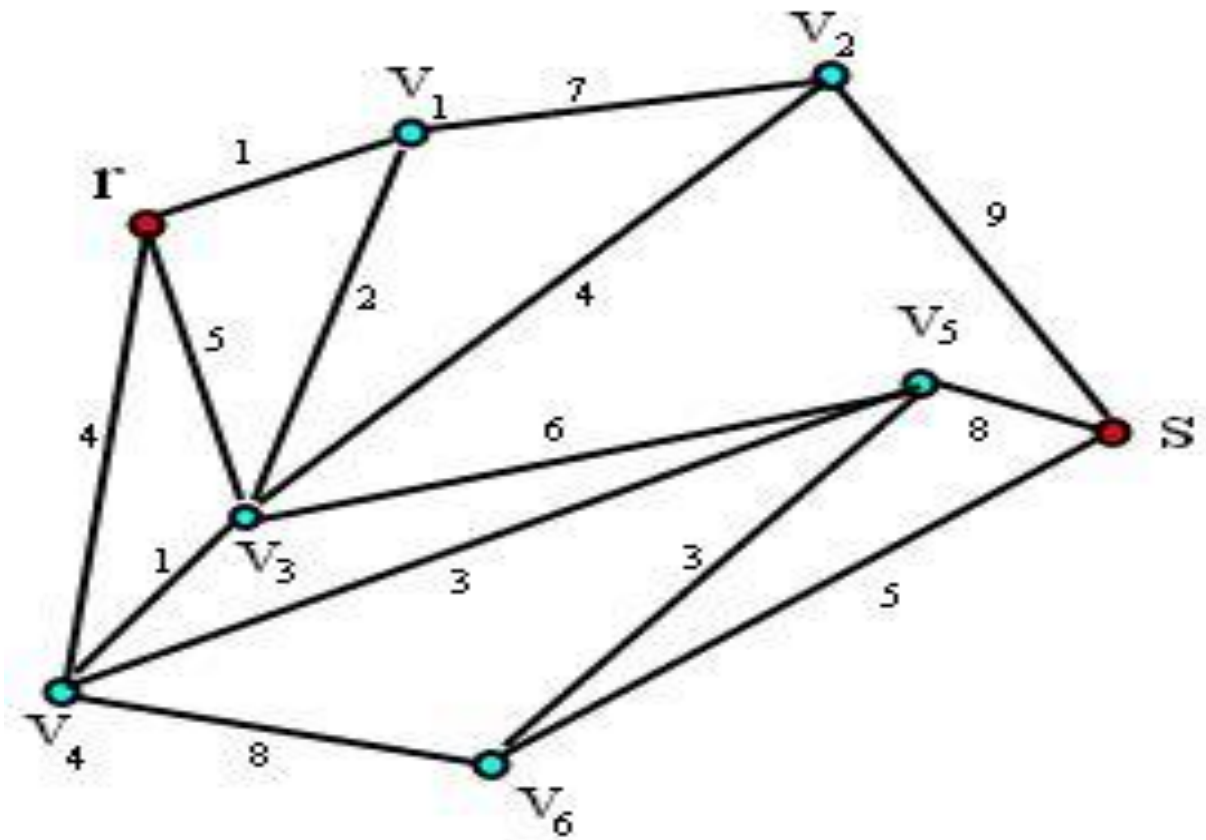
Χρήσιμοι Αλγόριθμοι

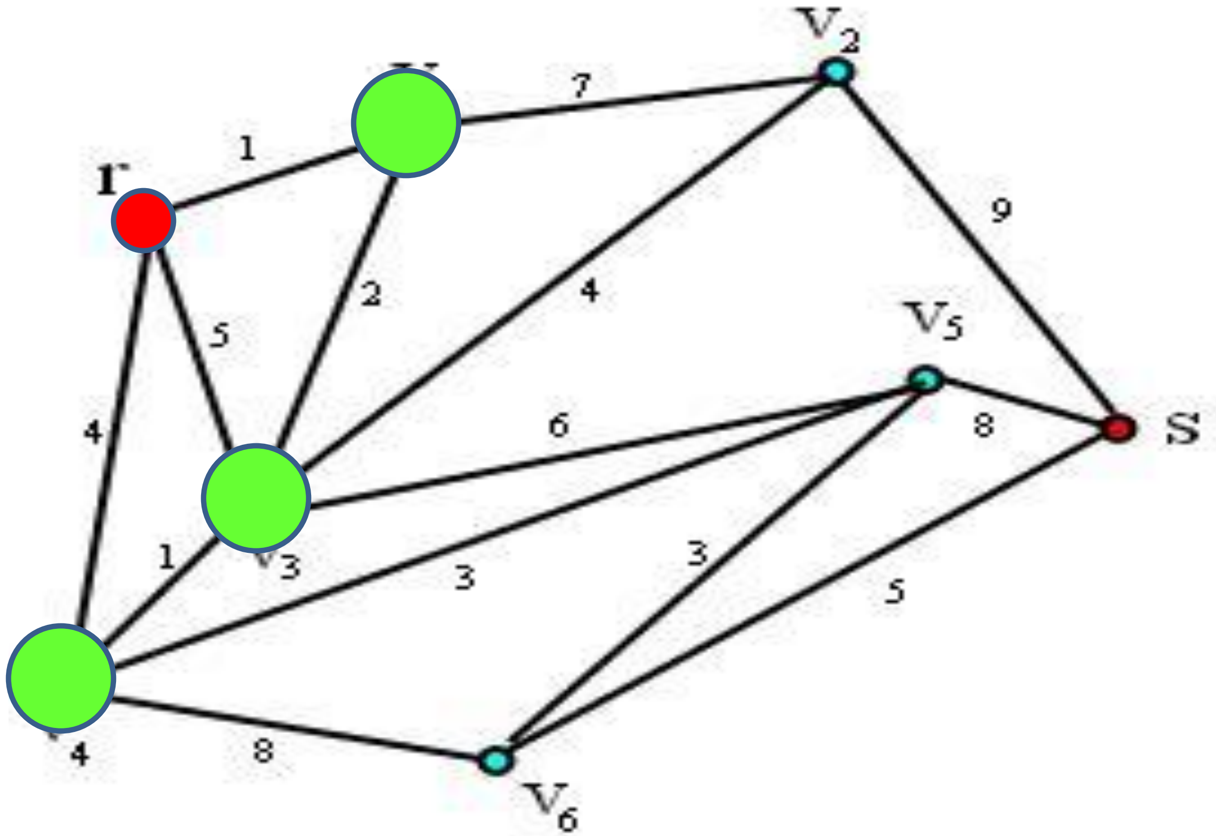
- Διαδρομή ελαχίστου βάρους από την **a** στην **s**
- Κύκλωμα ελάχιστου μήκους, περιέχει όλες τις ακμές σταθμισμένου συνεκτικού ψευδογράφου.
- Έλεγχος ύπαρξης κυκλώματος σε ψευδογράφημα.
- Εύρεση δένδρου σύνδεσης σε συνεκτικό ψευδογράφημα
- Εύρεση δένδρου σύνδεσης ελάχιστου βάρους
- Υπολογισμός του πλήθους των δένδρων σύνδεσης σε συνεκτικό ψευδογράφο.

Dijkstra’s Flow Chart

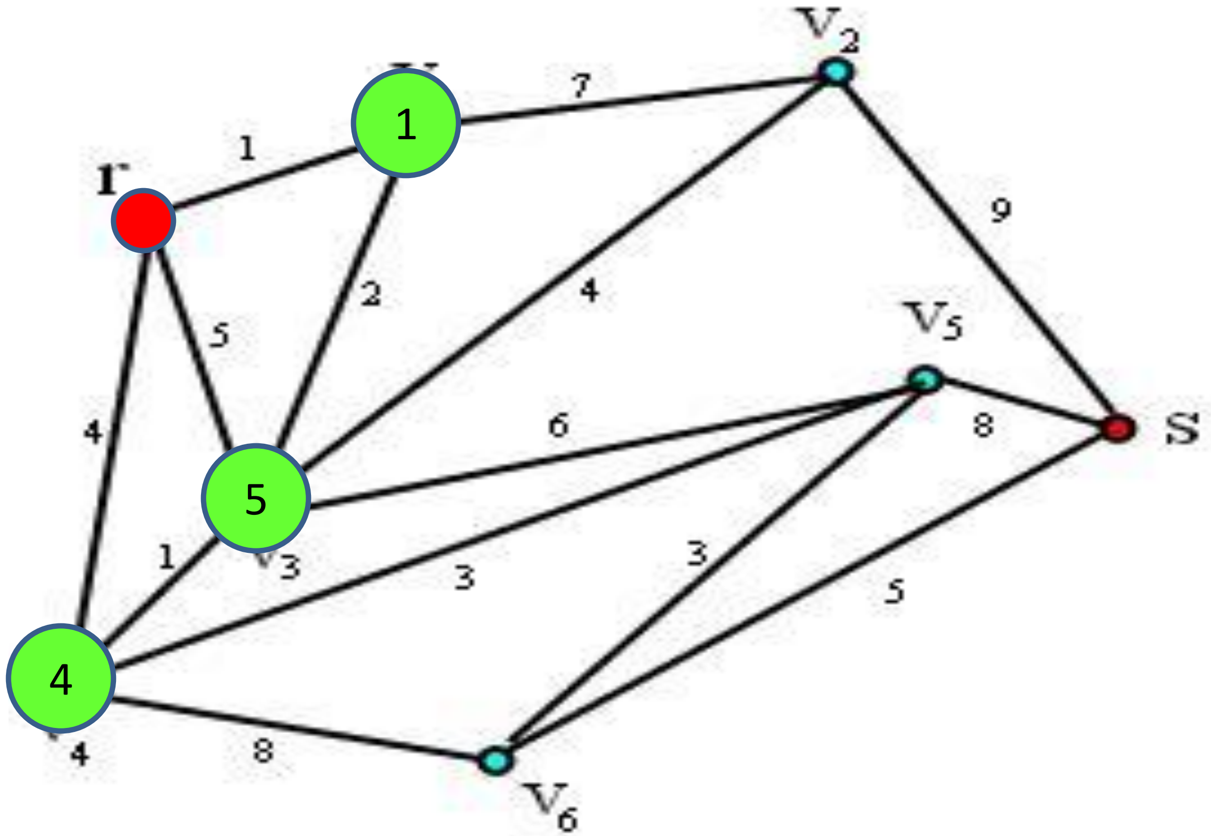
Επιλέγεται η κορυφή r



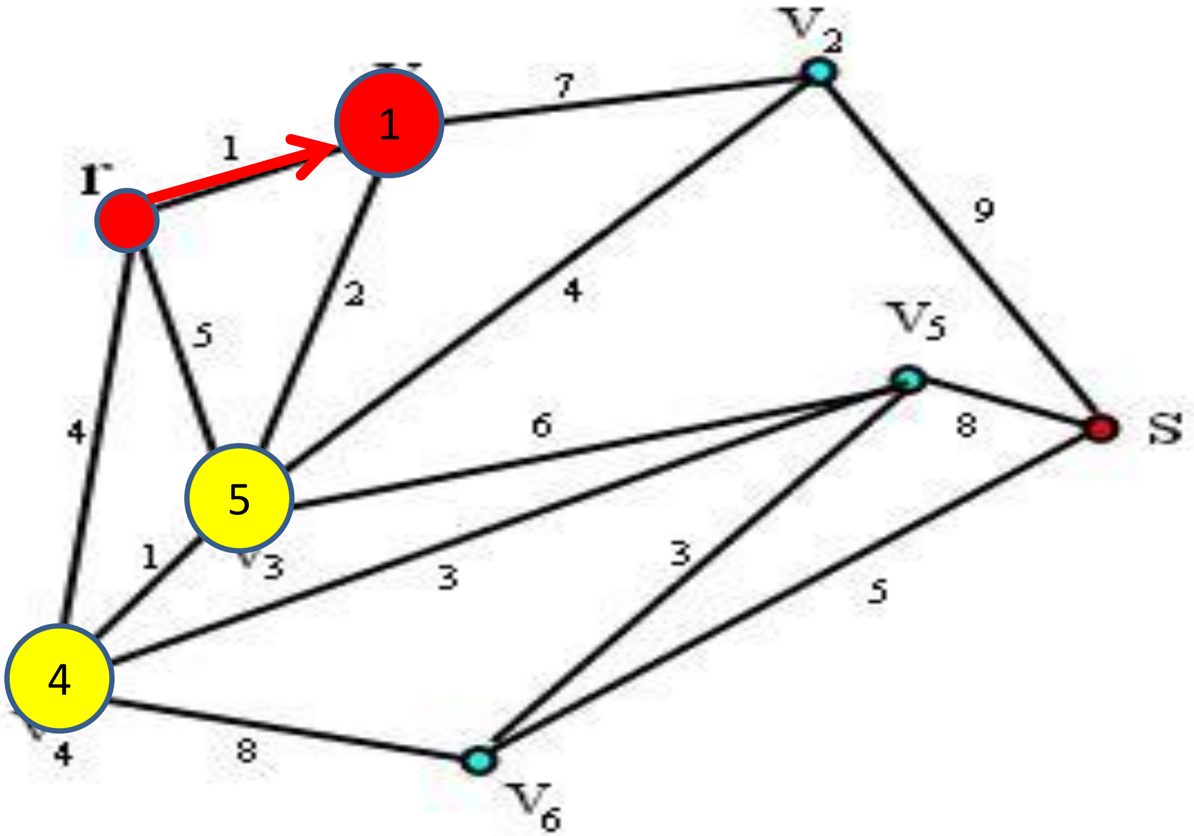
Ελέγχονται οι προσκείμενες στην r κορυφές



