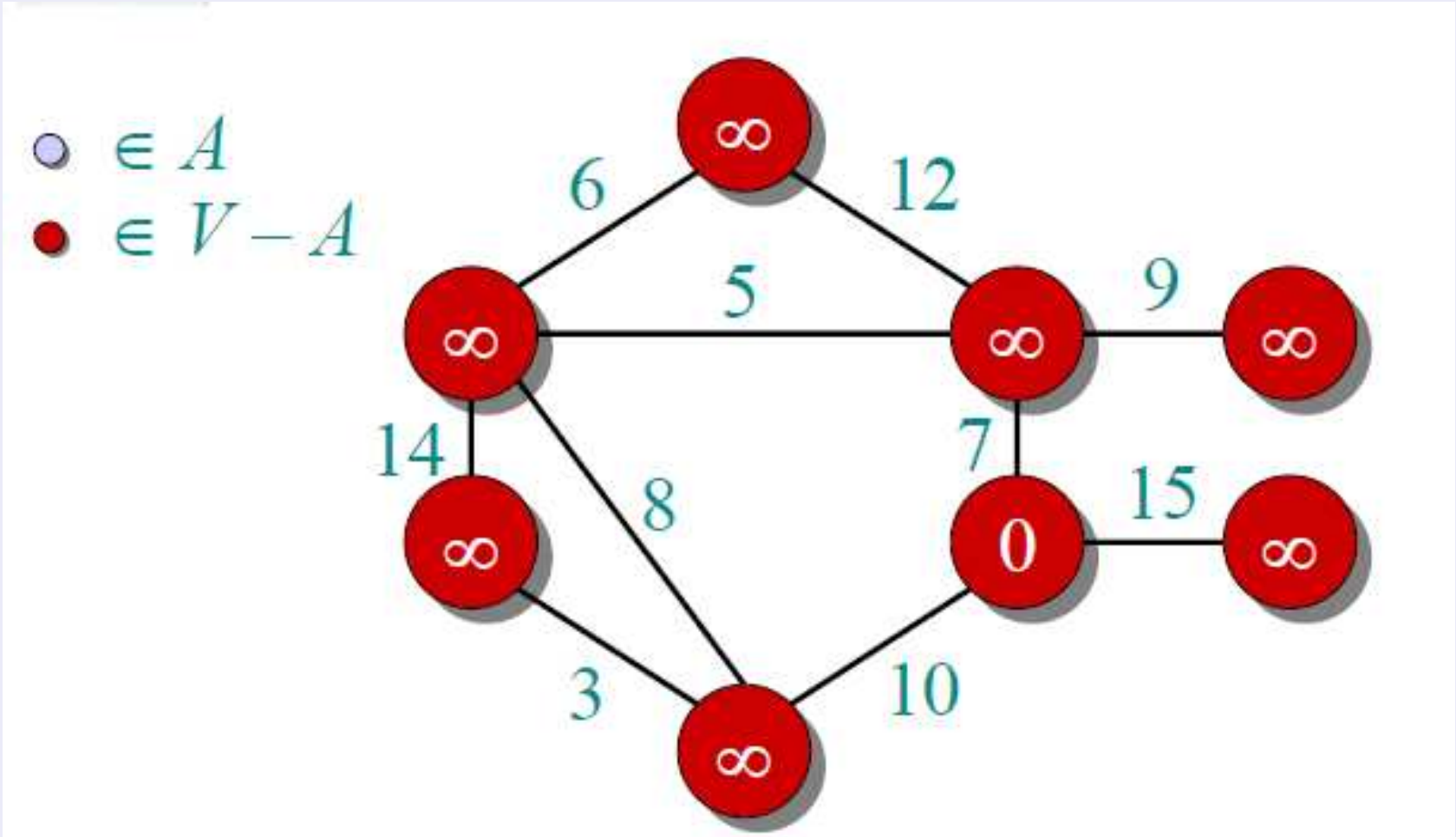
Αλγόριθμος του Prim

Ιδέα: Διατηρούμε τους κόμβους $V - A$ σε μια ουρά προτεραιότητας ελαχίστου Q με βάση ένα πεδίο κλειδί. Για κάθε κόμβο v το $key[v] =$ με το ελάχιστο από τα βάρη των ακμών που συνδέουν τον v με ένα κόμβο στο A.

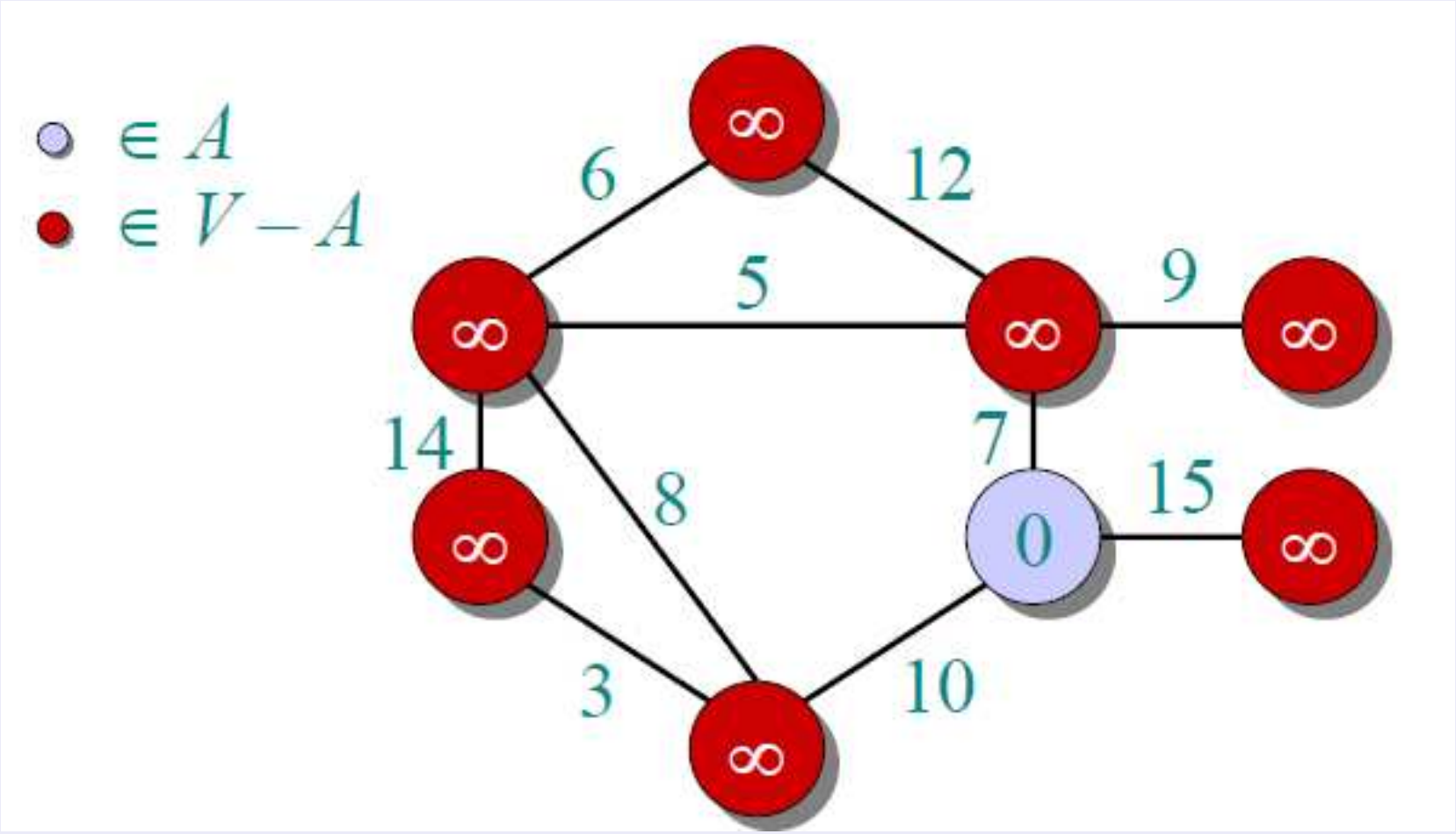
ΕΣΔ-PRIM(G, w, s)
$Q \leftarrow V$
$key[v] \leftarrow \infty$ for all $v \in V$
$key[s] \leftarrow 0$ for some arbitrary $s \in V$
`while` $Q \neq \varnothing$
    `do` $u \leftarrow$ EXTRACT-MIN(Q)
        `for each` $v \in Adj[u]$
            `do if` $v \in Q$ and $w(u, v) < key[v]$
                `then` $key[v] \leftarrow w(u, v)$ /*DECREASE-KEY */
                $\pi[v] \leftarrow u$