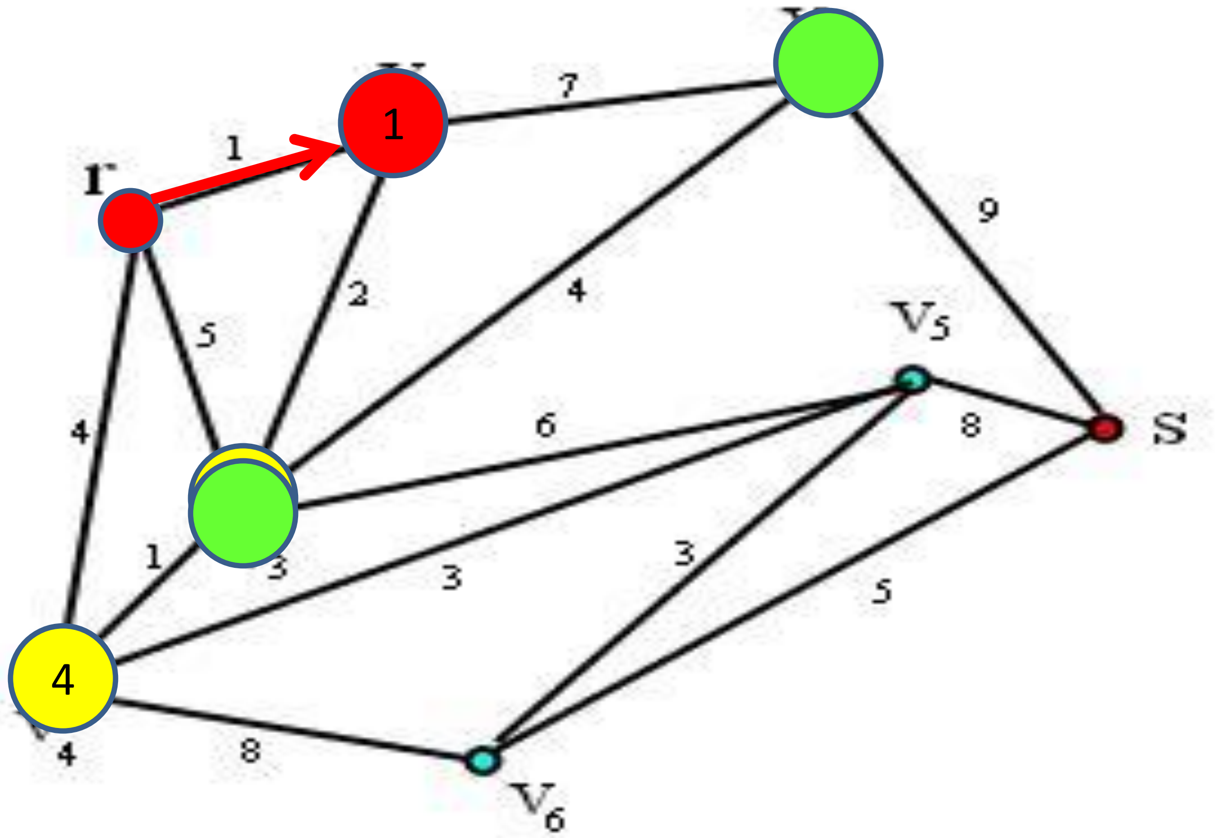
Υπολογίζονται τα βάρη



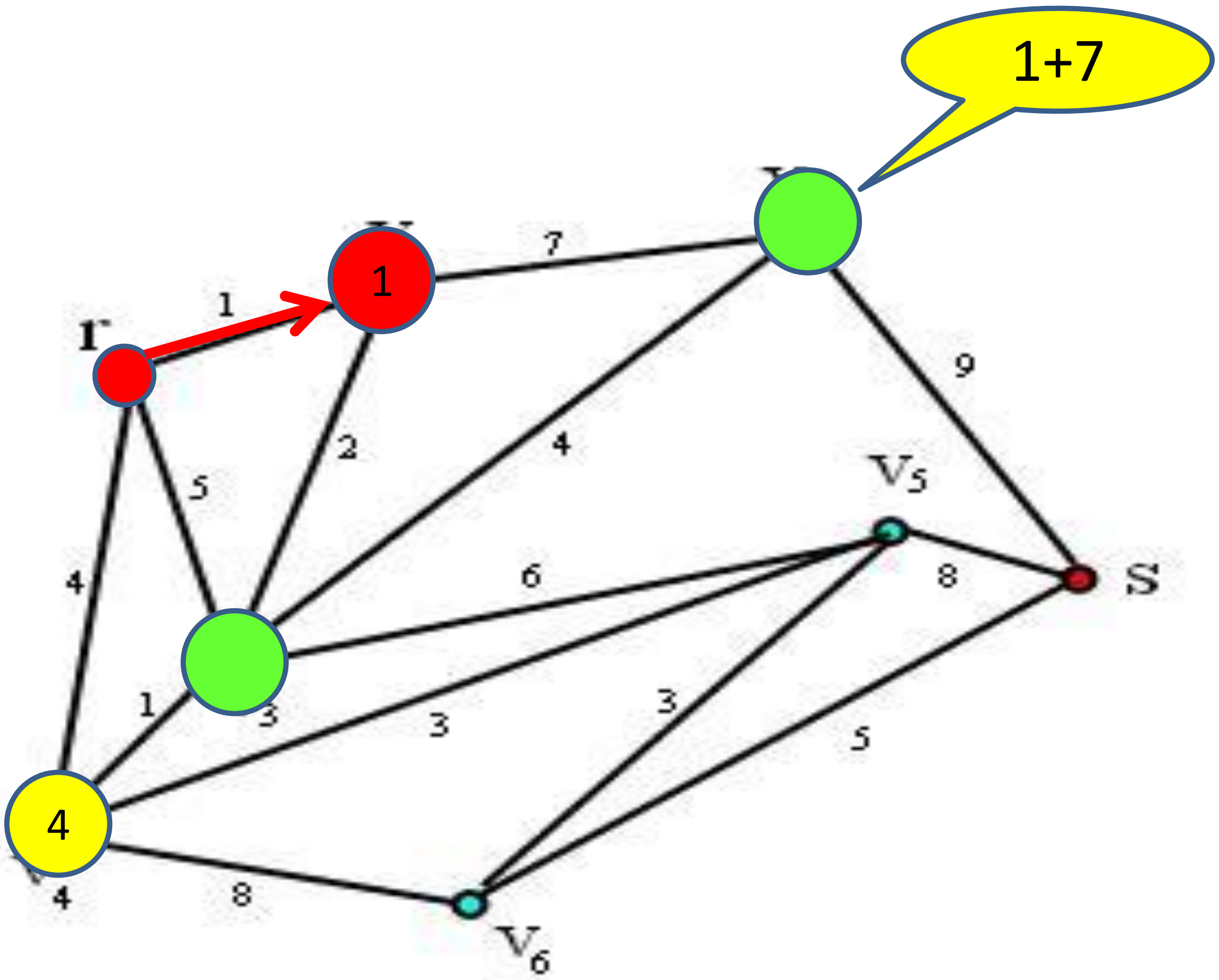
Επιλέγεται το μικρότερο



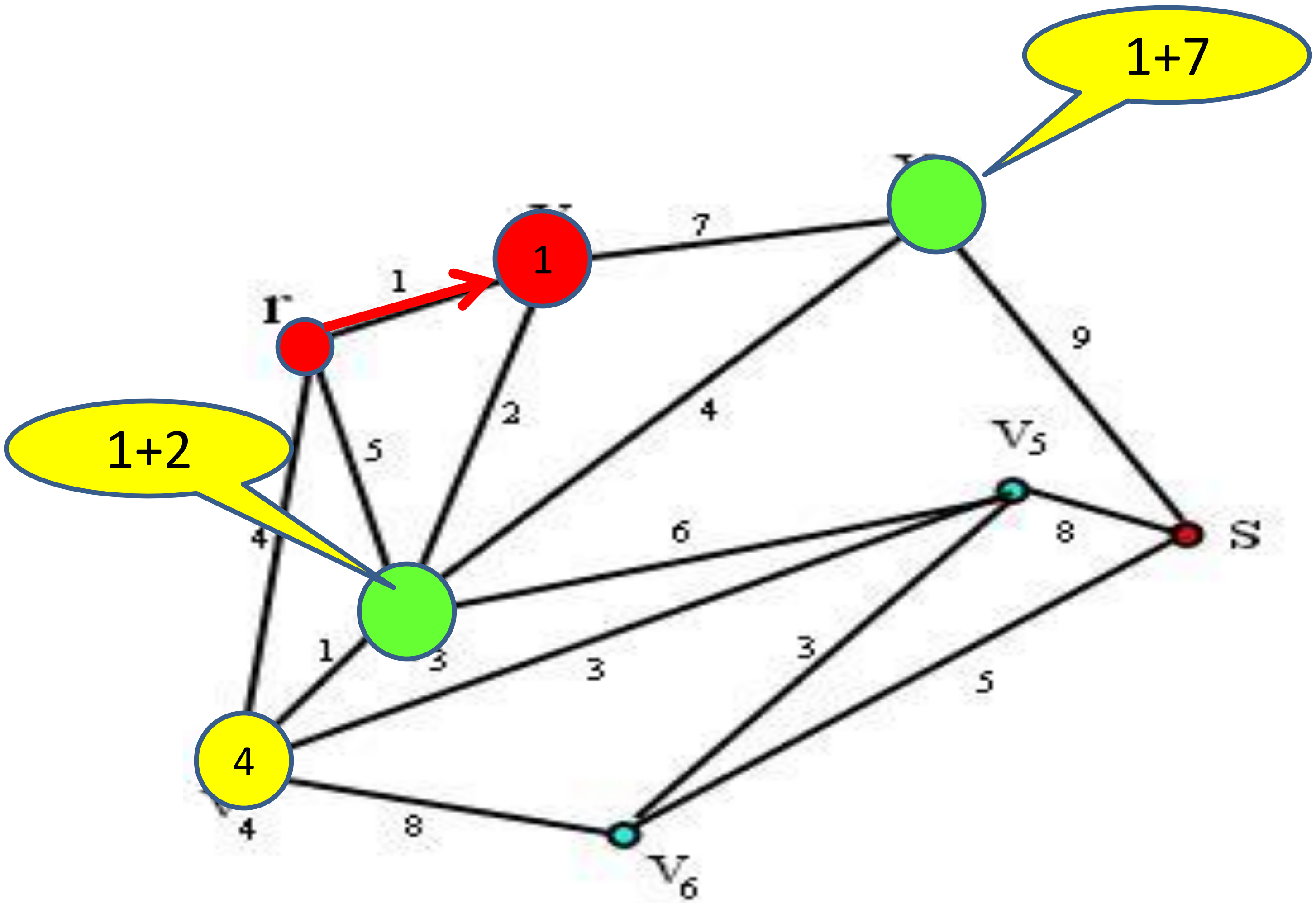
Ελέγχονται οι προσκείμενες στην v_1 κορυφές



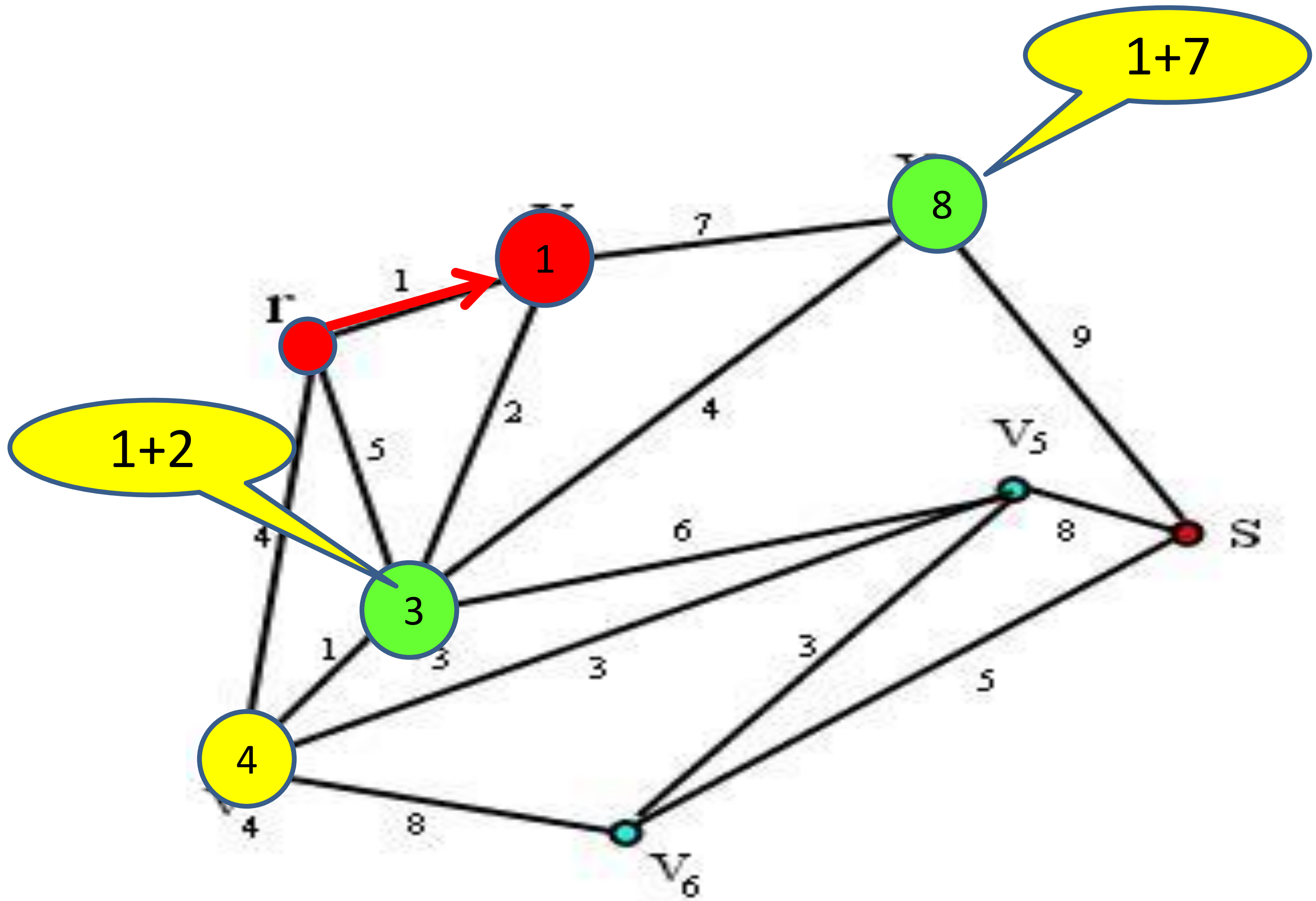
Υπολογίζονται τα βάρη



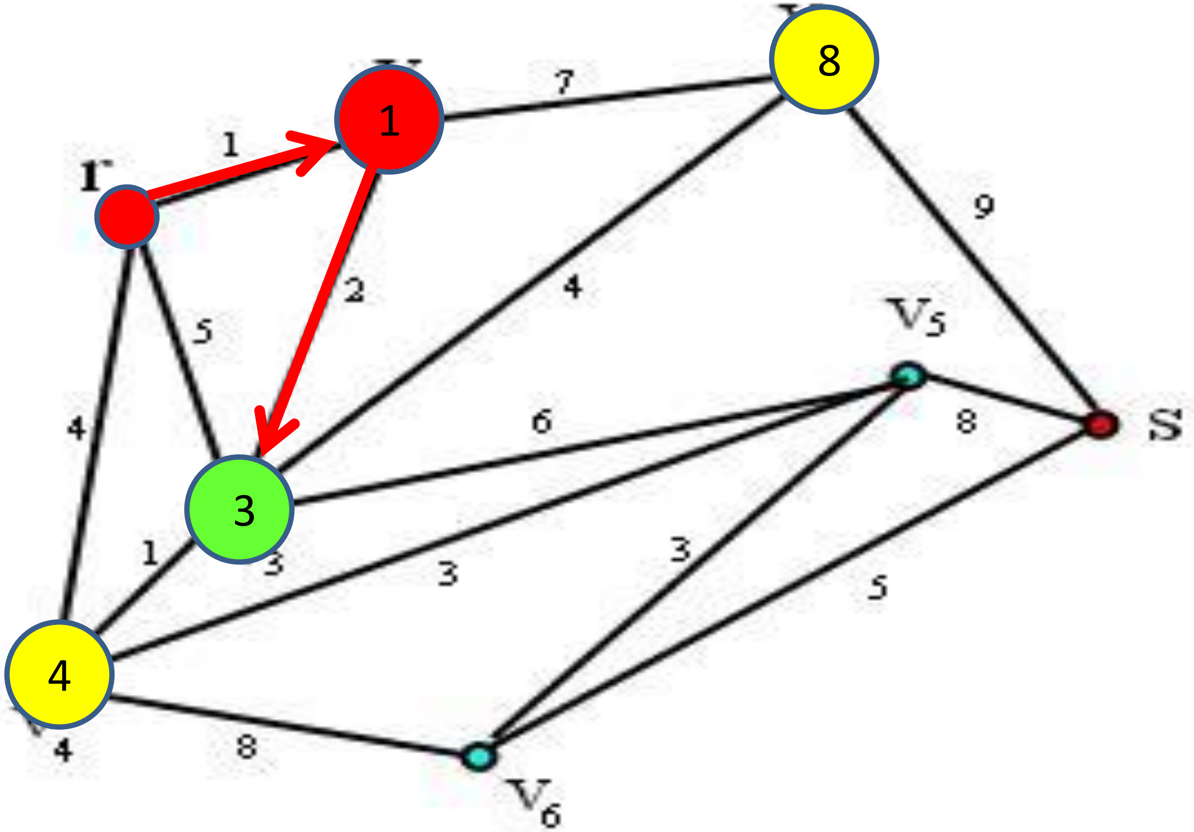
Υπολογίζονται τα βάρη



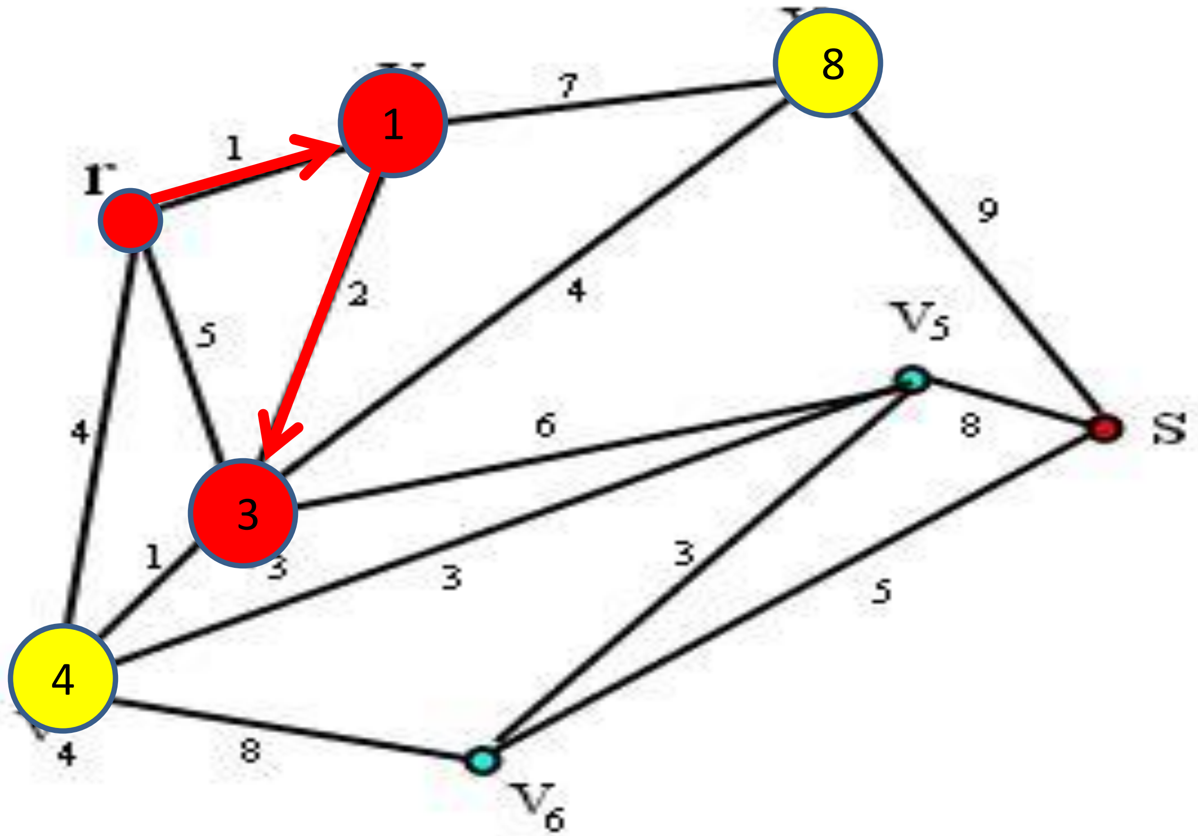
Υπολογίζονται τα βάρη



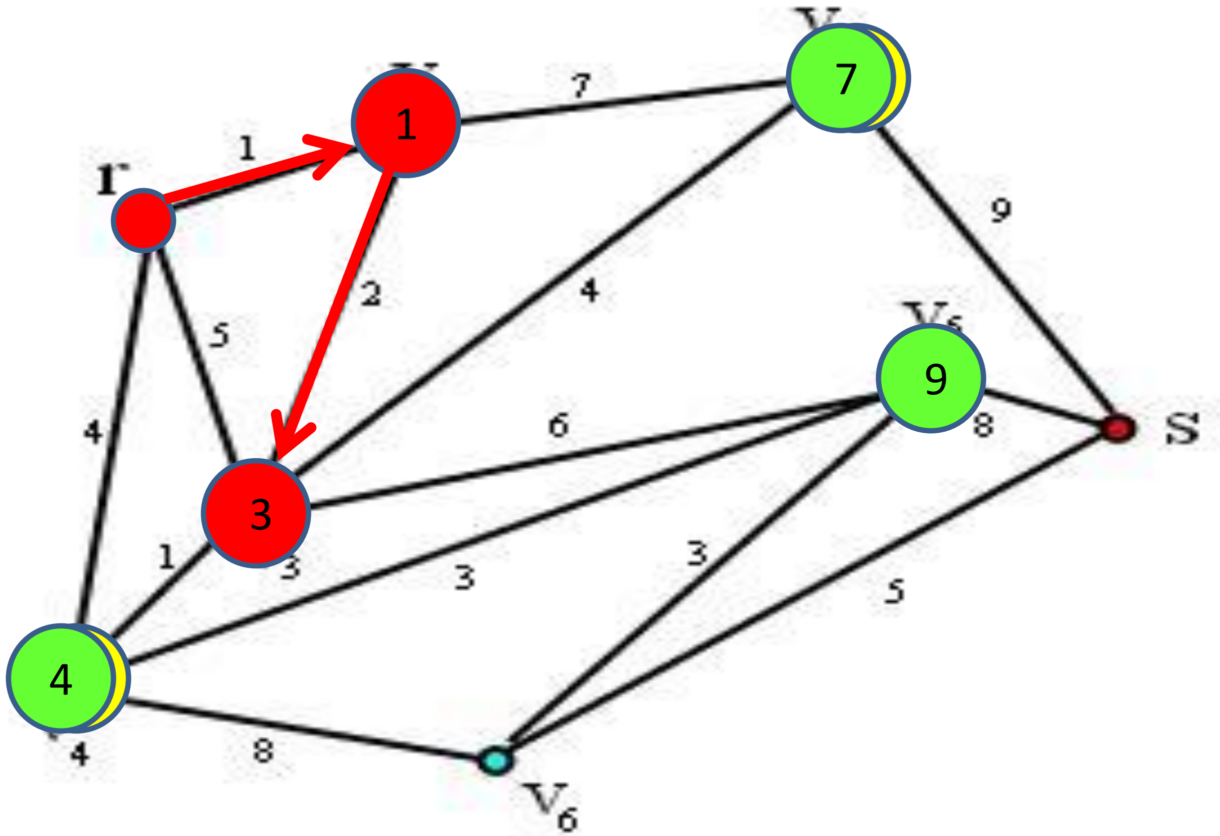
Επιλέγεται η κορυφή ελαχίστου βάρους



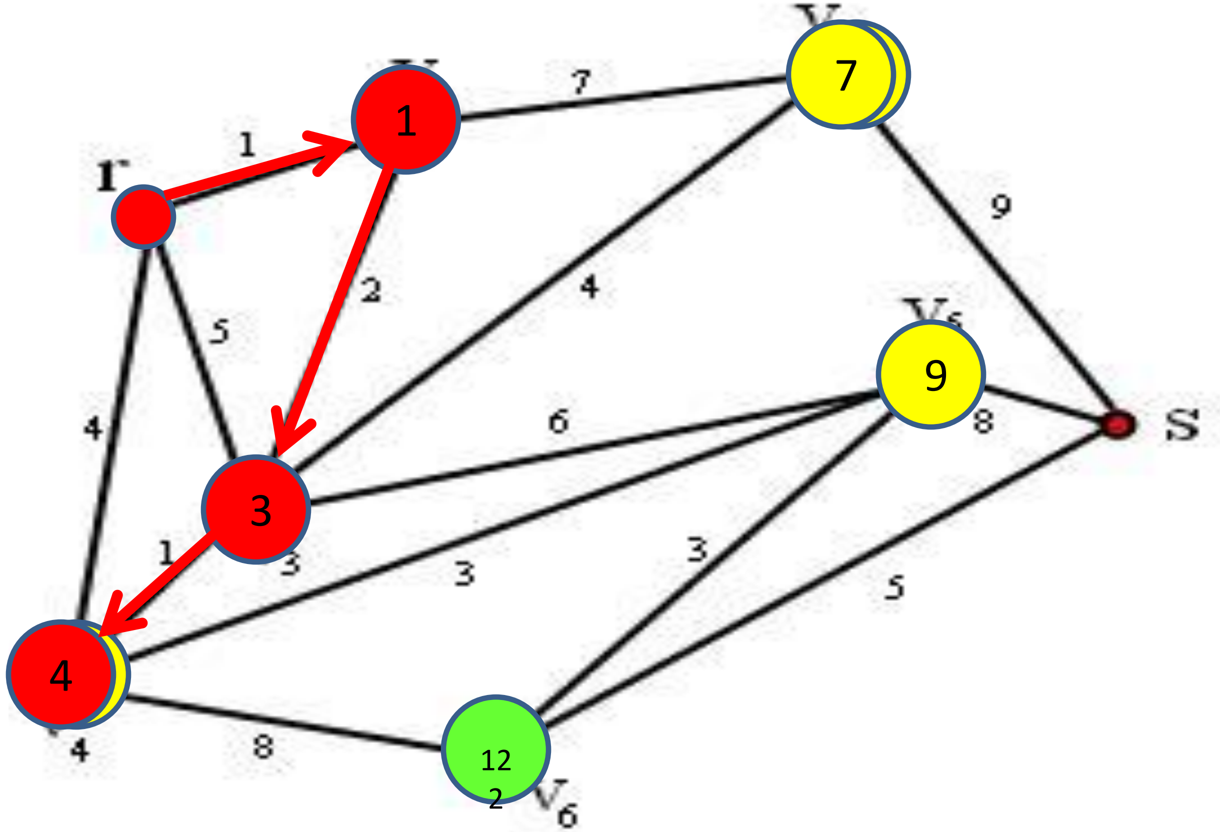
Επιλέγεται η κορυφή ελαχίστου βάρους



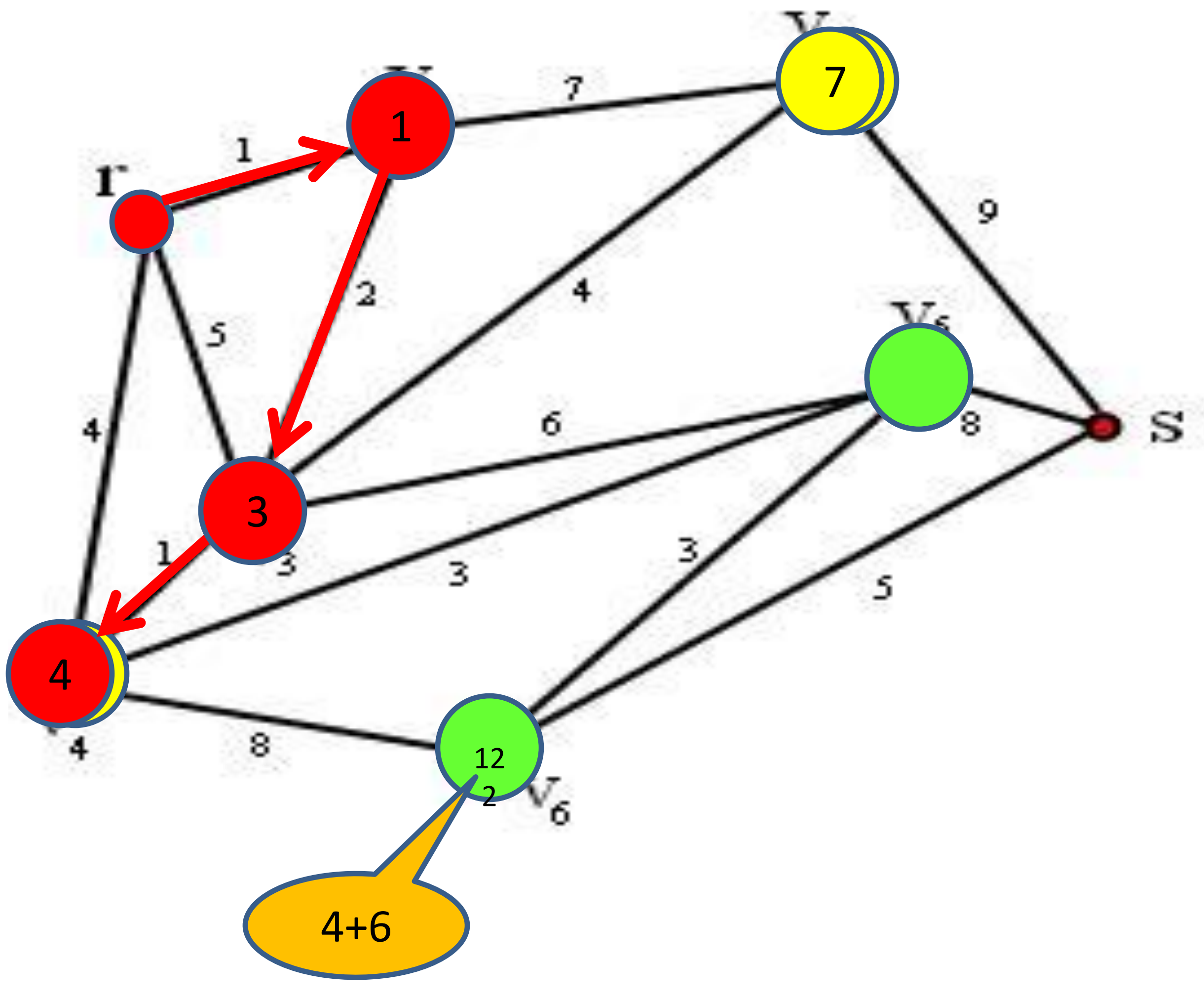
Ελέγχονται οι προσκείμενες στην v_3 κορυφές