Το $\{(u, \pi[v])\}$ σχηματίζει το Ελαφρύτατο Συνδετικό Δέντρο

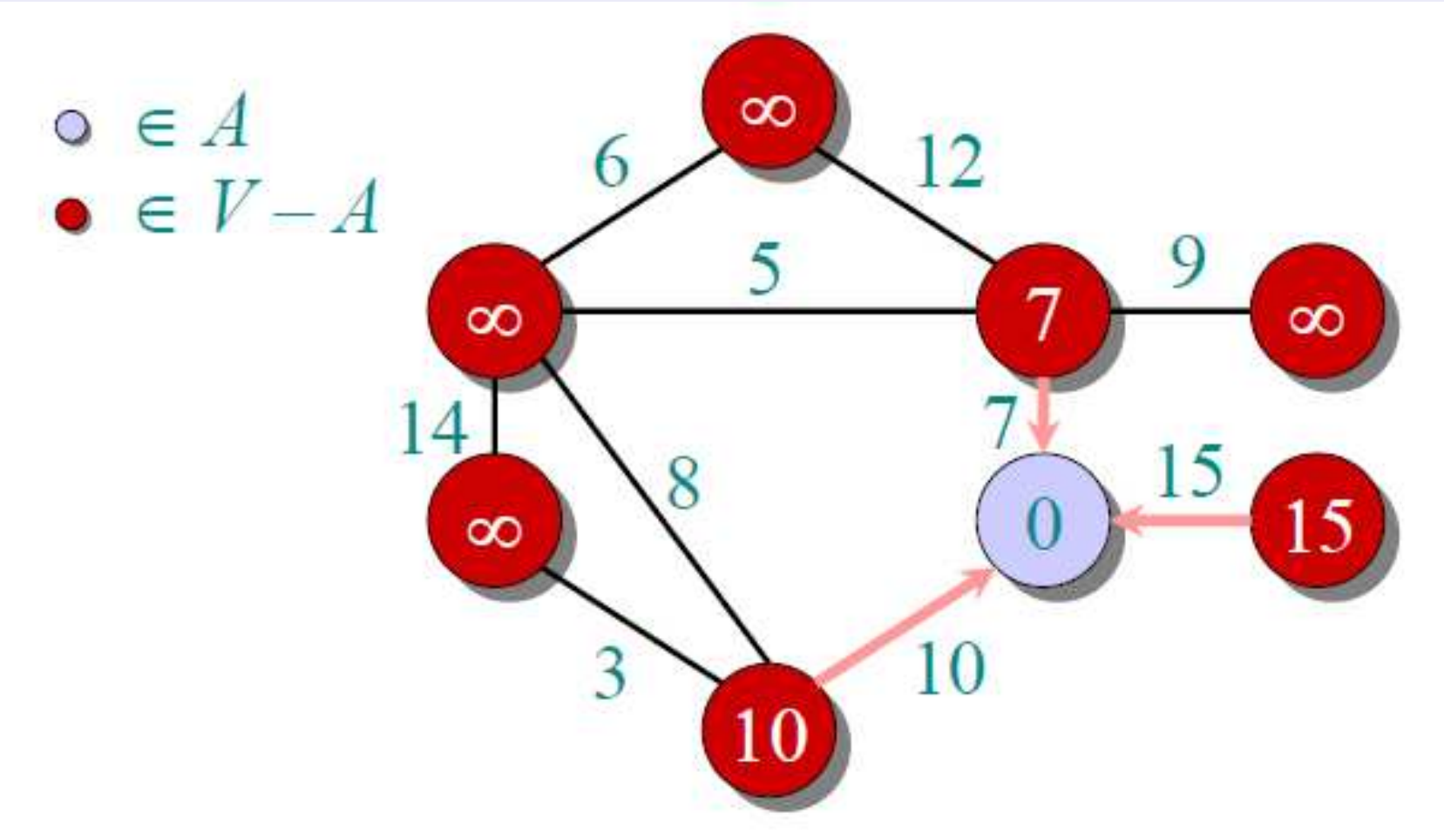
Παράδειγμα



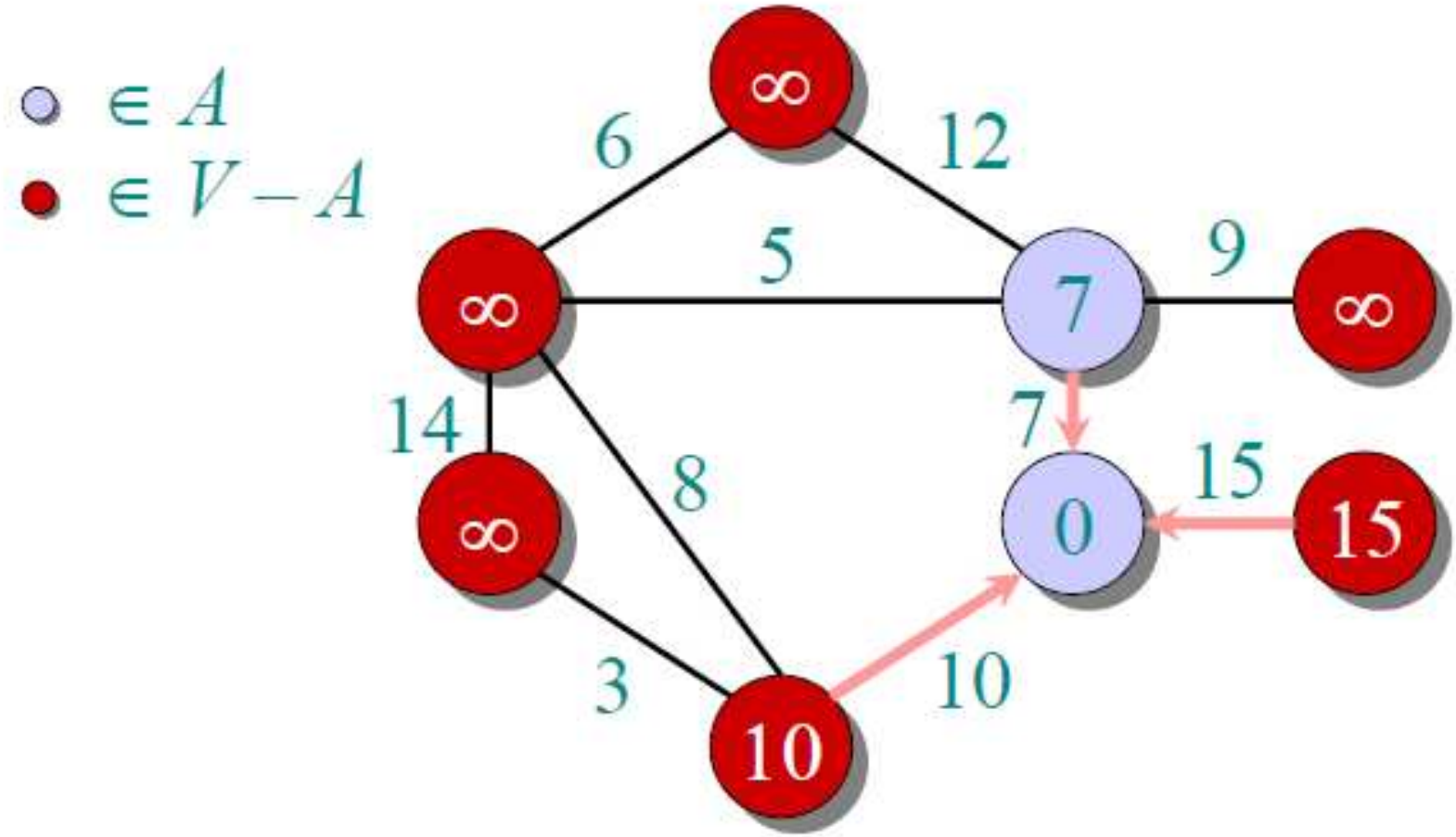
Παράδειγμα



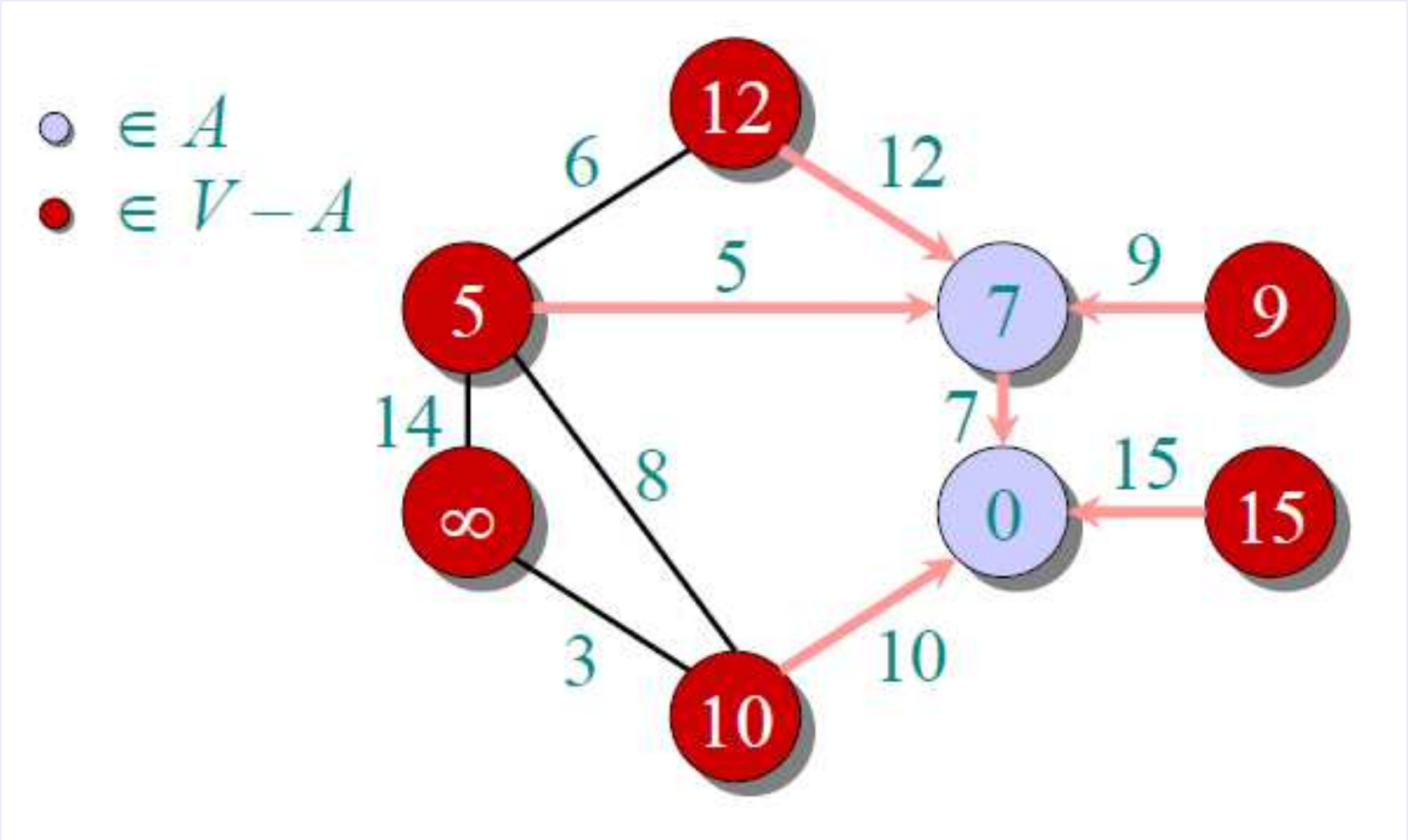
Παράδειγμα



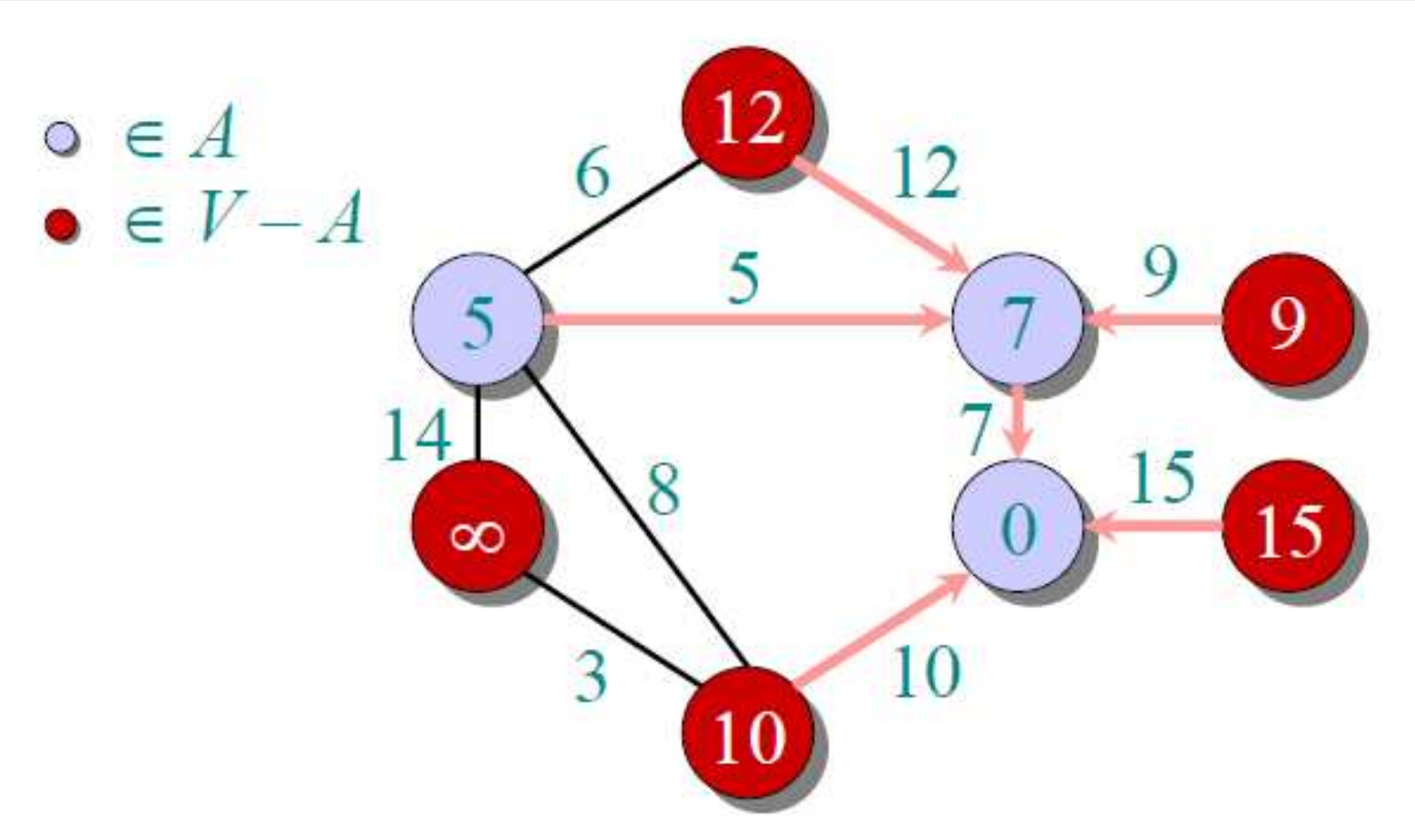
Παράδειγμα



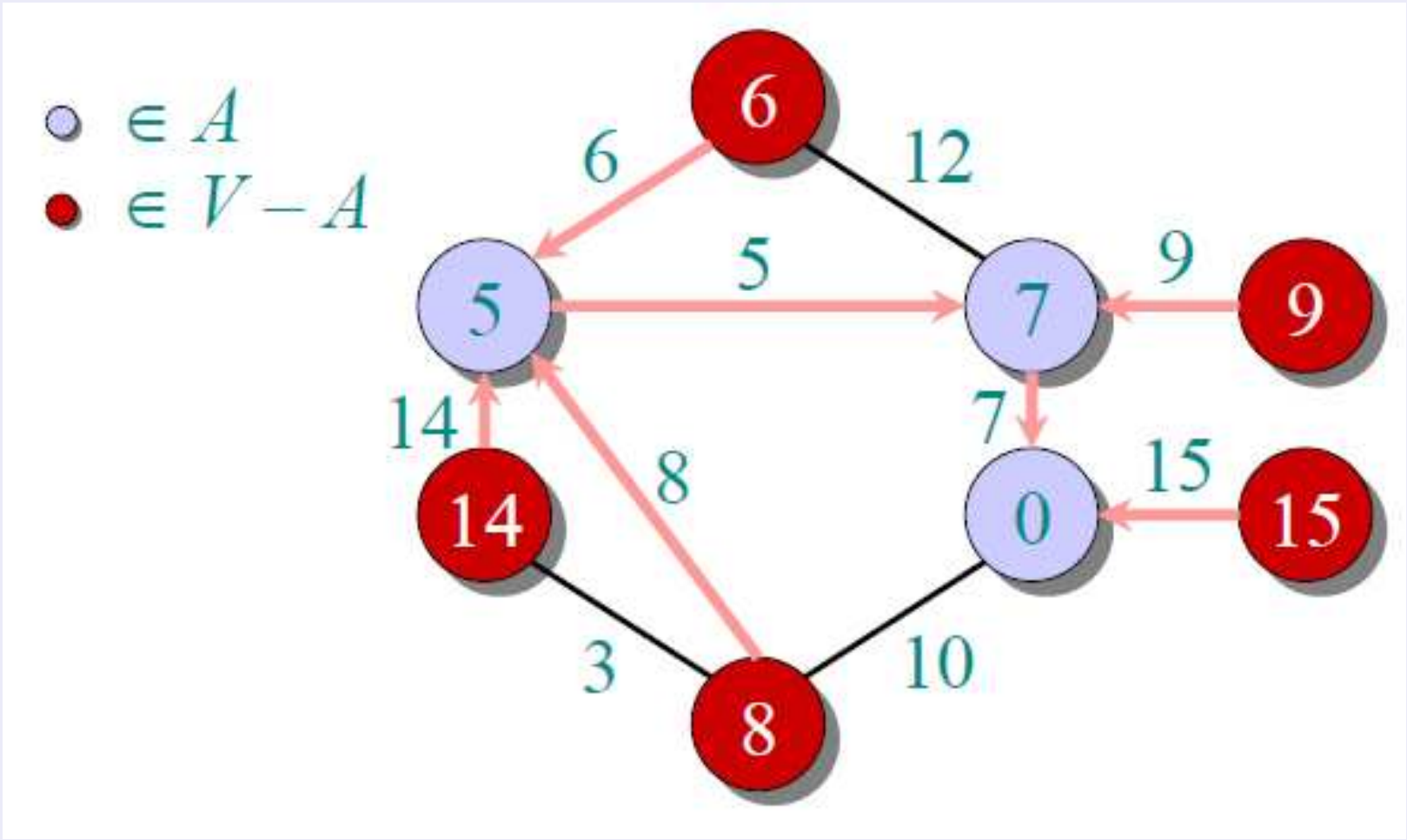
Παράδειγμα



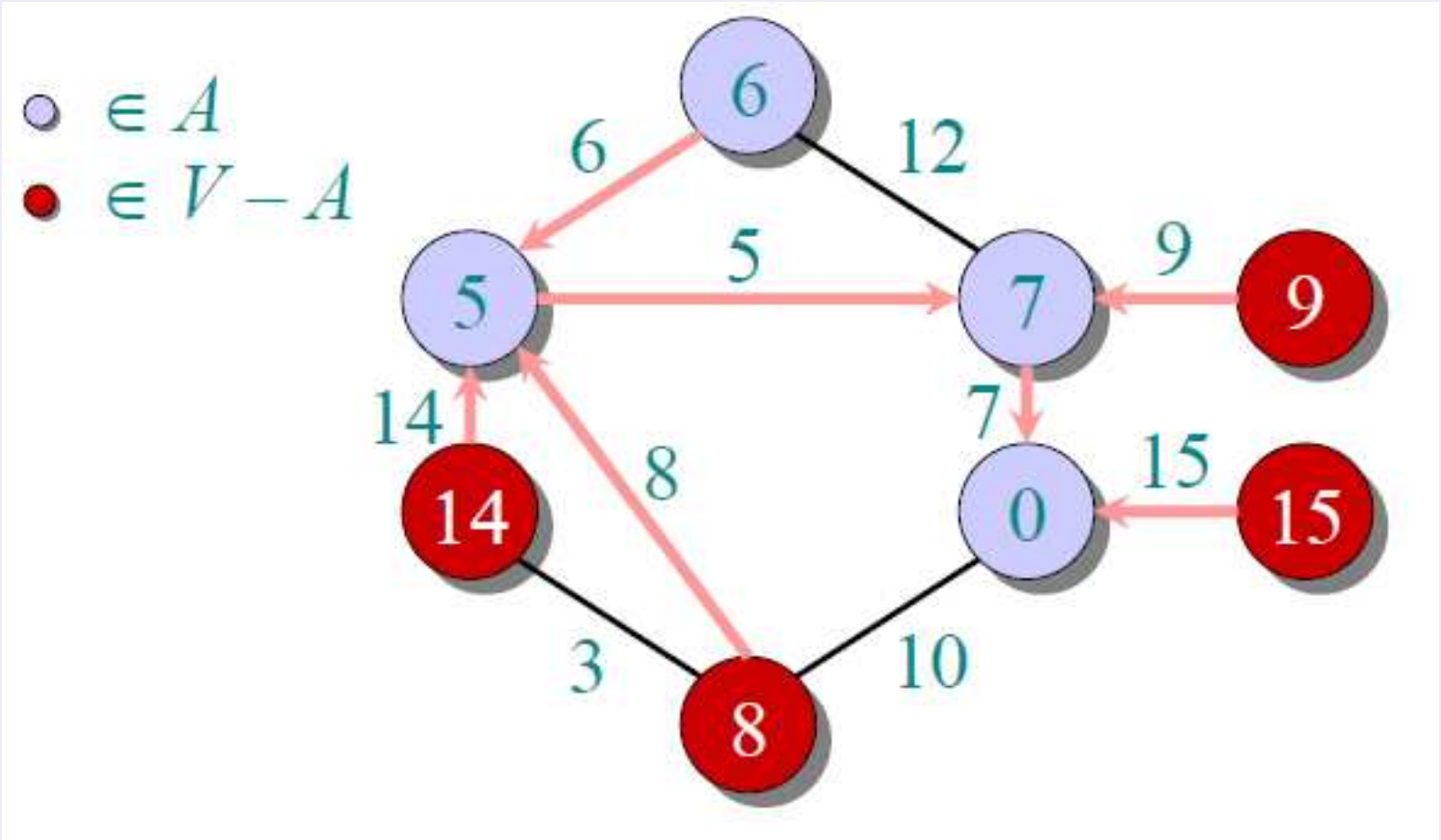
Παράδειγμα



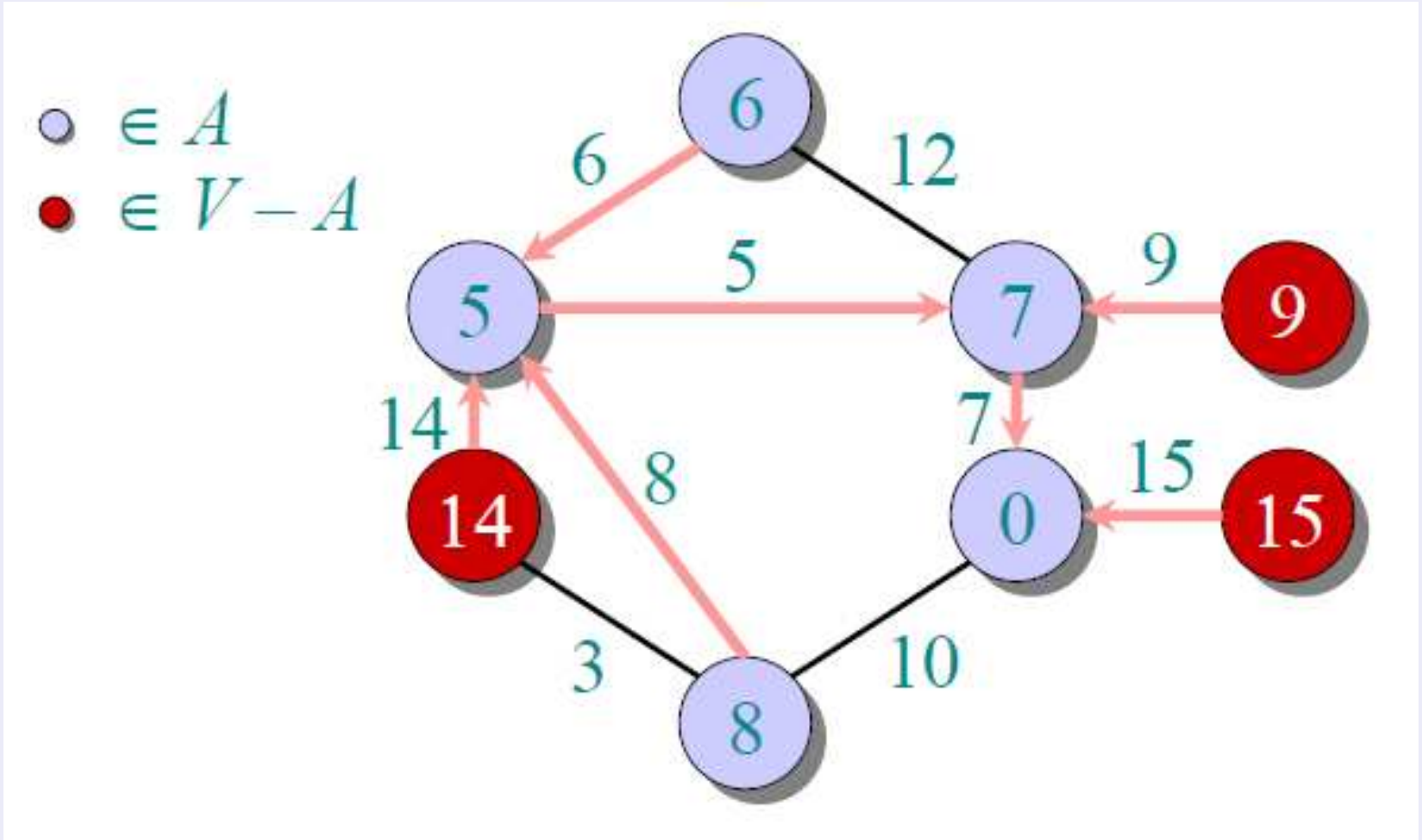
Παράδειγμα



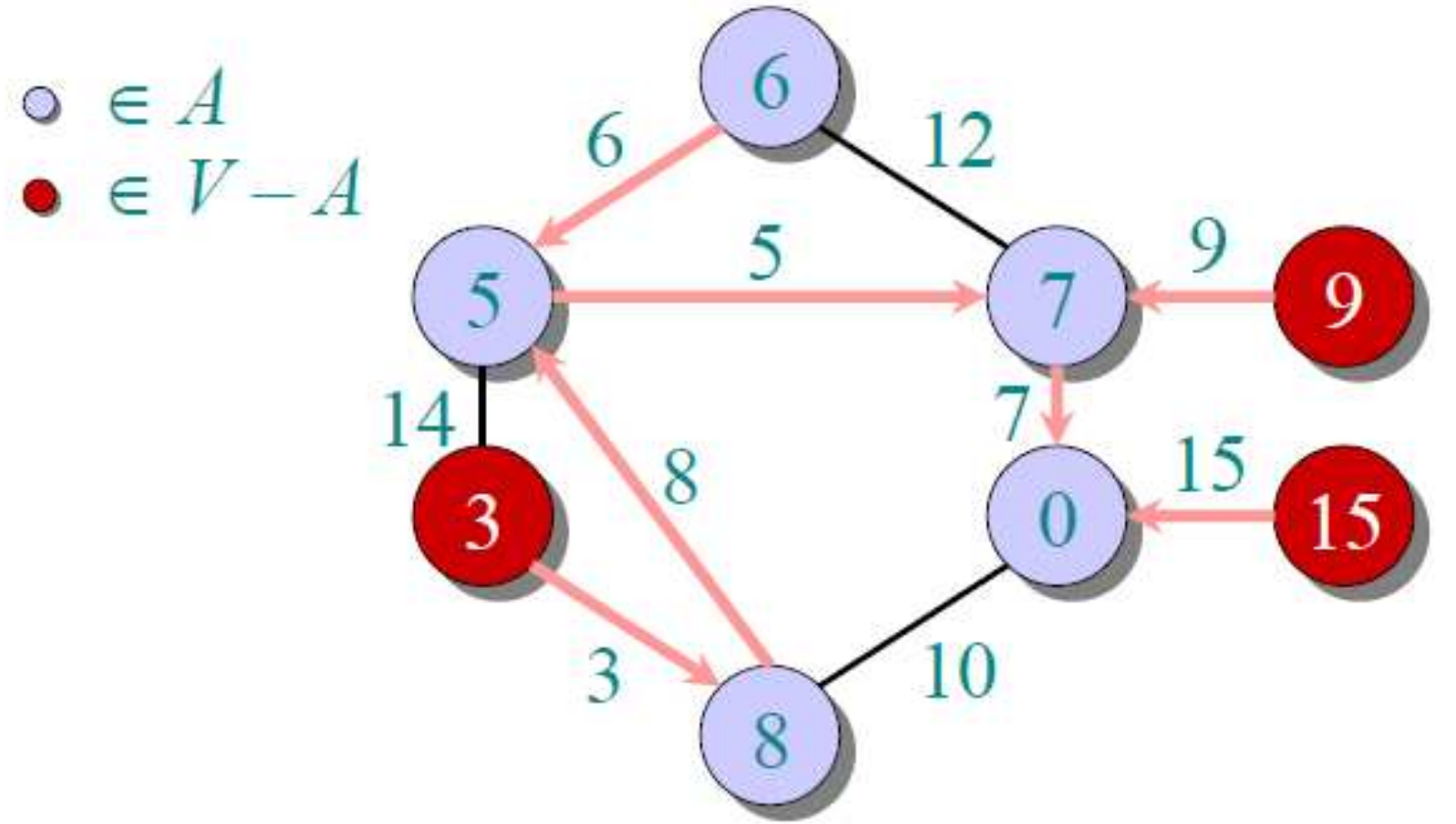
Παράδειγμα



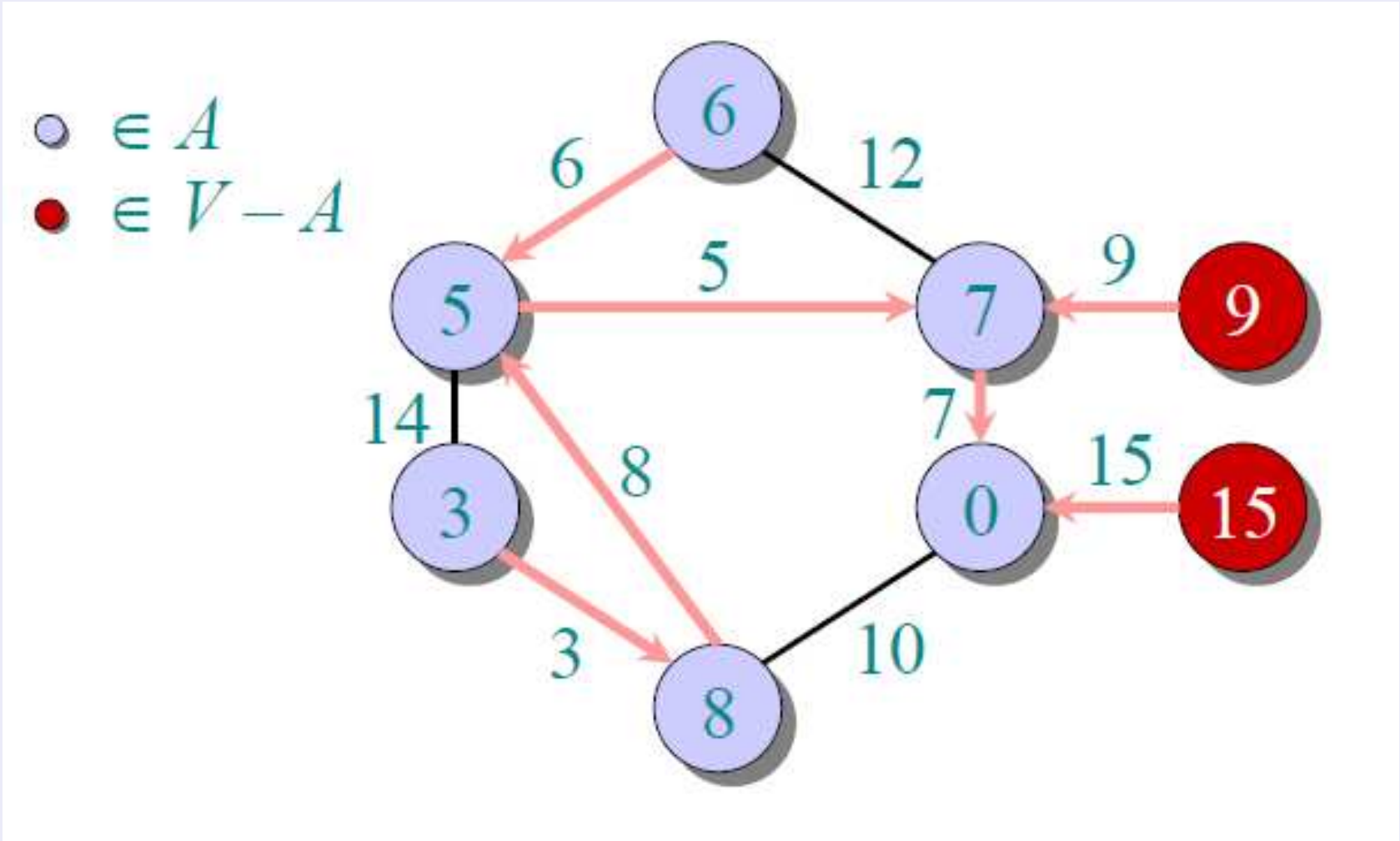
Παράδειγμα



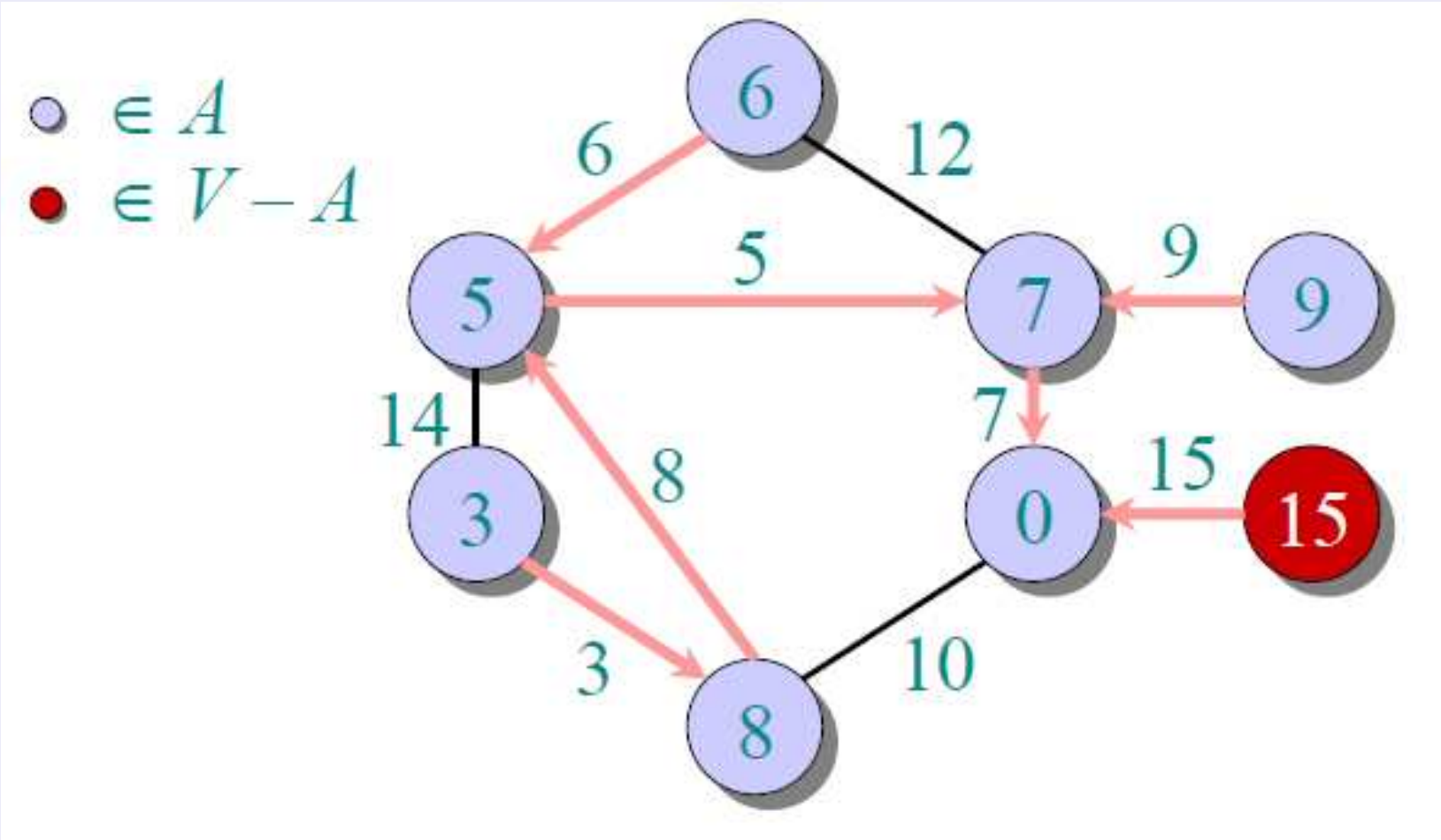
Παράδειγμα



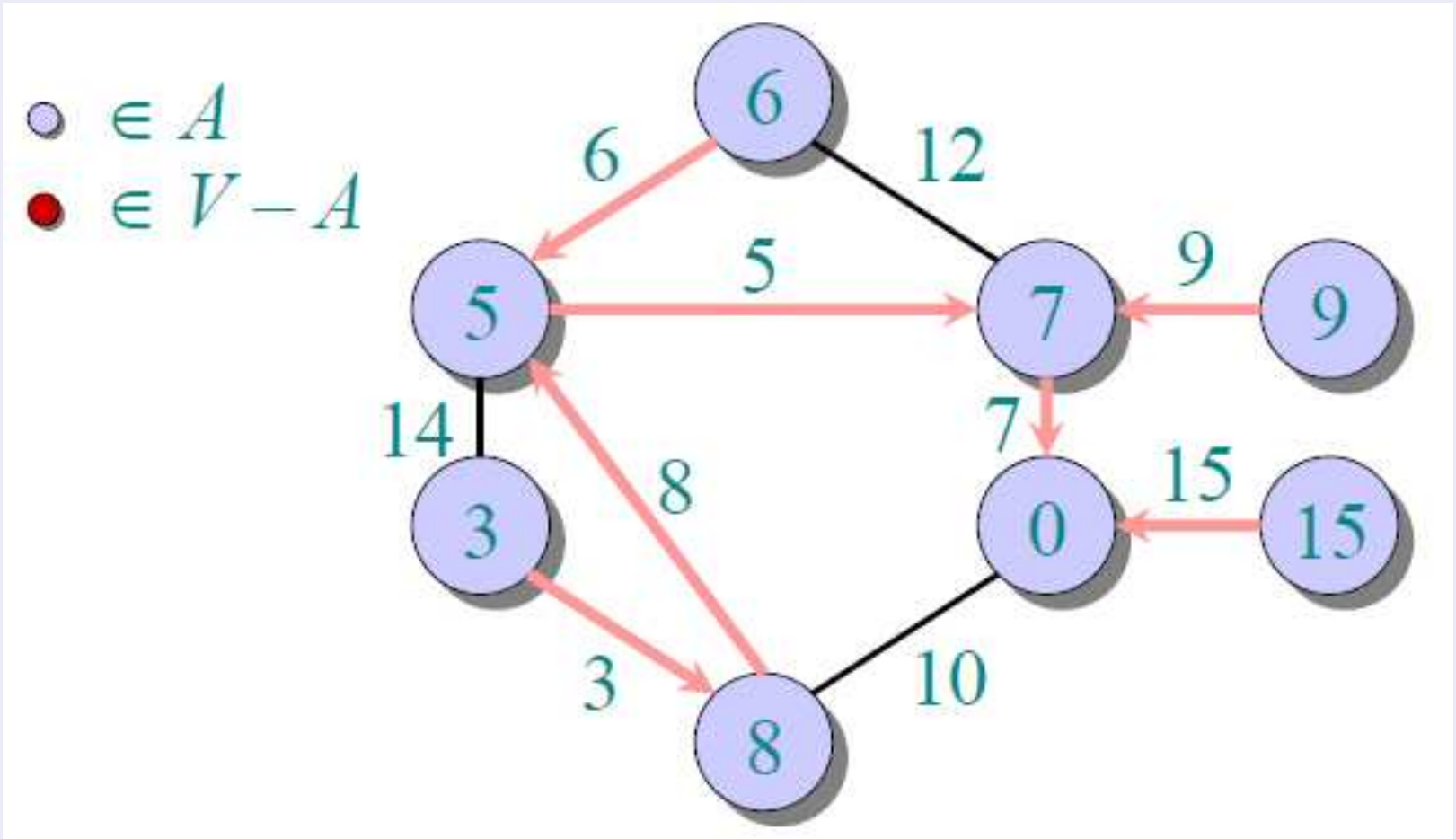
Παράδειγμα



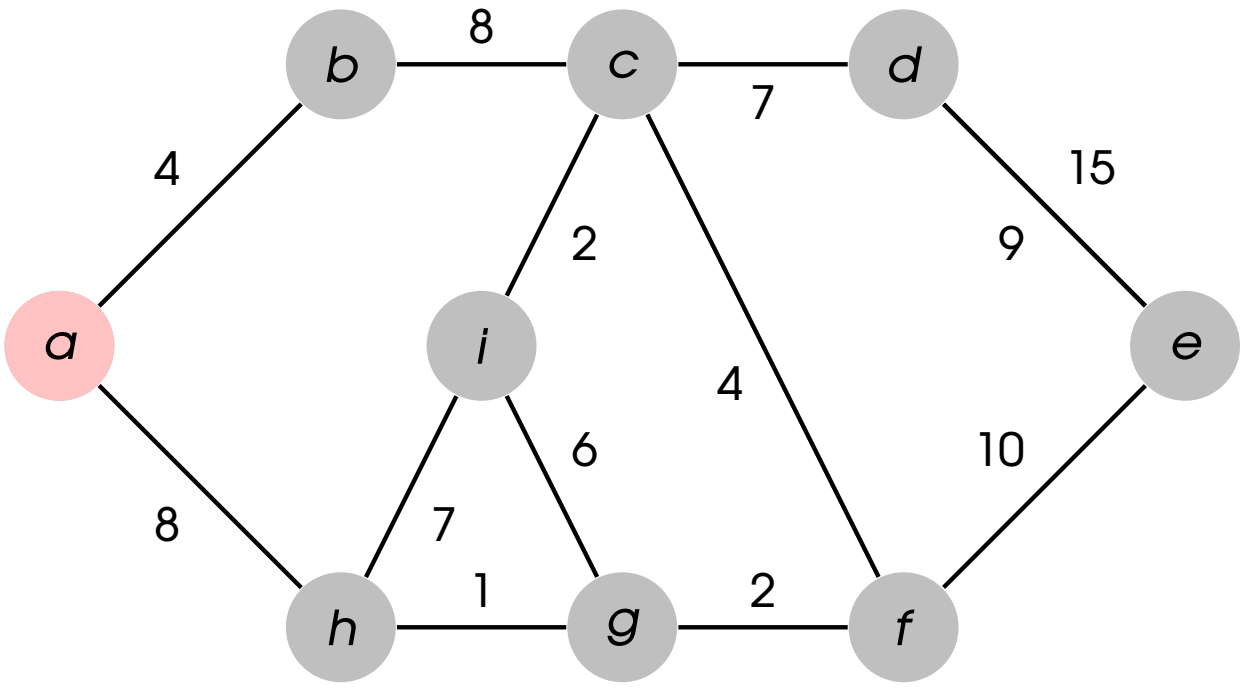
Παράδειγμα



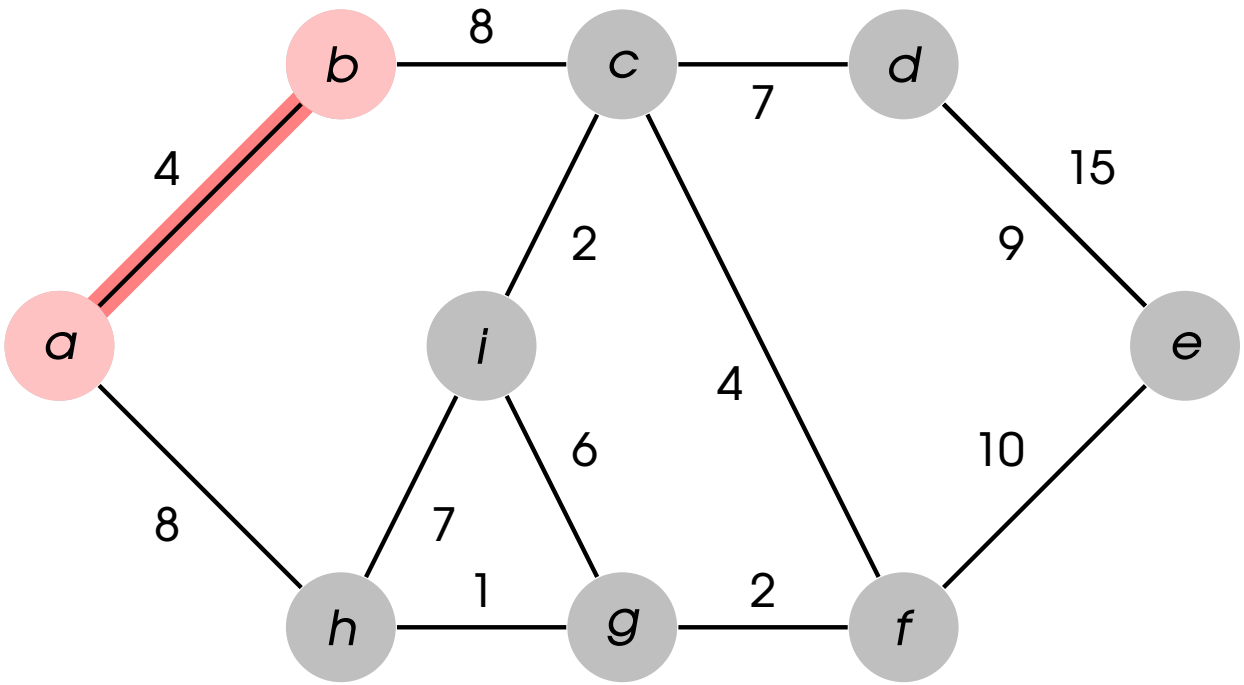
Παράδειγμα