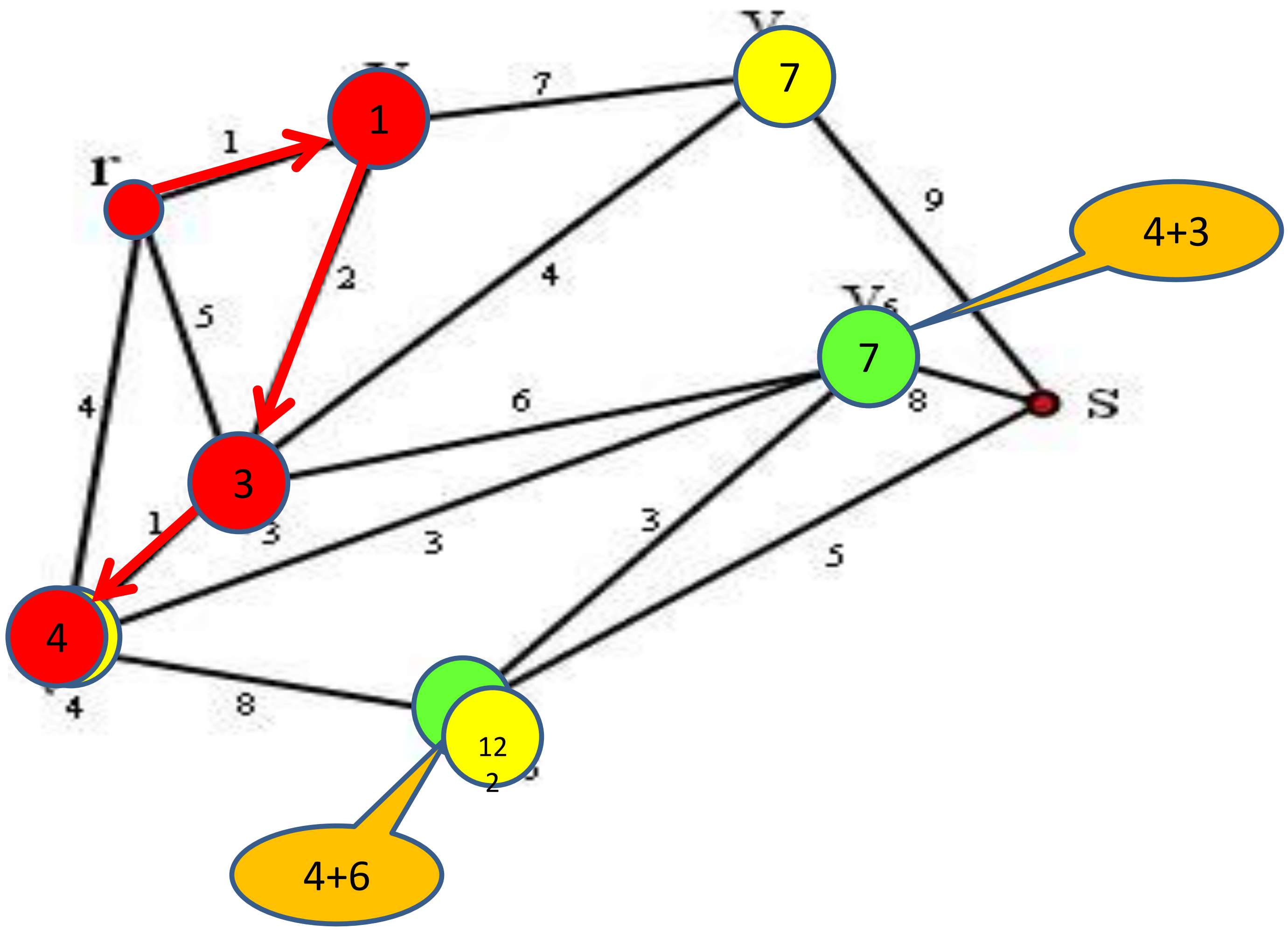
Επιλέγεται η v_4
(θα μπορούσε να επιλεγεί η διαδρομή r - v_4)



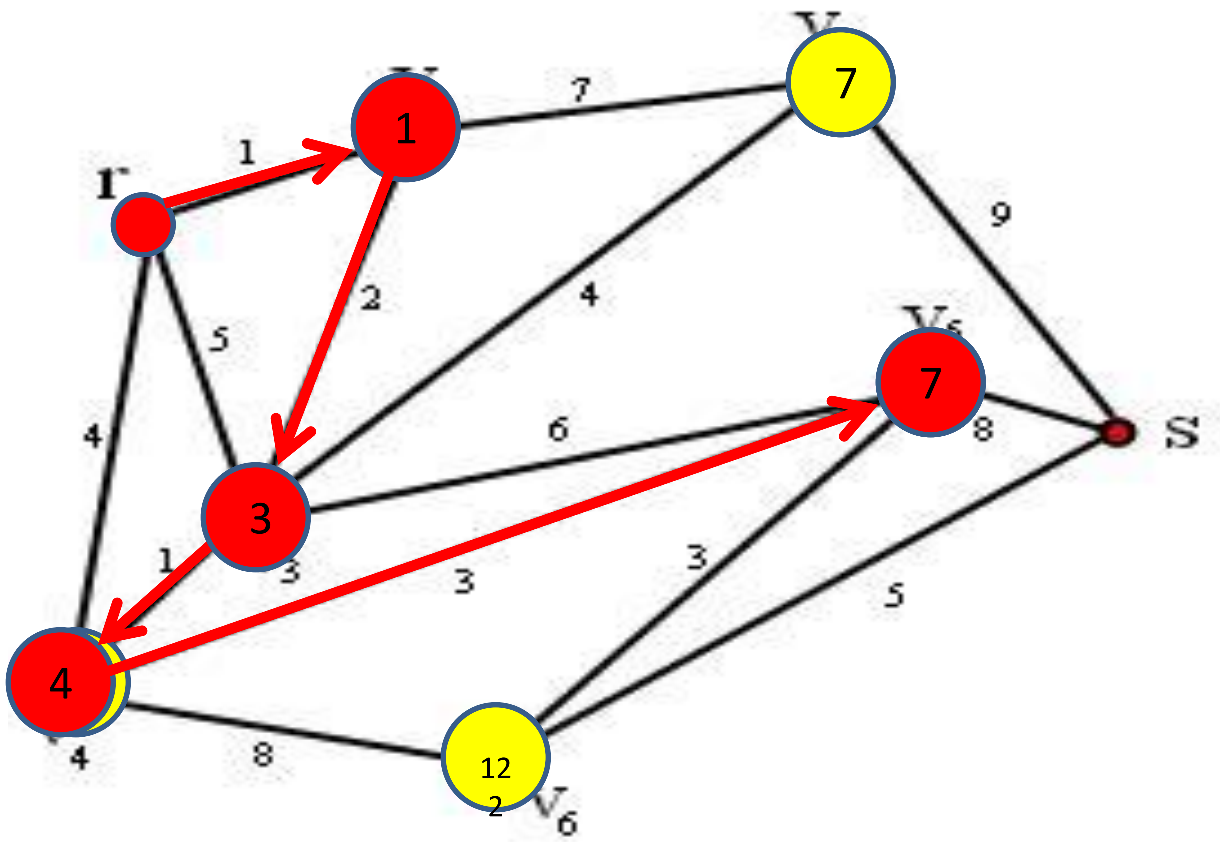
Ελέγχονται οι προσκείμενες στην v_4 κορυφές



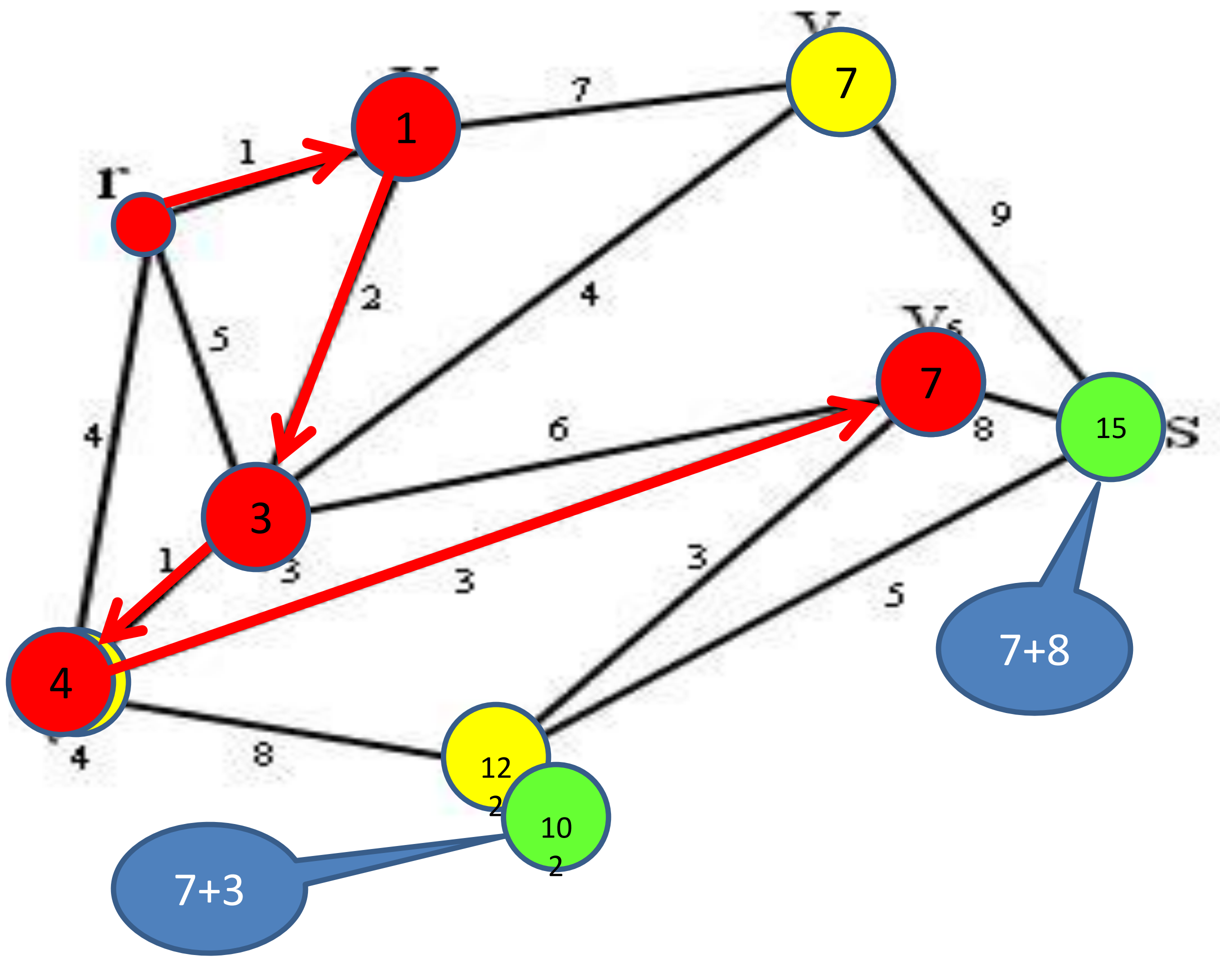
Ελέγχονται οι προσκείμενες στην v_4 κορυφές



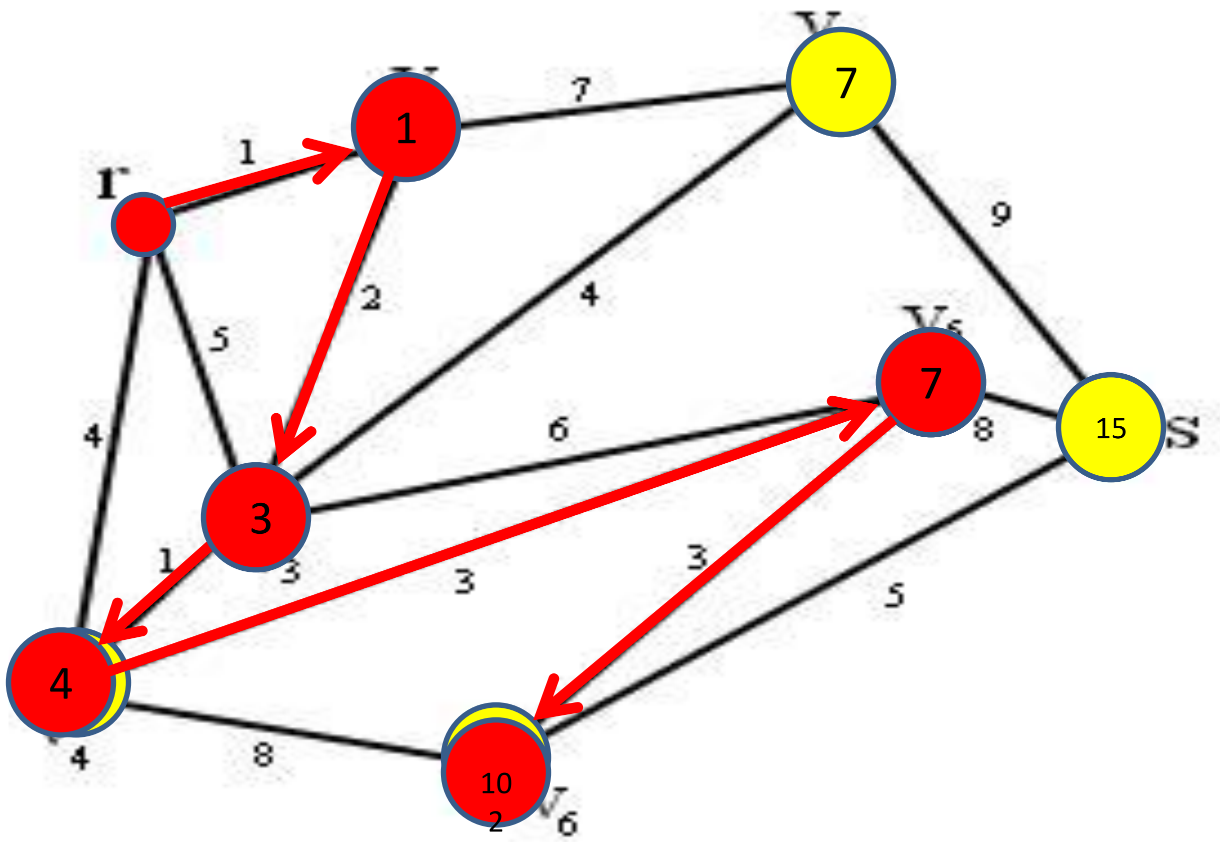
Επιλέγεται η v_6 κορυφή



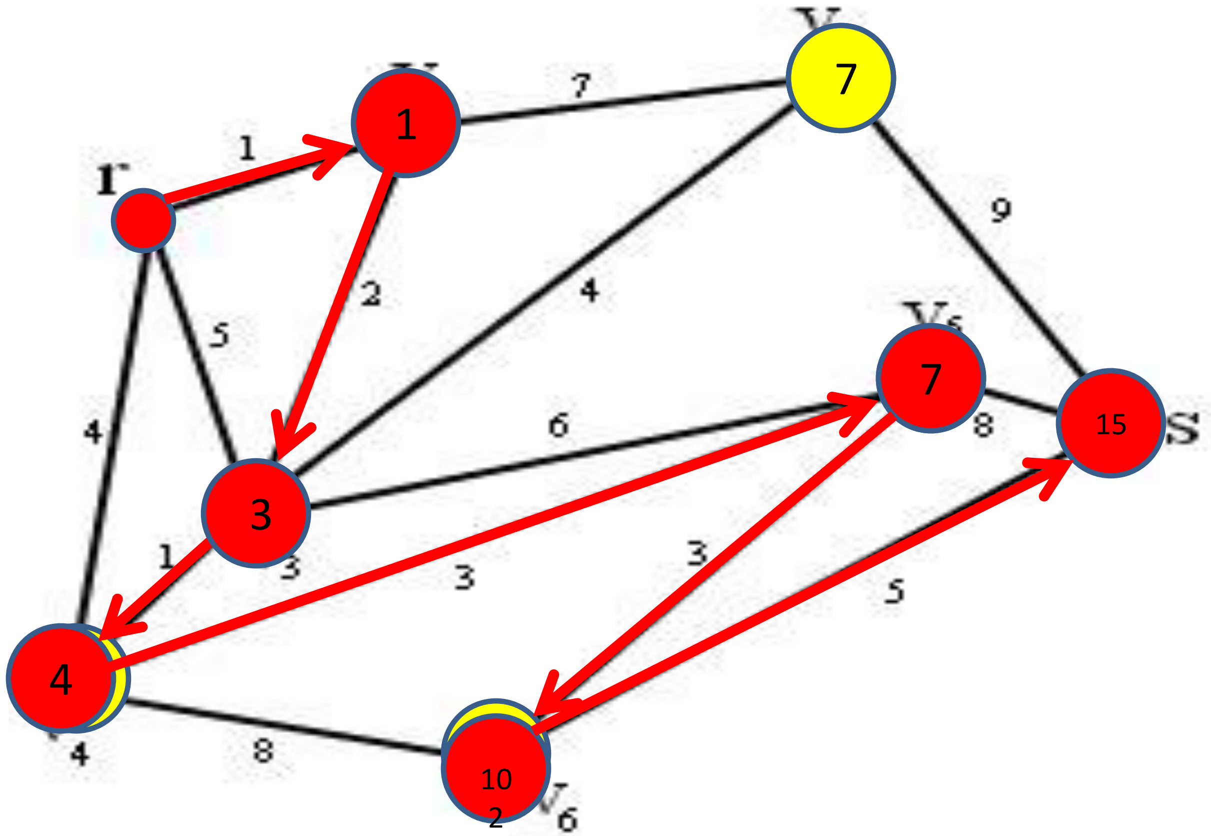
Επιλέγονται οι προσκείμενες v_6 κορυφή



Επιλέγονται οι προσκείμενες v_6 κορυφή



Επιλέγονται οι προσκείμενες v_6 κορυφή

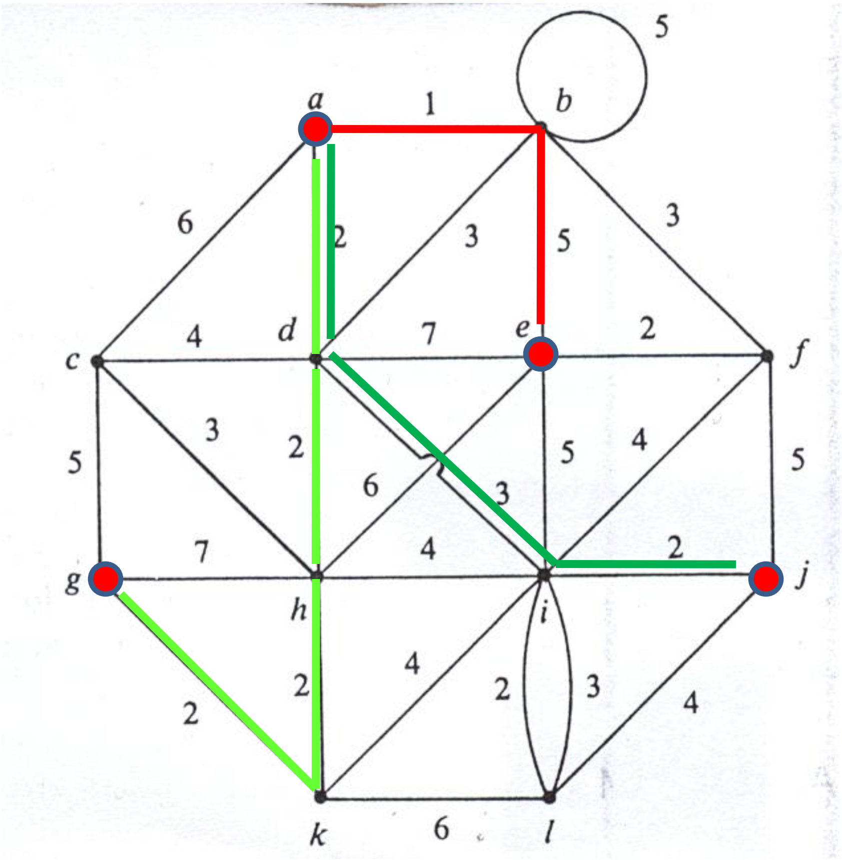


Κύκλωμα ελάχιστου μήκους που περιέχει όλες τις ακμές σταθμισμένου συνεκτικού ψευδογράφου G.

1. **Αν δεν υπάρχουν κορυφές περιττού βαθμού ο έλεγχος στο 7.**
2. **Σημειώνονται οι κορυφές περιττού βαθμού u_1, u_2, … ,u_{2m}.**
3. **Εφαρμόστε τον αλγόριθμο Dijkstra για κάθε ζεύγος κορυφών περιττού βαθμού.**
4. **Καταγράψτε όλους τους δυνατούς συνδυασμούς ζευγών περιττού βαθμού και σημειώστε το άθροισμα των βαρών σε κάθε περίπτωση.**
5. **Αντιγράψτε κάθε ακμή του G που αντιστοιχεί σε ελάχιστη διαδρομή ανάμεσα σε ζεύγη κορυφών που επιλέχθηκαν στο βήμα 4. Ονομάστε το νέο ψευδογράφο G'.**
6. **Βρείτε Οϊλεριανό κύκλωμα για το G'. Κάθε ακμή που δεν είναι στο G' αντιστοιχεί σε ακμή του G που διασχίζεται περισσότερο από μια φορά. STOP**
7. **Το Οϊλεριανό κύκλωμα για το Γ είναι το κύκλωμα ελαχίστου βάρους. STOP**

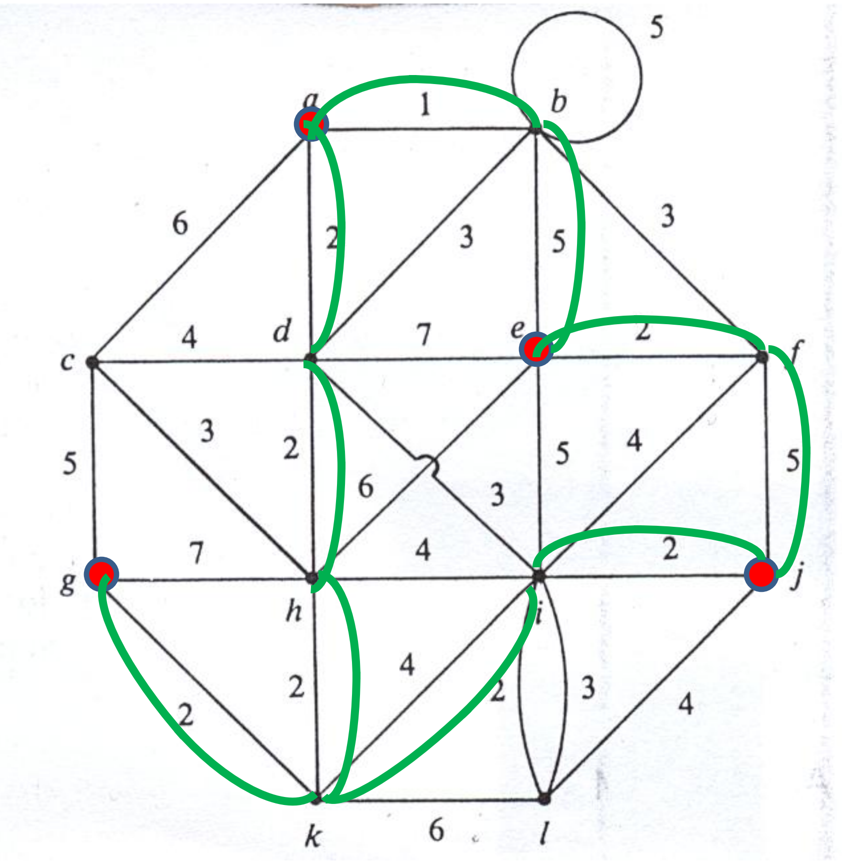
Κύκλωμα ελάχιστου μήκους που περιέχει όλες τις ακμές σταθμισμένου συνεκτικού ψευδογράφου

- Σημειώνονται a, e, g και j
- Από την εφαρμογή του αλγόριθμου Dijkstra στα έξι ζεύγη δίνει τις διαδρομές ελαχίστου βάρους:
  - w(a,e)=6 a-b-e (a-b-f-e)
  - w(a,g)=8 a-d-h-k-g.
  - w(a,j)=7 a-d-i-j
  - w(e,g)=10 e-h-k-g
  - w(e,j)=7 e-f-j (e-i-j)
  - w(g,j)=8 g-k-i-j
- Οι δυνατοί συνδυασμοί είναι
  - w(a,e)+w(g,j)=6+8=14
  - w(a,g)+w(e,j)=8+7=15
  - w(a,j)+w(e,g)=7+10=17



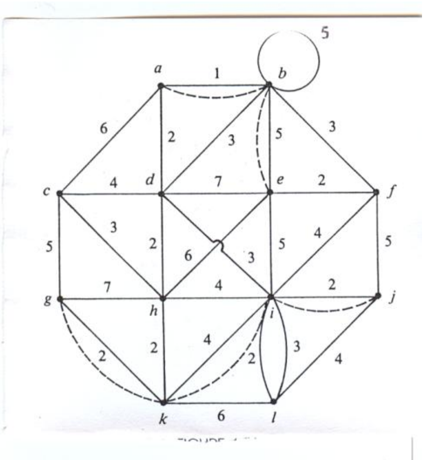
Κύκλωμα ελάχιστου μήκους που περιέχει όλες τις ακμές σταθμισμένου συνεκτικού ψευδογράφου

- Σημειώνονται a, e, g και j
- Οι δυνατοί συνδυασμοί είναι
  - w(a,e)+w(g,j)=6+8=14
  - w(a,g)+w(e,j)=8+7=15
  - w(a,j)+w(e,g)=7+10=17



Κύκλωμα ελάχιστου μήκους που περιέχει όλες τις ακμές σταθμισμένου συνεκτικού ψευδογράφου

- Επιλέγεται το πρώτο σύστημα ζευγών και προστίθενται οι αντίστοιχες διαδρομές στο G για να προκύψει ο γράφος της εικόνας
- Εφαρμόζεται ο αλγόριθμος Dijkstra



ab 1	ch 3	ik 4	eh 6	be* 5	ji 2	da 2	jl 4
bb 5	hg 7	kh 2	hd 2	ef 2	il 3	ab 1	ik 6
bd 3	gk 2	hi 4	de 7	fi 4	li 2	bf 3	kg 2
dc 4	ki 4	ie 5	eb 5	ij 2	id 3	fj 5	gc 5

Δένδρα σύνδεσης (Spanning trees)

- Ορισμός: Δένδρο σύνδεσης σε ένα συνεκτικό ψευδογράφο είναι κάθε υπογράφημα δένδρο T’ του G που συνδέει όλες τις κορυφές του G.

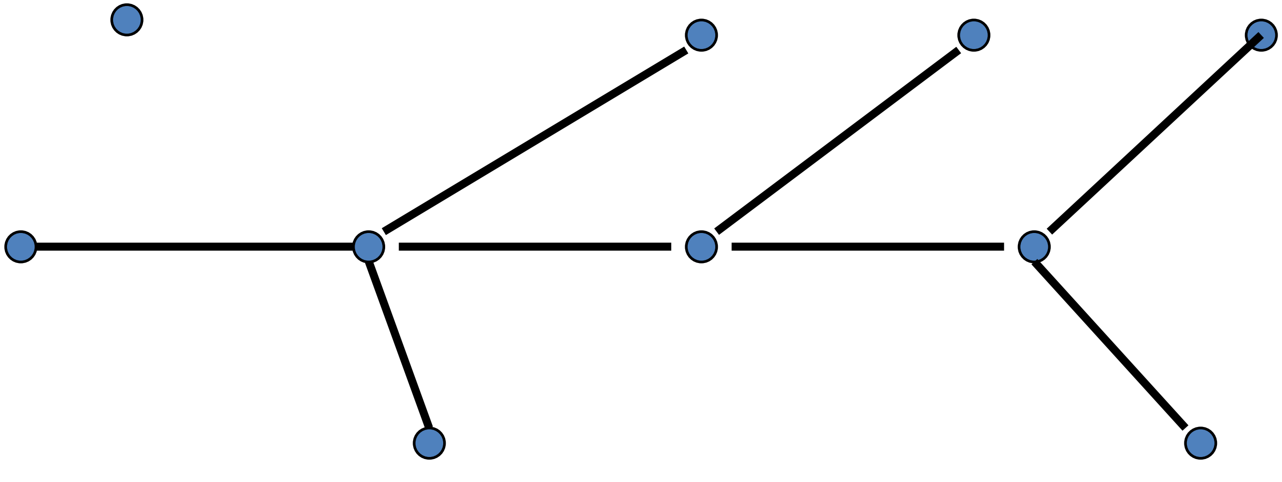
Δένδρα σύνδεσης

- Πρόβλημα 1^{o} : Να σχεδιαστεί ένα δένδρο σύνδεσης σε ένα γράφο G.
- Πρόβλημα 2^{o} : Να σχεδιαστεί ένα δένδρο σύνδεσης ελαχίστου βάρους σε ένα σταθμισμένο συνεκτικό γράφο G.
- Πρόβλημα 3^{o} : Να υπολογισθεί το πλήθος των δένδρων σύνδεσης σε ένα συνεκτικό ψευδογράφο.