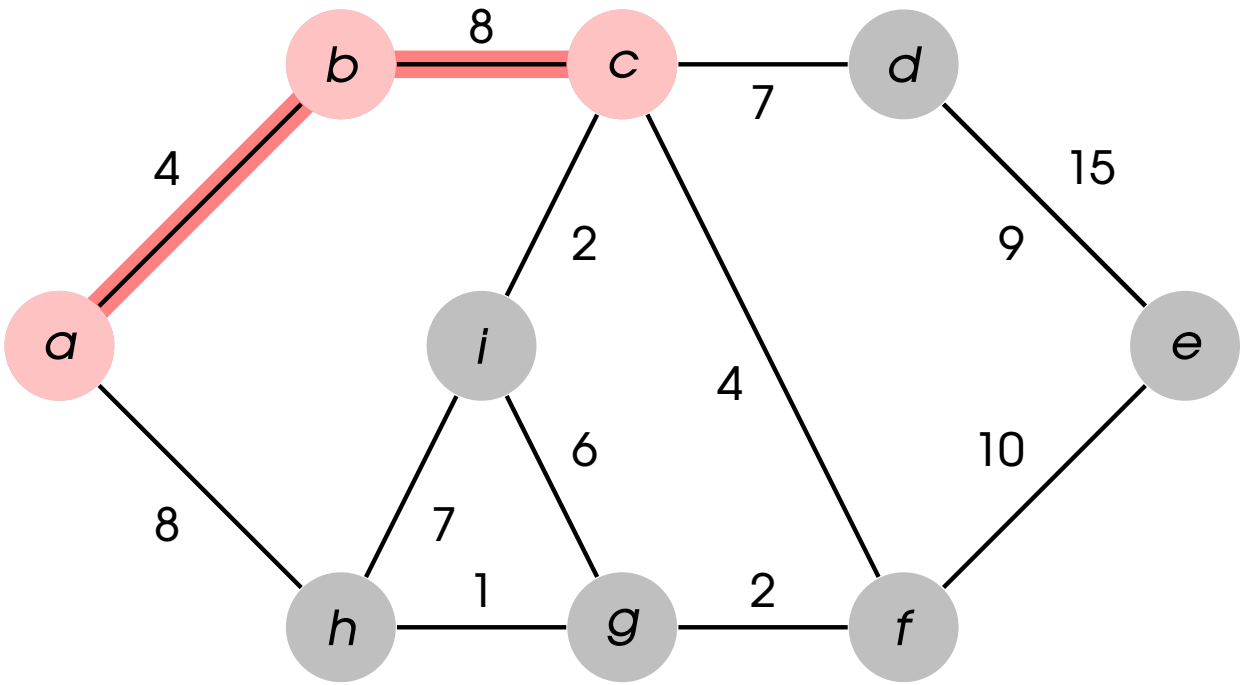
Prim



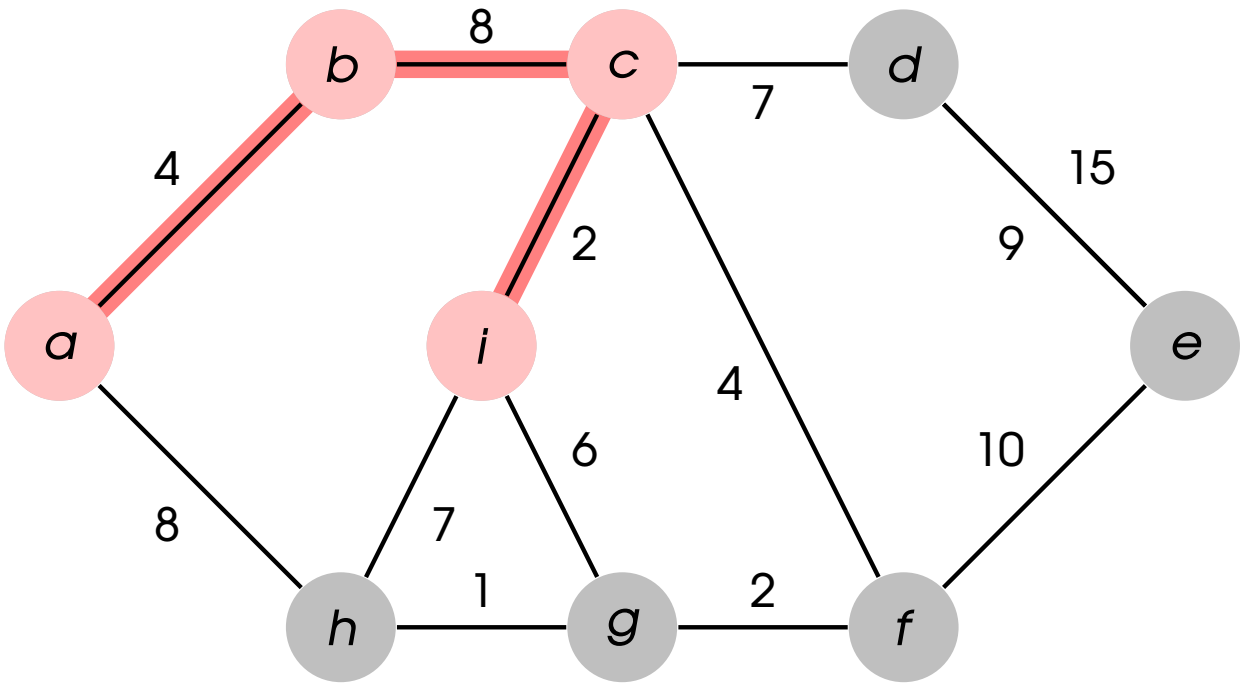
Prim



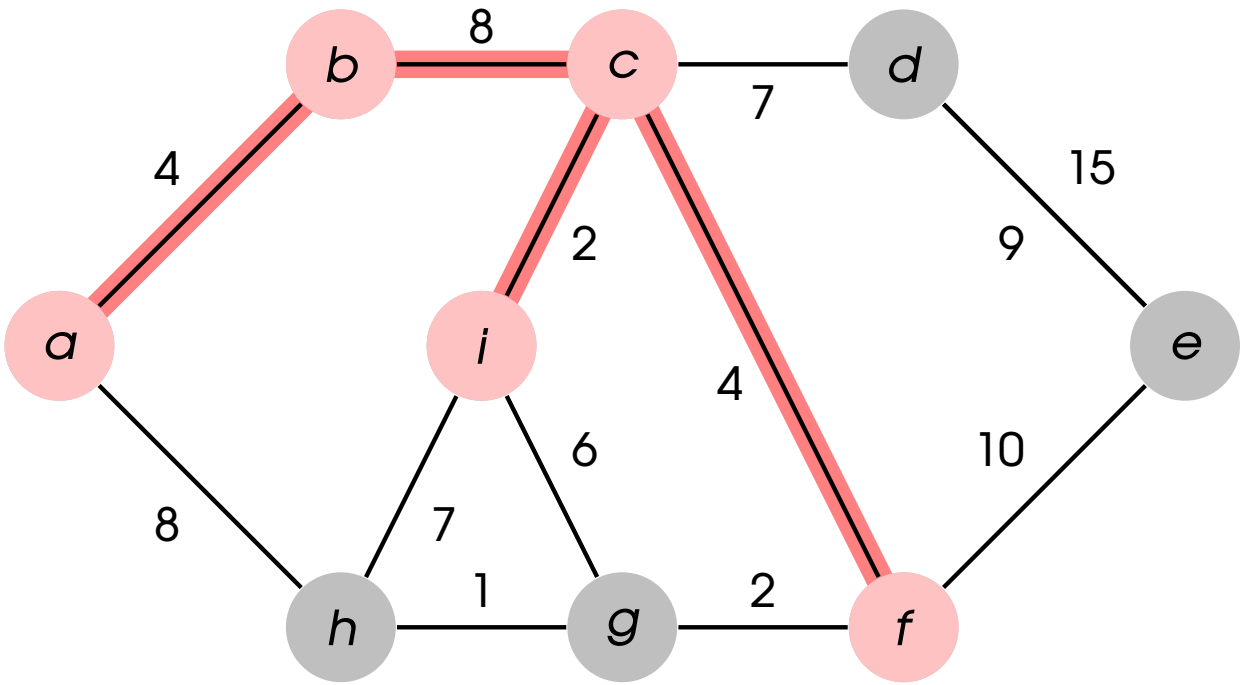
Prim



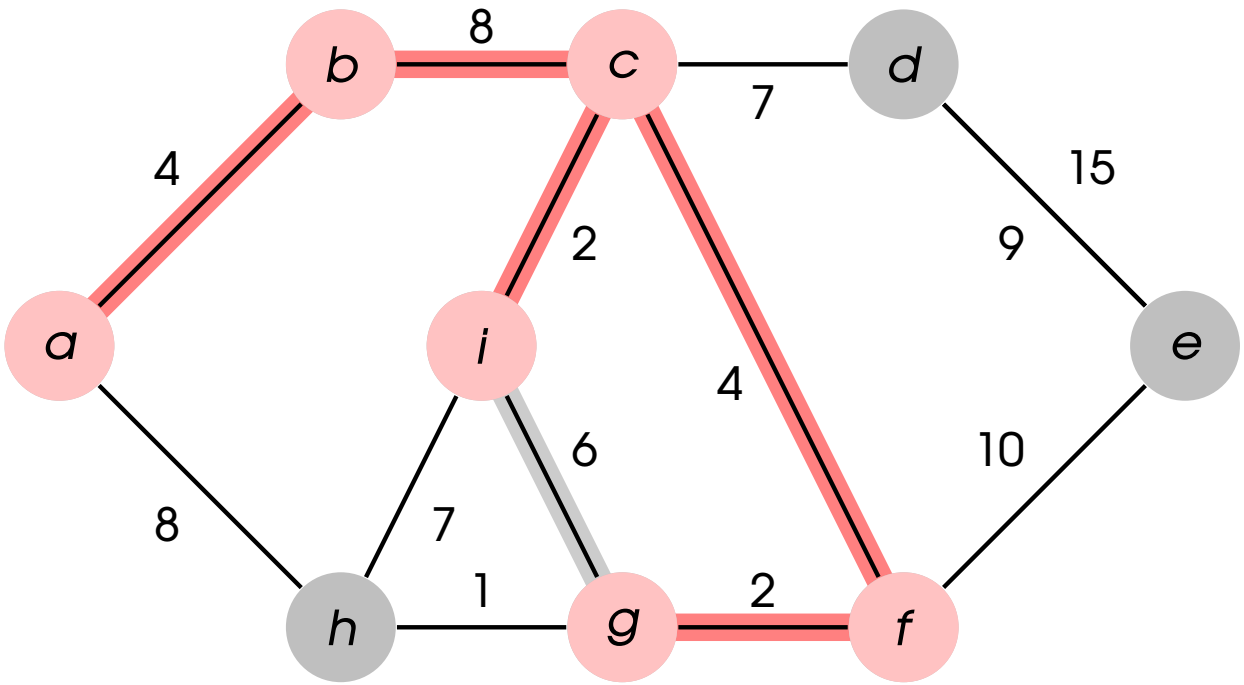
Prim



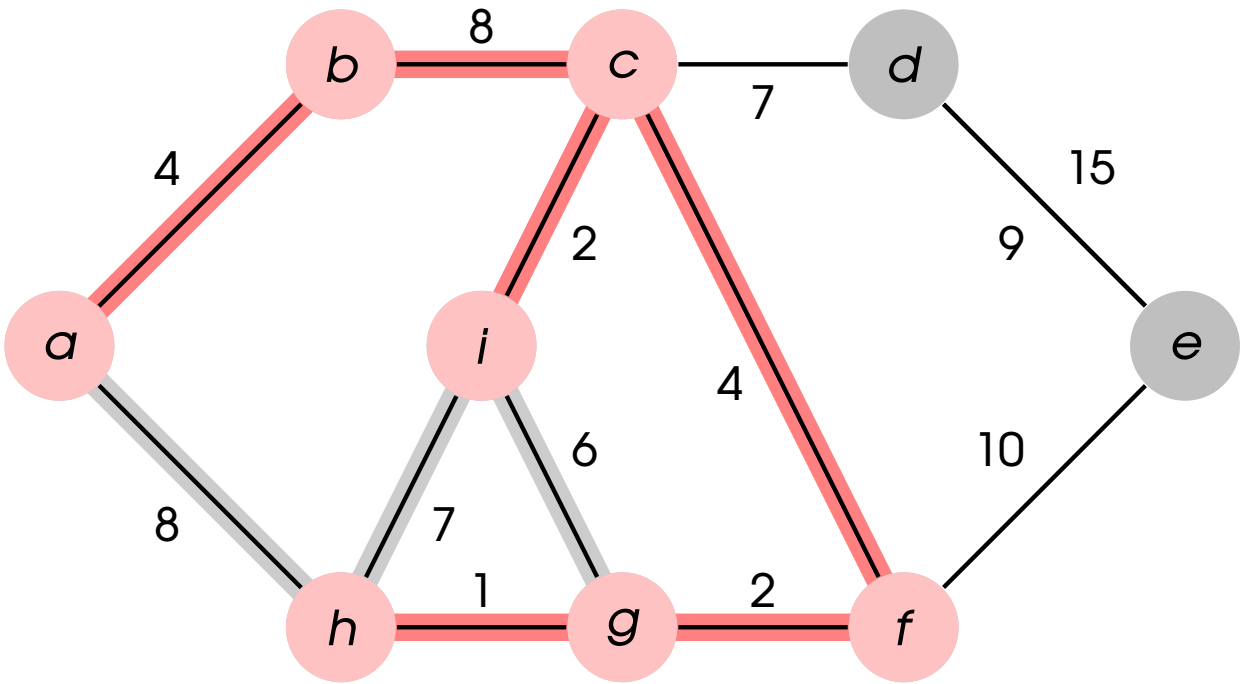
Prim



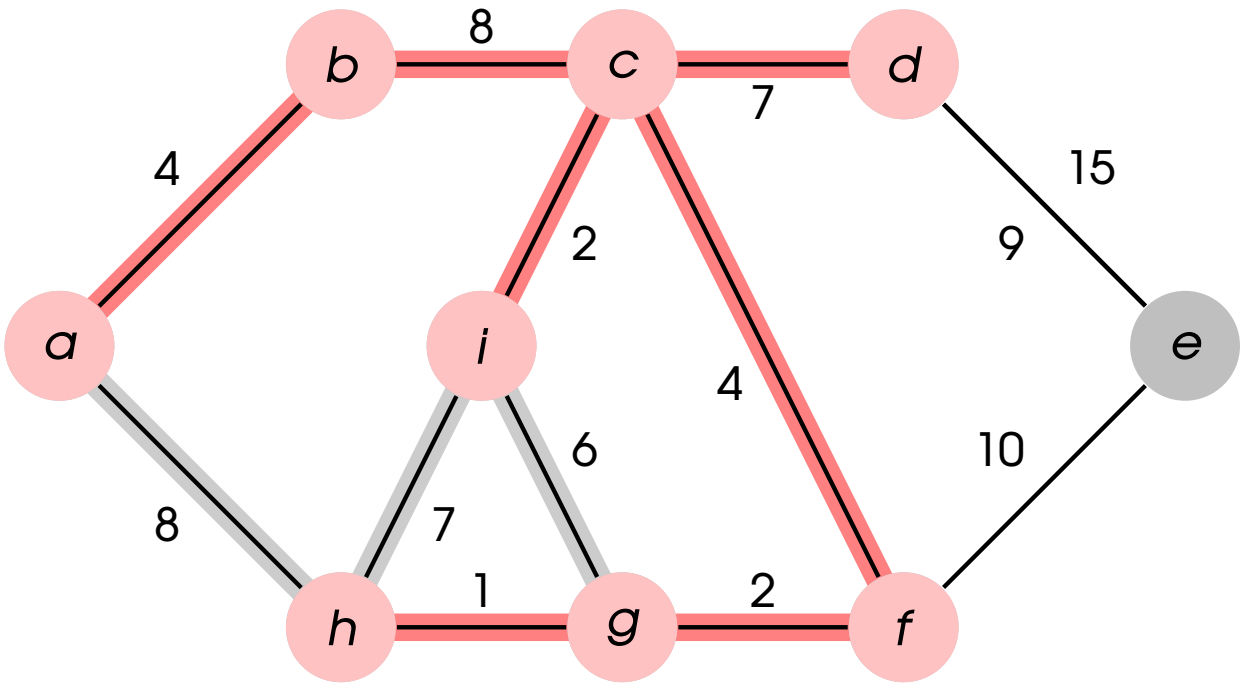
Prim



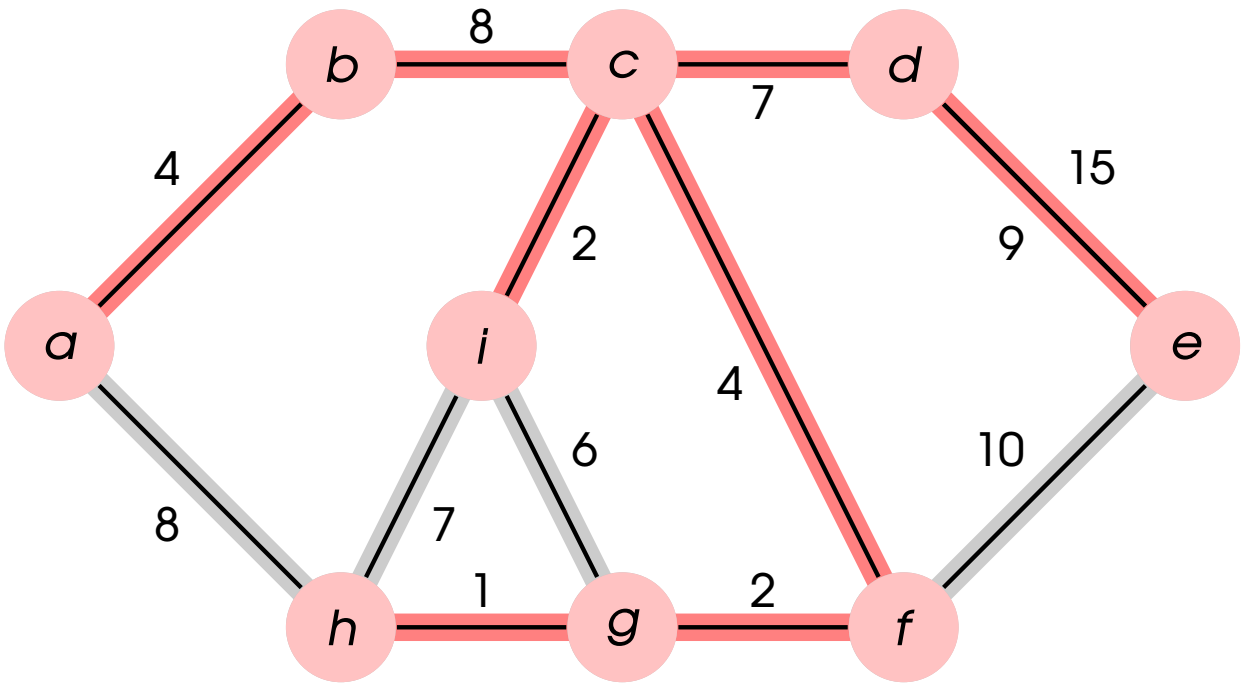
Prim



Prim



Prim



Ανάλυση Αλγορίθμου Prim

$$\underset{\text{total}}{\Theta(V)} \begin{cases} Q \leftarrow V \\ key[v] \leftarrow \infty \text{ for all } v \in V \\ key[s] \leftarrow 0 \text{ for some arbitrary } s \in V \end{cases}$$

$$\underset{\text{times}}{|V|} \begin{cases} \text{while } Q \neq \varnothing \\ \quad \text{do } u \leftarrow \textsc{Extract-Min}(Q) \\ \quad \underset{\text{times}}{\text{degree(u)}} \begin{cases} \text{for