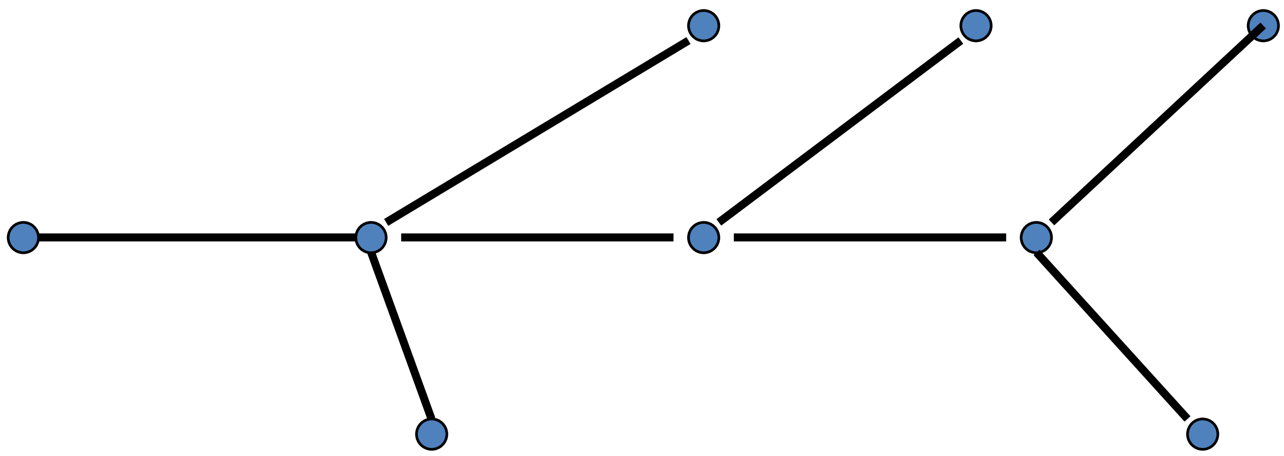
Αλγόριθμος ελέγχου ύπαρξης κυκλώματος

1. G←G Απόσυρση όλων των κορυφών μηδενικού βαθμού
2. Αν G έχει μια μόνο κορυφή πήγαινε στο 4.
3. Ο ψευδογράφος δεν έχει κυκλώματα. STOP
4. Αν G έχει μια κορυφή βαθμού 1, πήγαινε στο 6.
5. Ο ψευδογράφος δεν έχει κυκλώματα. STOP
6. G←G Απόσυρση όλων των κορυφών βαθμού 1 και των ακμών που καταλήγουν σε αυτές.
7. Πήγαινε στο 1.

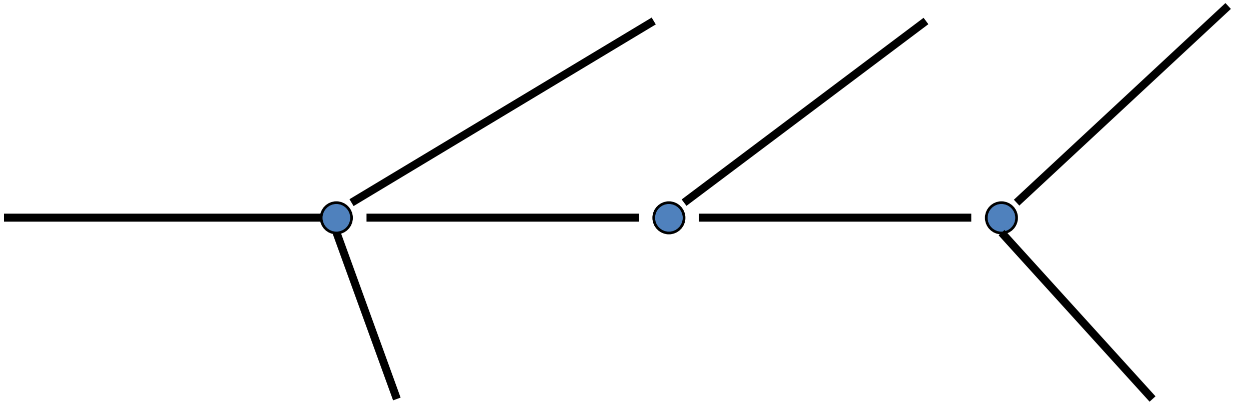
Εφαρμογή



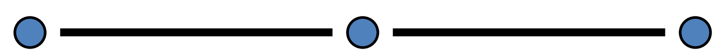
Εφαρμογή



Εφαρμογή



Εφαρμογή



Εφαρμογή

Εφαρμογή

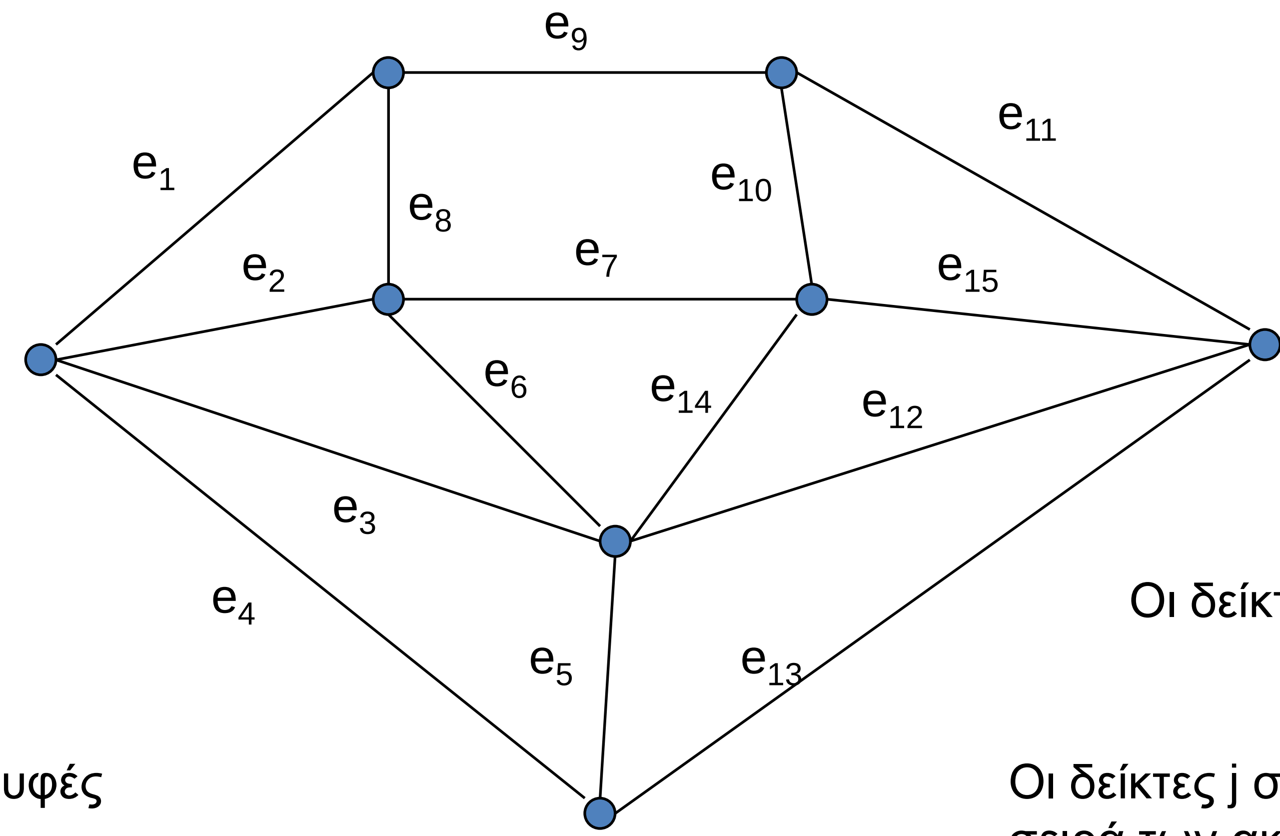
Εφαρμογή

- (να γραφούν σχόλια)

Πρόβλημα 1ο - Αλγόριθμος

1. Διαμορφώστε μια διατεταγμένη λίστα L.
2. Αρχικά $L \leftarrow \emptyset$, $j \leftarrow o$, $i \leftarrow 1$
3. $j=n-1$; Αν όχι ο έλεγχος στο 5.
4. “L είναι το σύνολο ακμών το δένδρου σύνδεσης”. STOP
5. H $\leftarrow$ο υπογράφος του G με σύνολο ακμών $L \cup \{e_i\}$.
6. Διακρίνεται στον H ένα κύκλωμα; Αν ναι ο έλεγχος στο 8.
7. $L \leftarrow L \cup \{e_i\}$, $j \leftarrow j+1$
8. $i \leftarrow i+1$
9. Ο έλεγχος στο 3.

Πρόβλημα 1ο - Αλγόριθμος

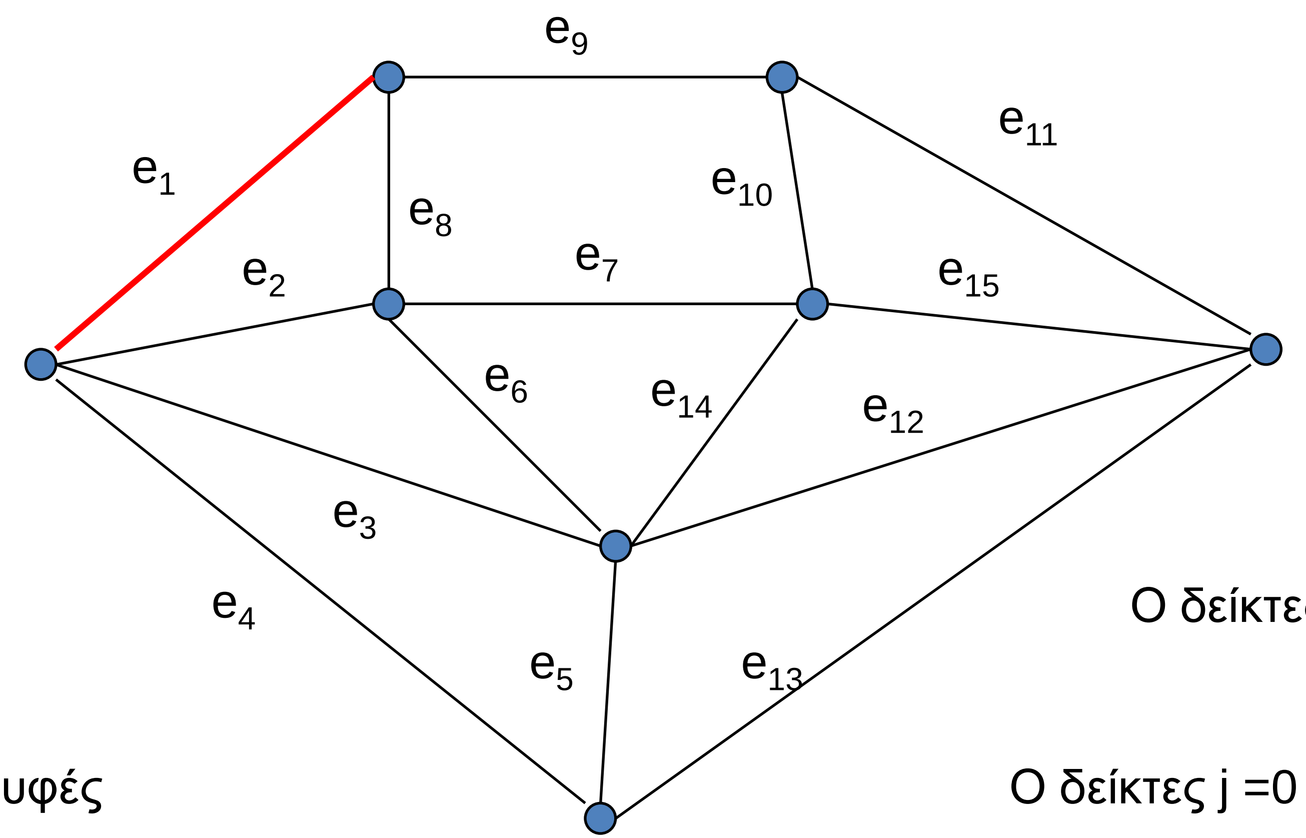


Οι δείκτες i των ακμών

n=8 κορυφές

Οι δείκτες j σημειώνουν τη σειρά των ακμών που προστίθενται στο δένδρο σύνδεσης.

Πρόβλημα 1ο - Αλγόριθμος



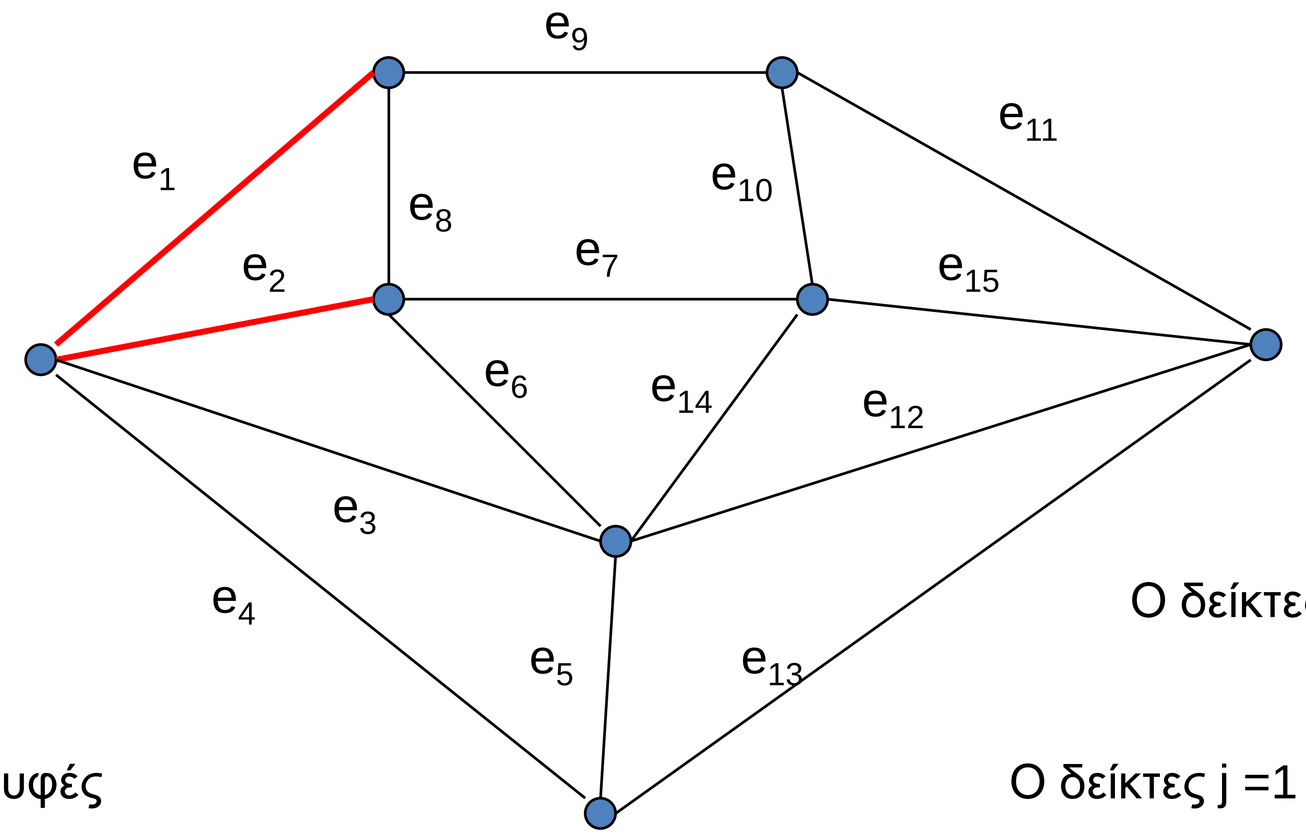
Ο δείκτες i =0

n=8 κορυφές

Ο δείκτες j =0

L={e_1}

Πρόβλημα 1ο - Αλγόριθμος



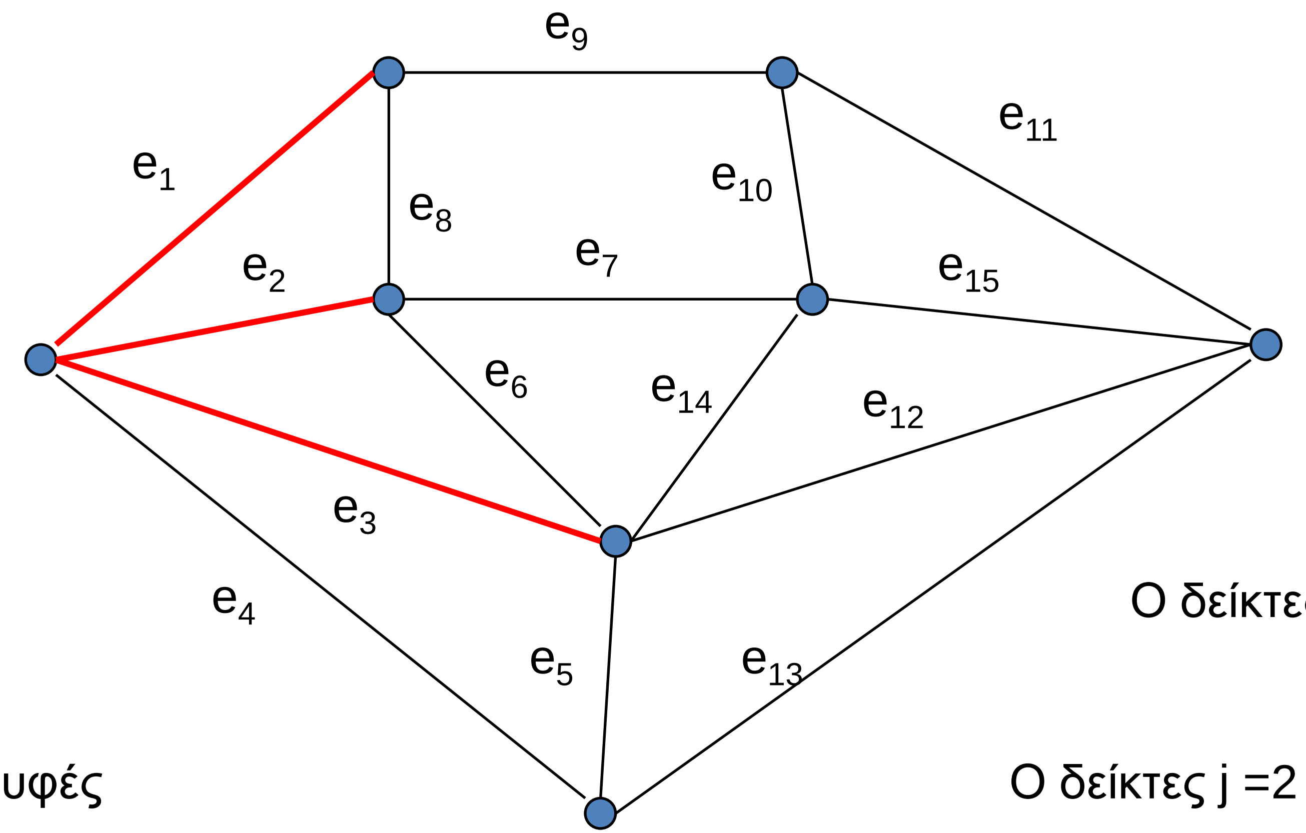
O δείκτες i =1

n=8 κορυφές

O δείκτες j =1

L={e_1, e_2}

Πρόβλημα 1ο - Αλγόριθμος



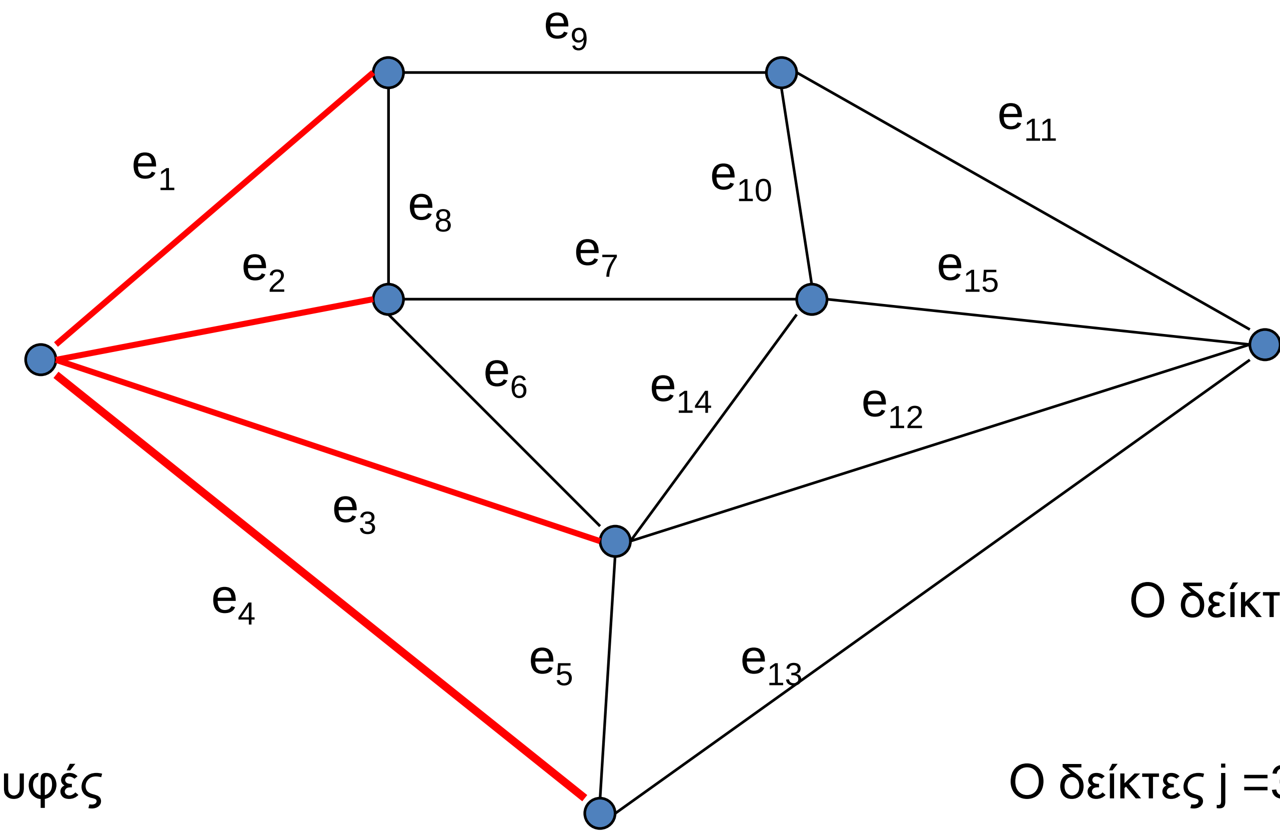
Ο δείκτες i =2

n=8 κορυφές

Ο δείκτες j =2

L={e_1, e_2, e_3}

Πρόβλημα 1ο - Αλγόριθμος



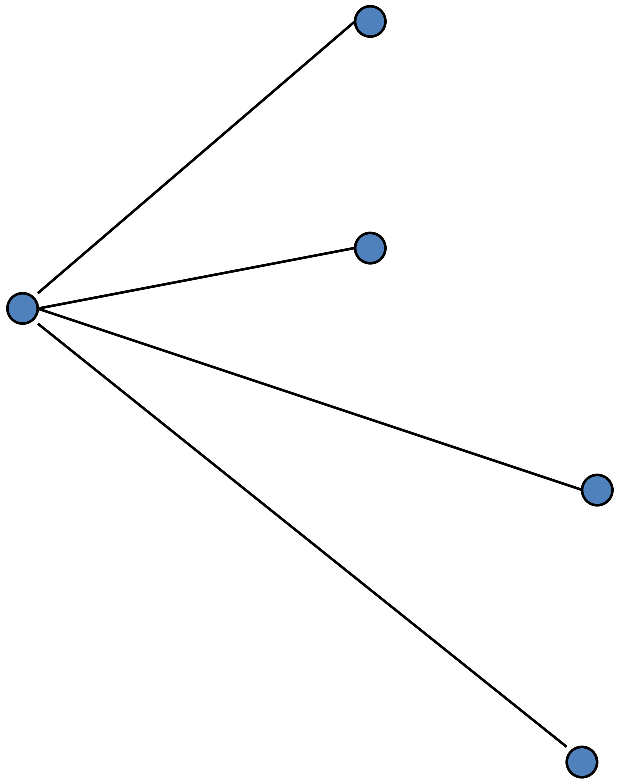
Ο δείκτες i =3

n=8 κορυφές

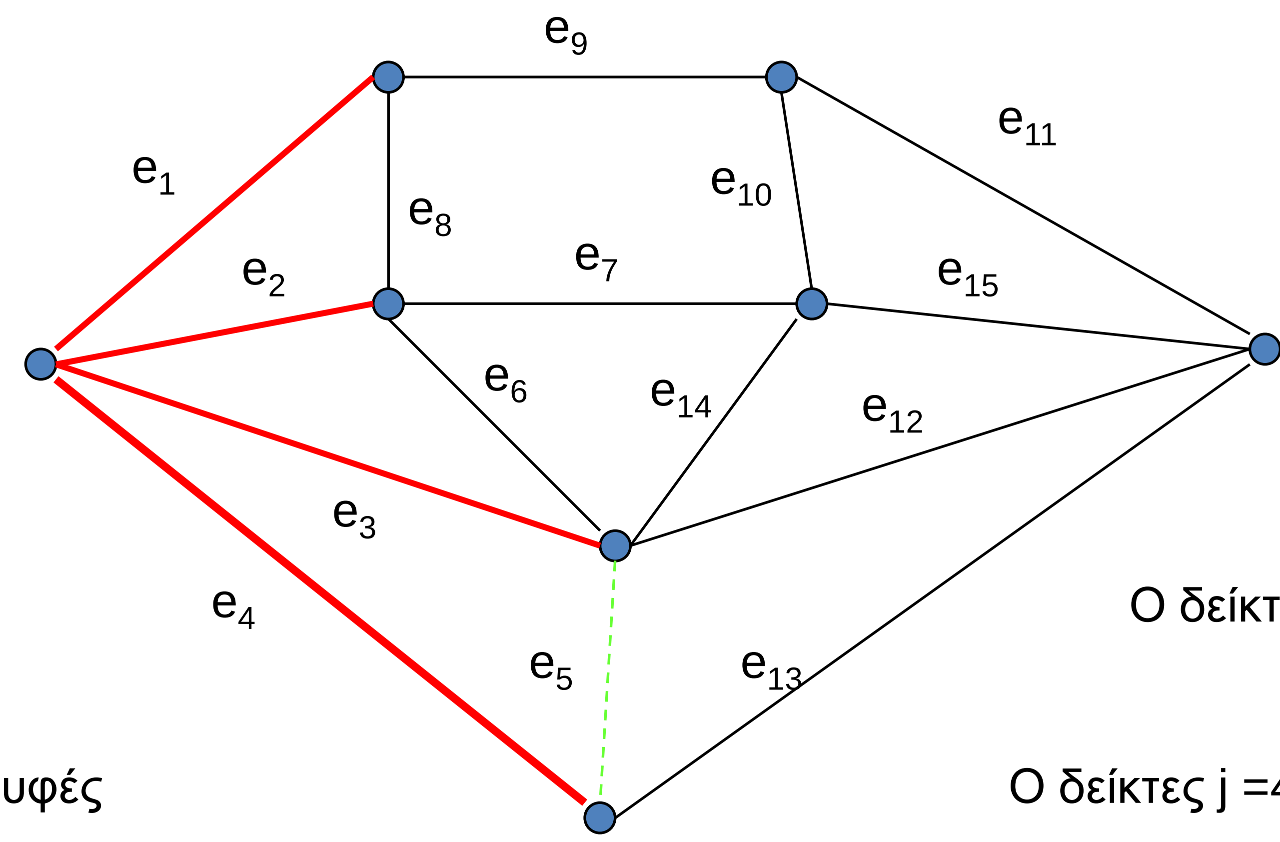
Ο δείκτες j =3

$L=\{e_1, e_2, e_3, e_4\}$

Πρόβλημα 1ο - Αλγόριθμος



Πρόβλημα 1ο - Αλγόριθμος



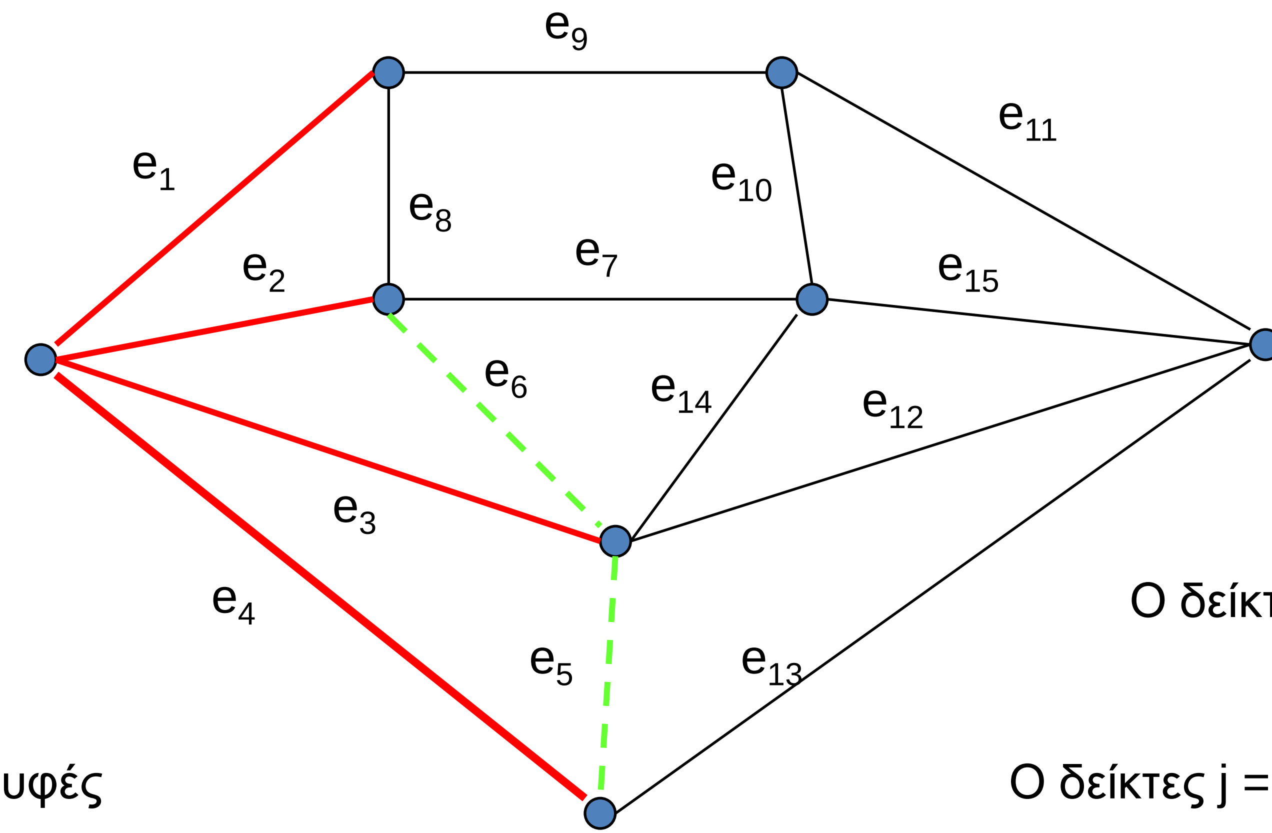
O δείκτες i =3

n=8 κορυφές

O δείκτες j =4

L={e_1, e_2, e_3, e_4}

Πρόβλημα 1ο - Αλγόριθμος



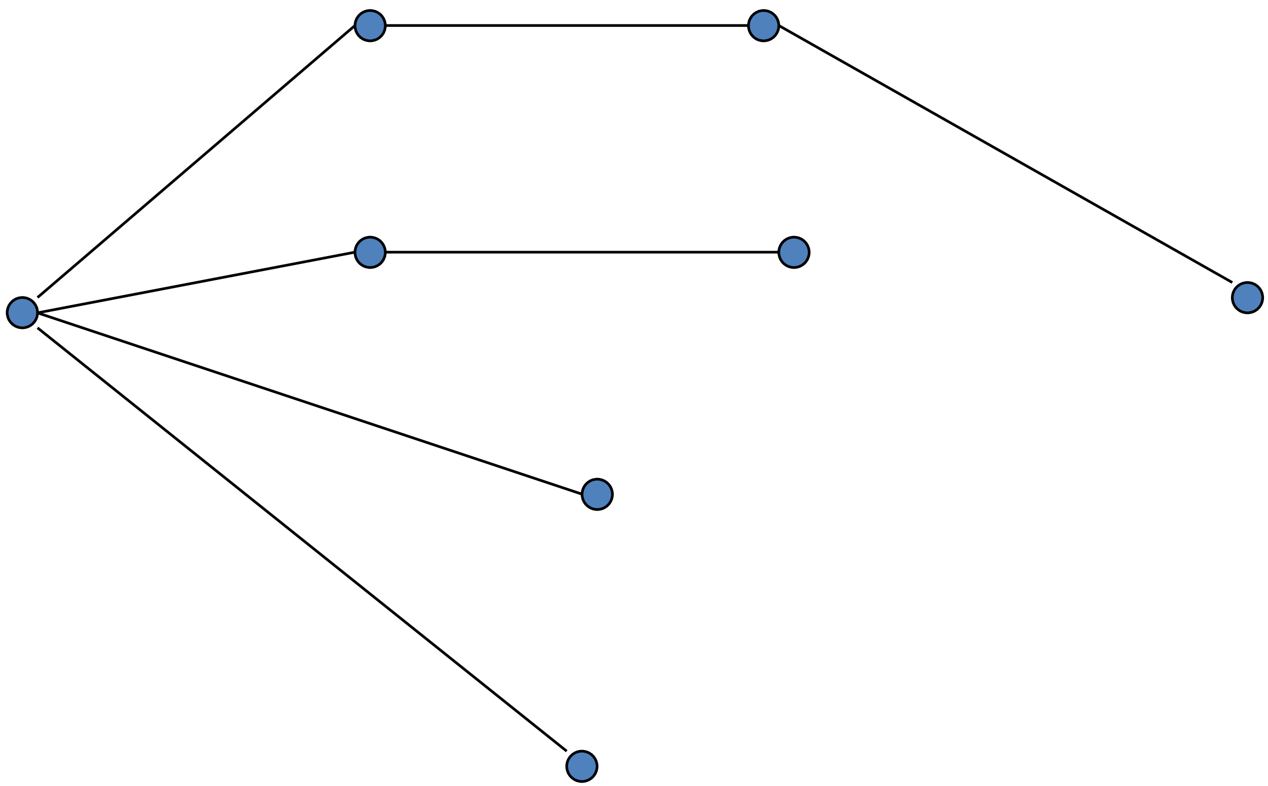
Ο δείκτες i =3

n=8 κορυφές

Ο δείκτες j =5