each } v \in Adj[u] \\ \quad \text{do if } v \in Q \text{ and } w(u,v) < key[v] \\ \quad\quad \text{then } key[v] \leftarrow w(u,v) \quad \Theta(E) \\ \quad\quad\quad \pi[v] \leftarrow u \end{cases} \end{cases}$$

Συνολικός Χρόνος:

$$T = \Theta(V) \cdot T_{\textsc{Extract-Min}} + \Theta(E) \cdot T_{\textsc{Decrease-Key}}$$

Ανάλυση Αλγορίθμου Prim (συνέχεια)

$$T = \Theta(V) \cdot T_{\text{EXTRACT-MIN}} + \Theta(E) \cdot T_{\text{DECREASE-KEY}}$$

Q	$T_{\text{EXTRACT-MIN}}$	$T_{\text{DECREASE-KEY}}$	Συνολικό
array	$O(V)$	$O(1)$	$O(V^2)$
binary Heap	$O(\log V)$	$O(\log V)$	$O(E \log V)$
fibonacci Heap	$O(\log V)$	$O(1)$	$O(E + V \log V)$

Αλγόριθμοι MST

- Αλγόριθμος του Kruskal
  - χρησιμοποιεί τις συνεκτικές συνιστώσες
  - έχει πολυπλοκότητα $O(E \log V)$
- Karger, Klein, Tarjan (1993)
  - πιθανοτικός αλγόριθμος
  - αναμενόμενος χρόνος $O(V + E)$

Ο αλγόριθμος του Kruskal

Στον αλγόριθμο του Kruskal

- Το σύνολο A είναι ένα δάσος.
- Η ασφαλής ακμή που προστίθεται στο A είναι πάντοτε μια ακμή ελαχίστου βάρους στο γράφημα, η οποία συνδέει δυο διαφορετικές συνιστώσες.

Στον αλγόριθμο του Prim

- Το σύνολο A συνιστά ένα μοναδικό δένδρο.
- Η ασφαλής ακμή που προστίθεται στο A είναι πάντοτε μια ακμή ελαχίστου βάρους η οποία συνδέει το δένδρο με έναν κόμβο εκτός του δένδρου.

Ο αλγόριθμος του Kruskal

Ο αλγόριθμος του Kruskal

- Προσδιορίζει την ασφαλή ακμή που θα προστεθεί στο επεκτεινόμενο δάσος εντοπίζοντας από όλες τις ακμές που συνδέουν δυο οποιαδήποτε δένδρα του δάσους, μια ακμή (u, v) ελαχίστου βάρους.
- Ανήκει στην κατηγορία των άπληστων αλγορίθμων, διότι σε κάθε βήμα προσθέτει στο δάσος μια ακμή με το ελάχιστο δυνατό βάρος.

Ο αλγόριθμος του Kruskal

ΕΣΔ-Kruskal

Η υλοποίησή μας είναι παρόμοια με τον αλγόριθμο για τον υπολογισμό των συνδεδεμένων συνιστωσών. Βασίζεται στην τήρηση κάποιων ξένων συνόλων από στοιχεία μέσω μιας δομής δεδομένων ξένων συνόλων.

- Κάθε τηρούμενο σύνολο περιέχει τους κόμβους ενός δένδρου του τρέχοντος δάσους.
- Η πράξη Ευρεση Συνολου(u) επιστρέφει ένα στοιχείο-αντιπρόσωπο του συνόλου στο οποίο ανήκει ο u.
- Επομένως, μπορούμε να προσδιορίσουμε, εάν δυο κόμβοι u και v ανήκουν στο ίδιο δένδρο ελέγχοντας εάν το στοιχείο Ευρεση Συνολου(u) συμπίπτει με το Ευρεση Συνολου(v).
- Η σύζευξη των δένδρων πραγματοποιείται μέσω της πράξης Συνενωση.

Ο αλγόριθμος του Kruskal

ΕΣΔ-Kruskal

ΕΣΔ-Kruskal(G, w)

1 $A \leftarrow \varnothing$

2 **για** κάθε κόμβο $v \in V[G]$

3 Κατασκευη Συνολου$(v) \rightarrow O(v)$

4 ταξινομούμε τις ακμές E κατά μη φθίνουσα σειρά ως προς το βάρος w $\rightarrow O(ElogE)$

5 **για** κάθε ακμή $(u, v) \in E$, κατά μη φθίνουσα σειρά ως προς το βάρος

6 **αν** Ευρεση Συνολου$(u) \neq$ Ευρεση Συνολου$(v) \rightarrow O(logV)$

7 Ε φορες **τότε** $A \leftarrow A \cup \{(u, v)\} \rightarrow O(1)$

8 Συνενωση(u, v)

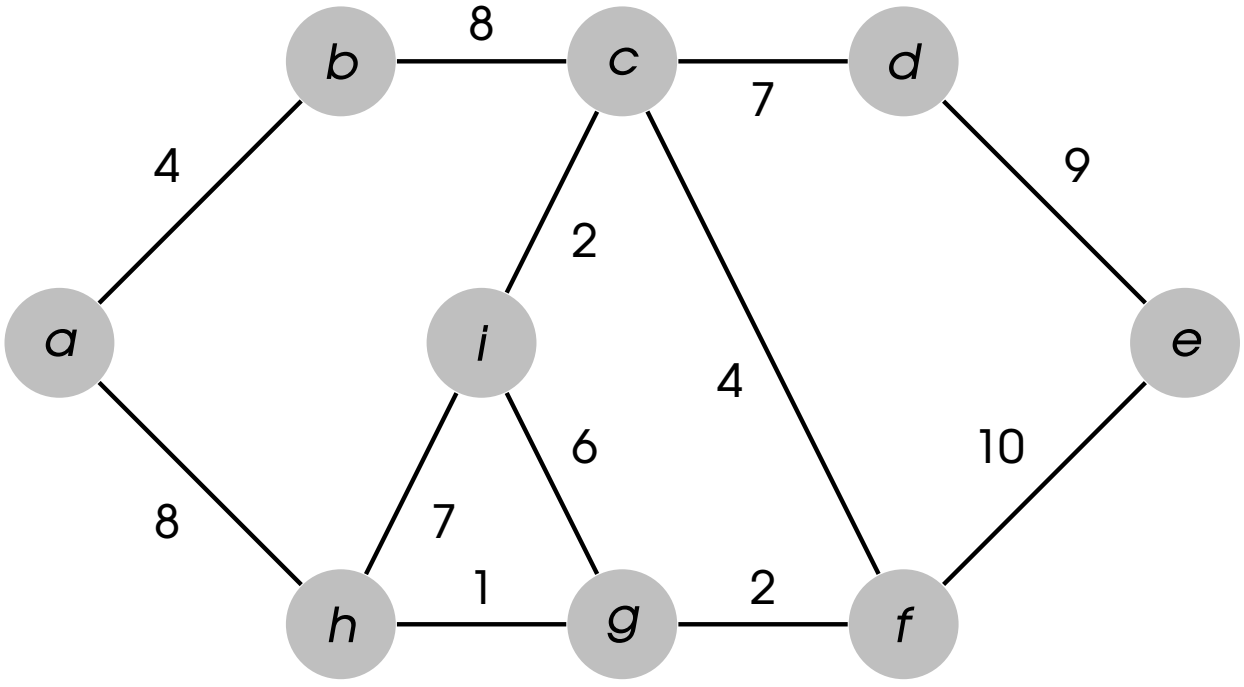
9 επιστροφή A

Πολυπλοκότητα

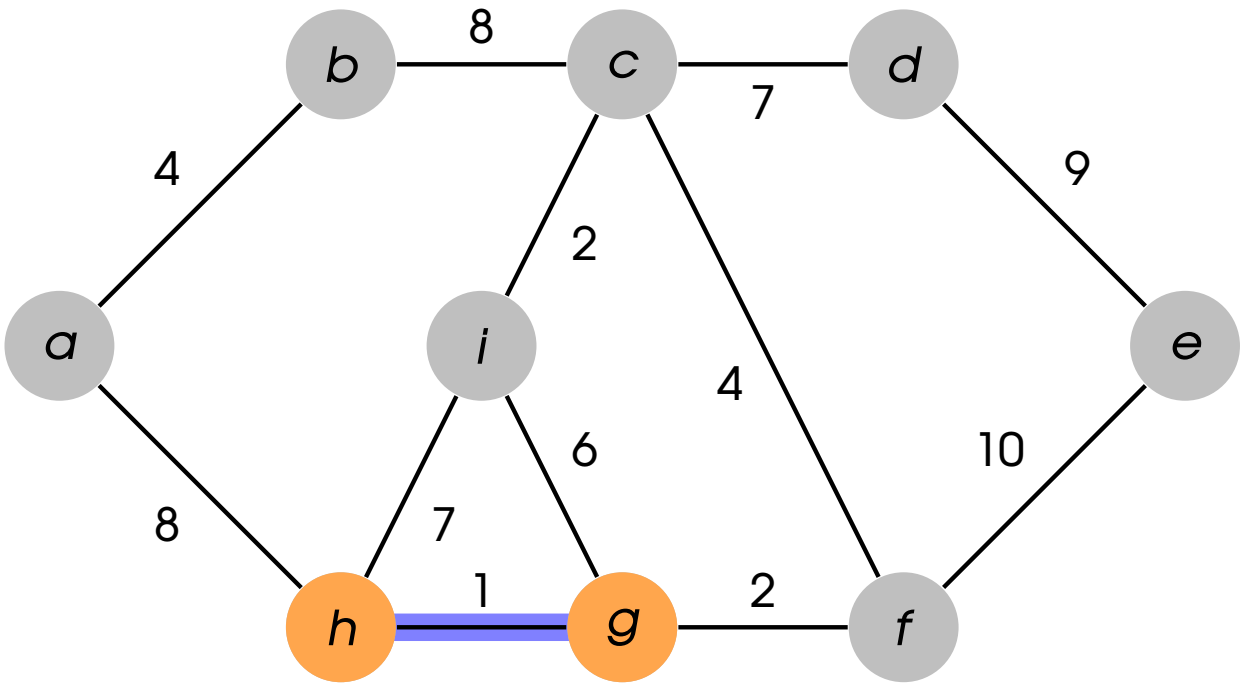
$$O(V + E \log E + E \log V) = O(E \log V)$$

$$\log E \simeq \log V.$$

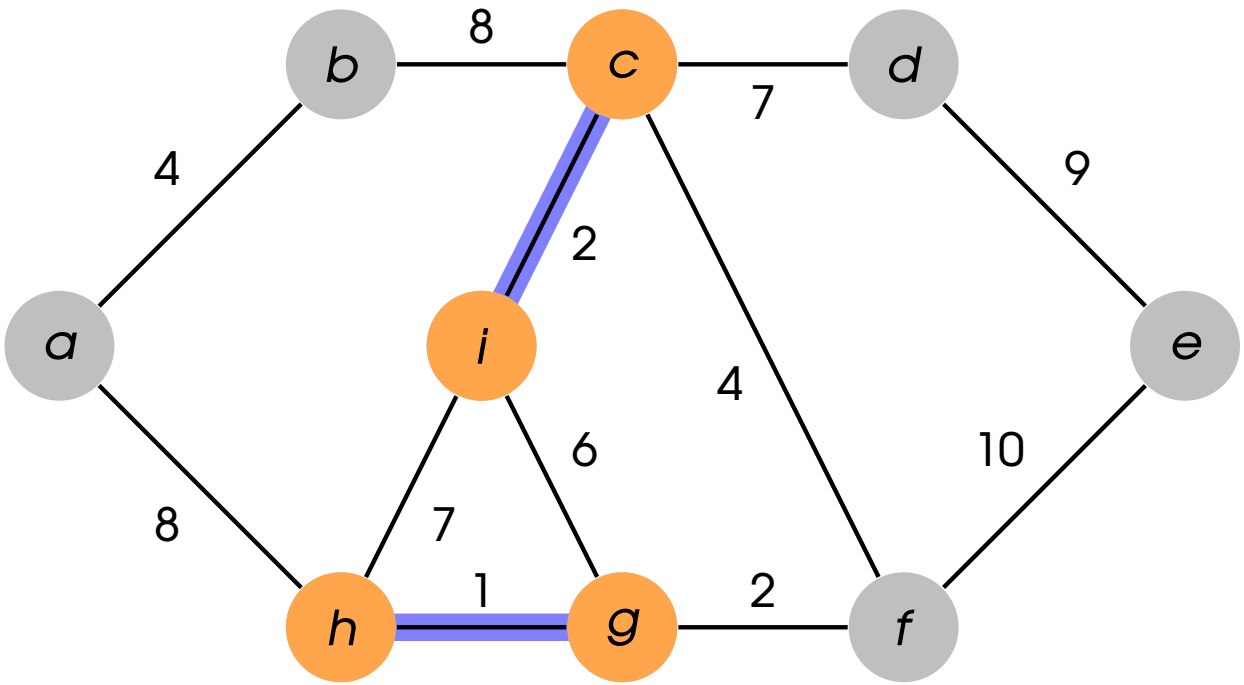
Ο αλγόριθμος του Kruskal



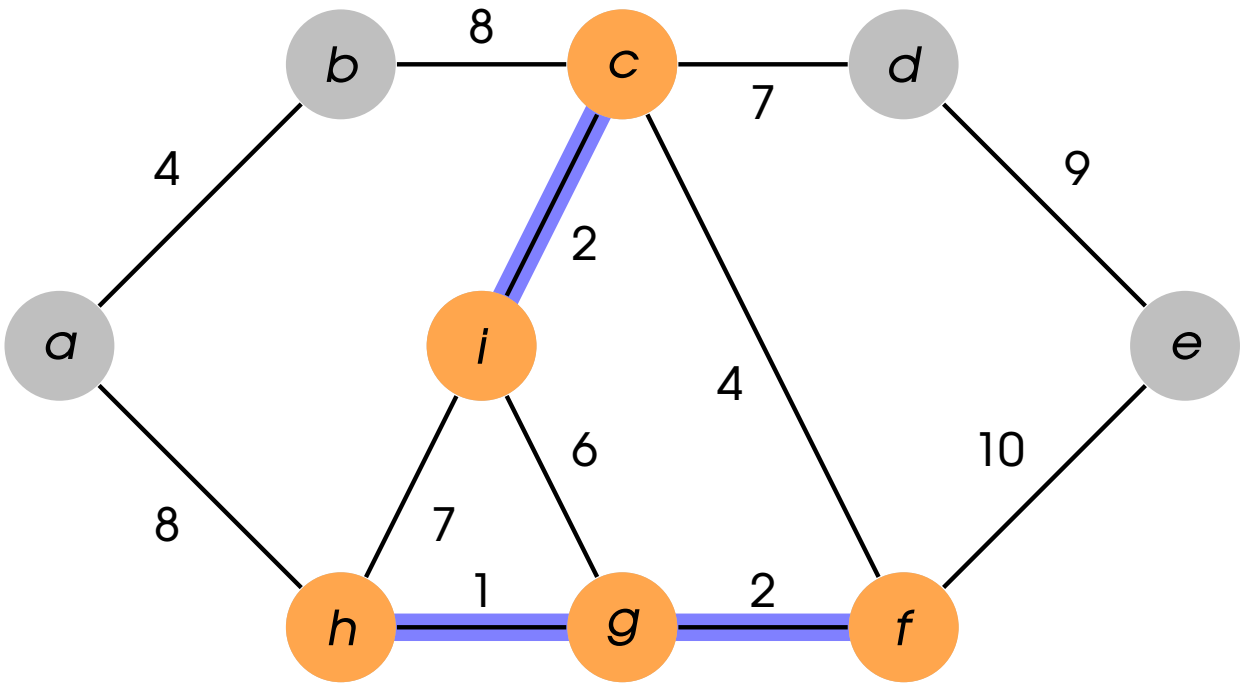
Ο αλγόριθμος του Kruskal



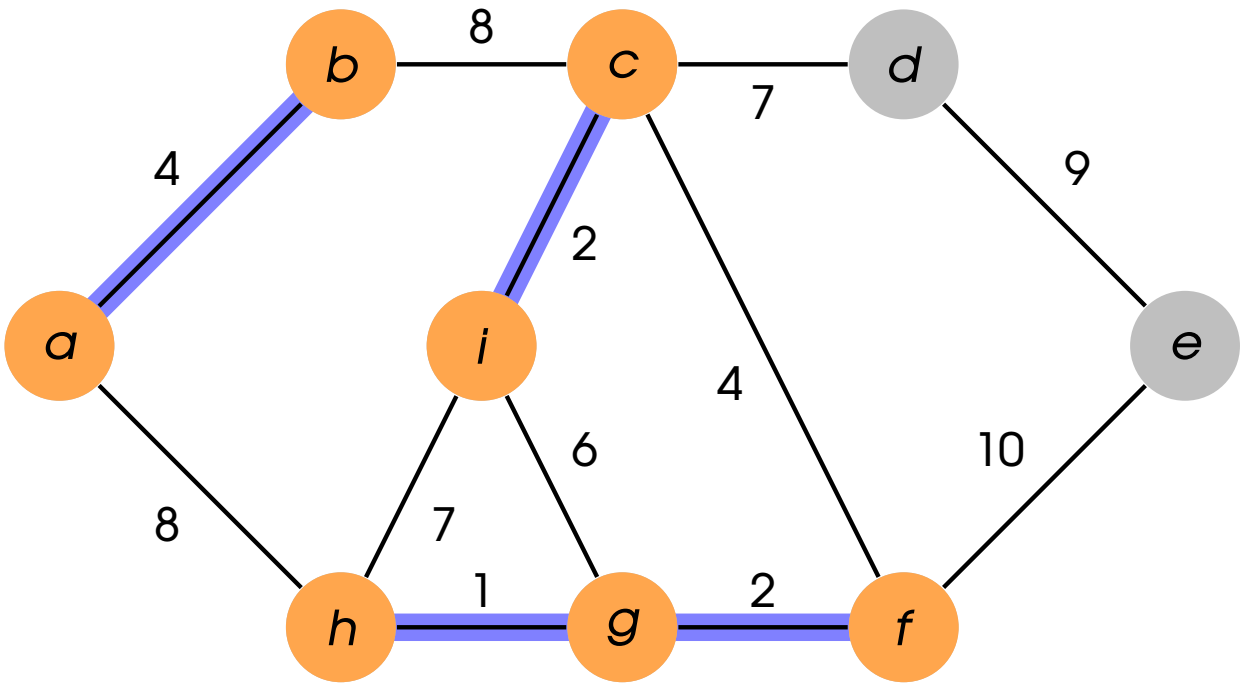
Ο αλγόριθμος του Kruskal



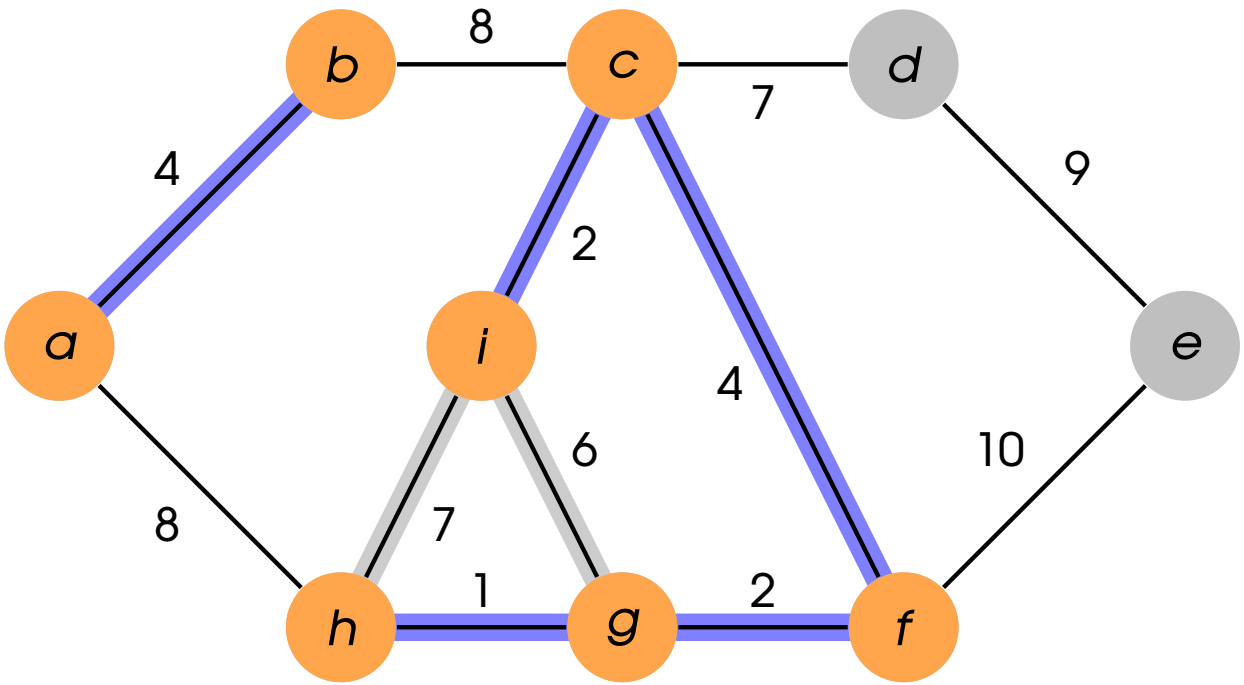
Ο αλγόριθμος του Kruskal



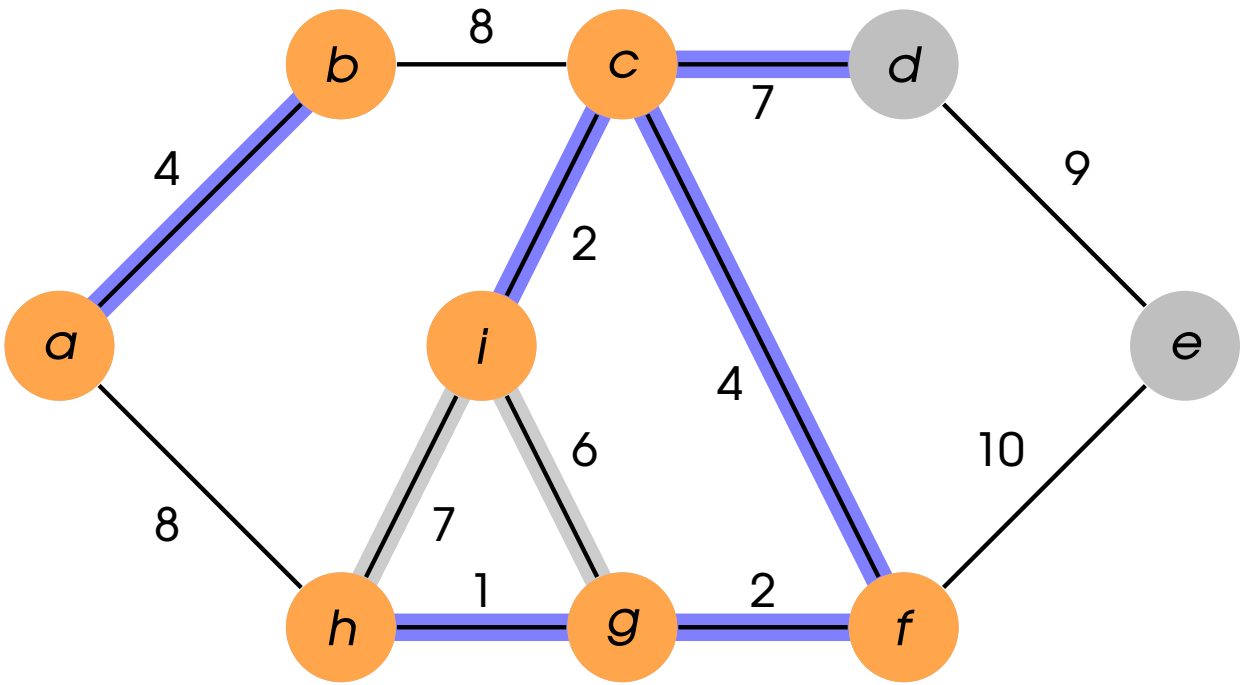
Ο αλγόριθμος του Kruskal



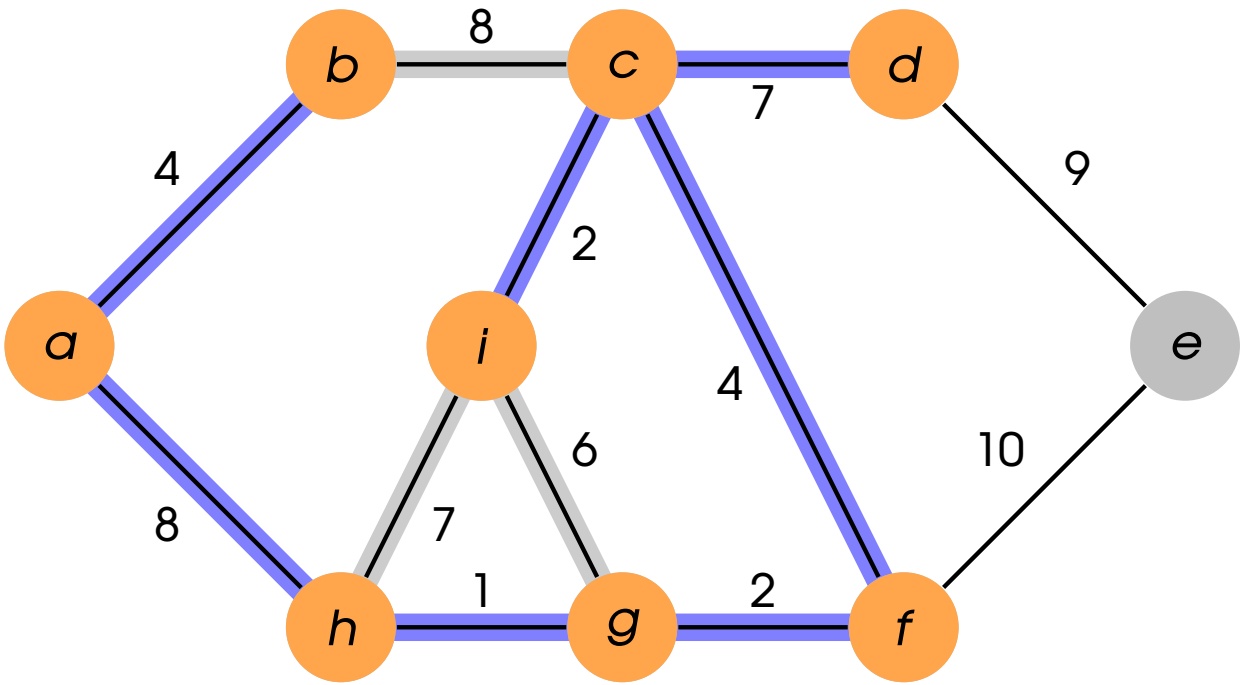
Ο αλγόριθμος του Kruskal



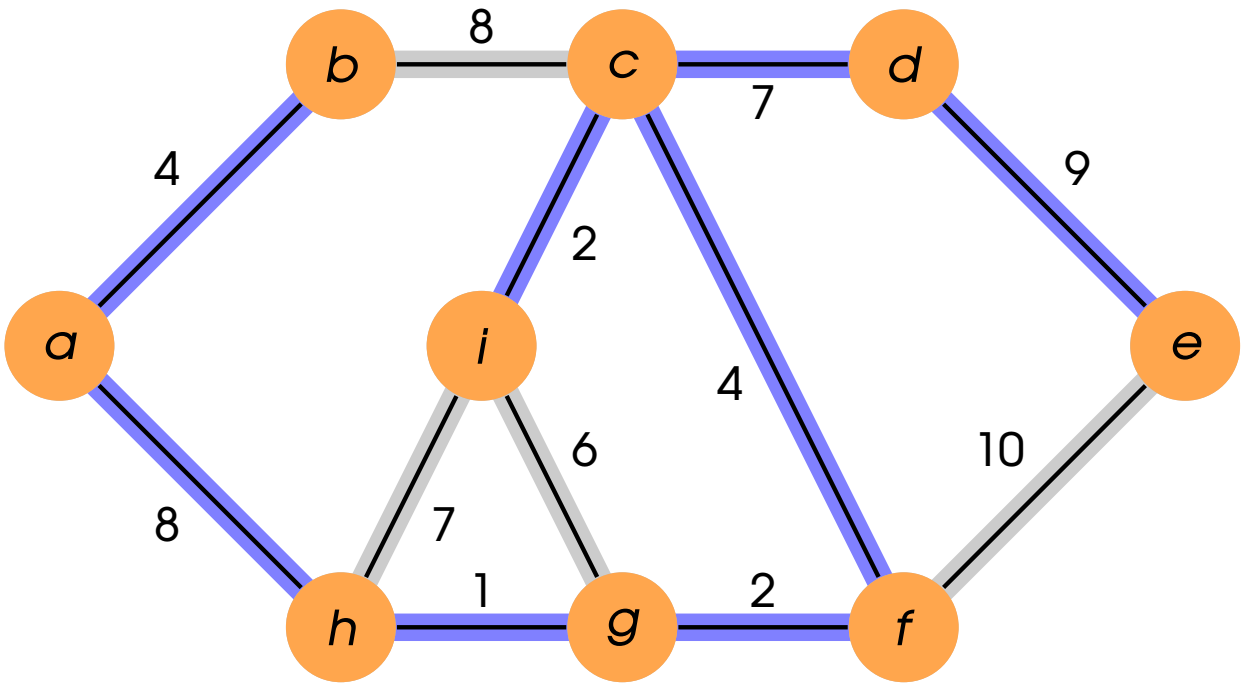
Ο αλγόριθμος του Kruskal



Ο αλγόριθμος του Kruskal



Ο αλγόριθμος του Kruskal



Σημειώματα

Σημείωμα Αναφοράς

Σημείωμα Αδειοδότησης

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.