L={e_1, e_2, e_3, e_4}

Πρόβλημα 1ο - Αλγόριθμος



Χρήσιμοι υπερσύνδεσμοι

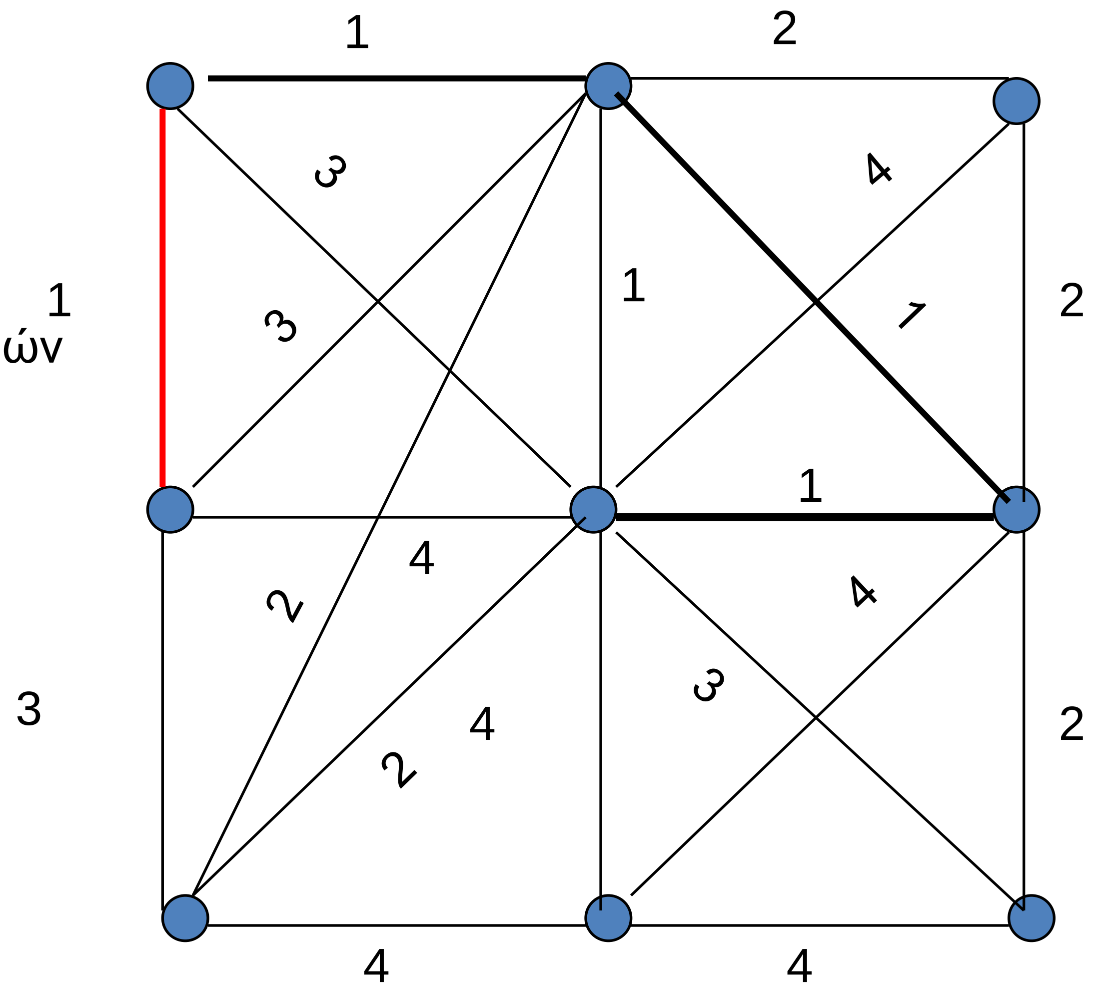
Jawa Γεννήτρια γράφων

Πρόβλημα 2ο - Αλγόριθμος Kruskal

1. Αριθμήστε τις ακμές του G, κατά την τάξη μεγέθους των βαρών.
2. Ονομάστε S το γράφο που αποτελείται μόνο από τις ακμές του G.
3. $i \leftarrow 1$
4. Αν, το S έχει n-1 ακμές ο έλεγχος στο 8.
5. Αν η προσάρτηση της ακμής e_i κλείνει κύκλωμα, ο έλεγχος στο 7.
6. Προσάρτησε στο S την e_i του G. Ονόμασε το νέο γράφο, S.
7. $i \leftarrow i+1$, ο έλεγχος στο 4.
8. "S είναι δένδρο σύνδεσης ελάχιστου βάρους". STOP

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Αριθμός ακμών
n=9



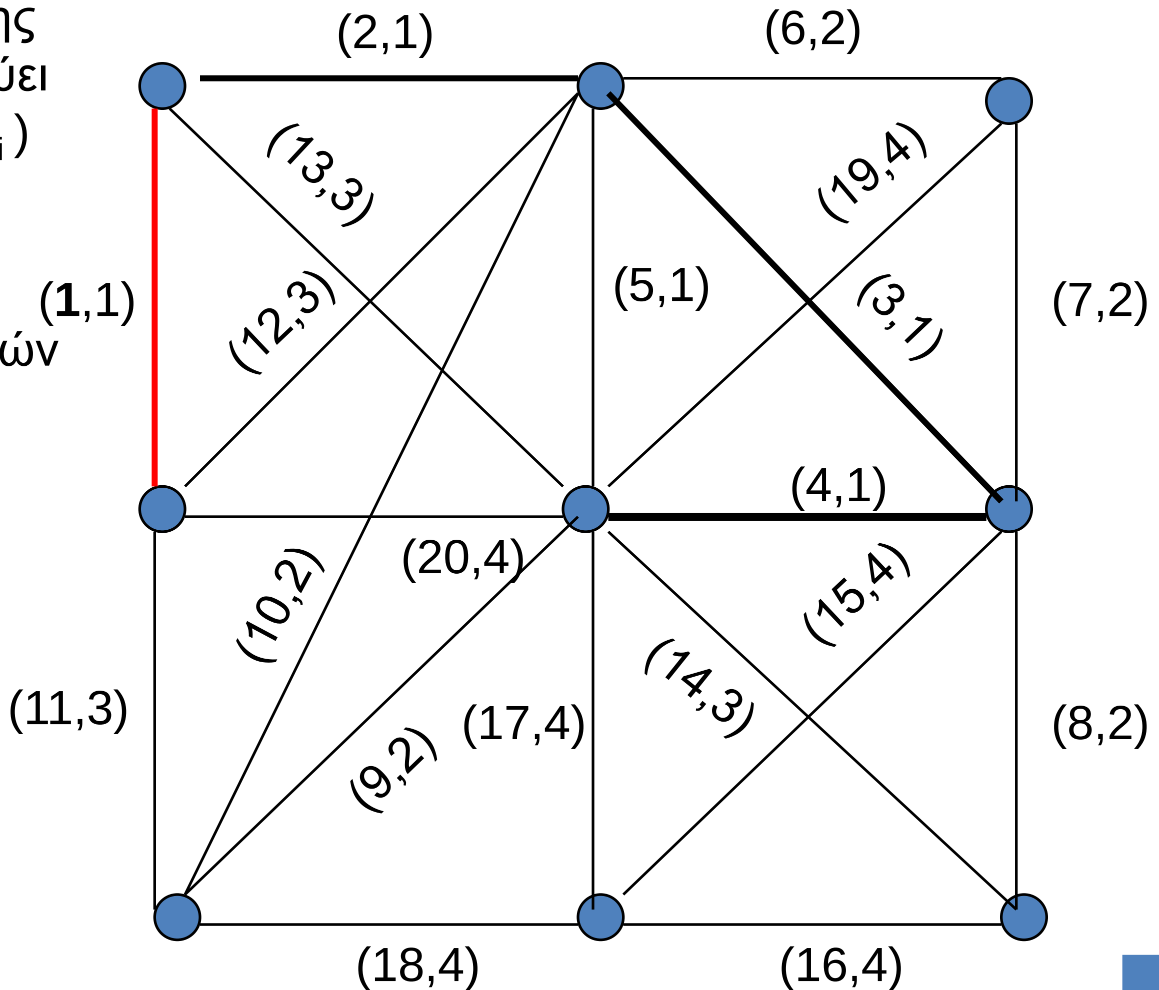
Έλεγχος:
1. Το δένδρο έχει n-1 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Εφαρμογή

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9



Έλεγχος:

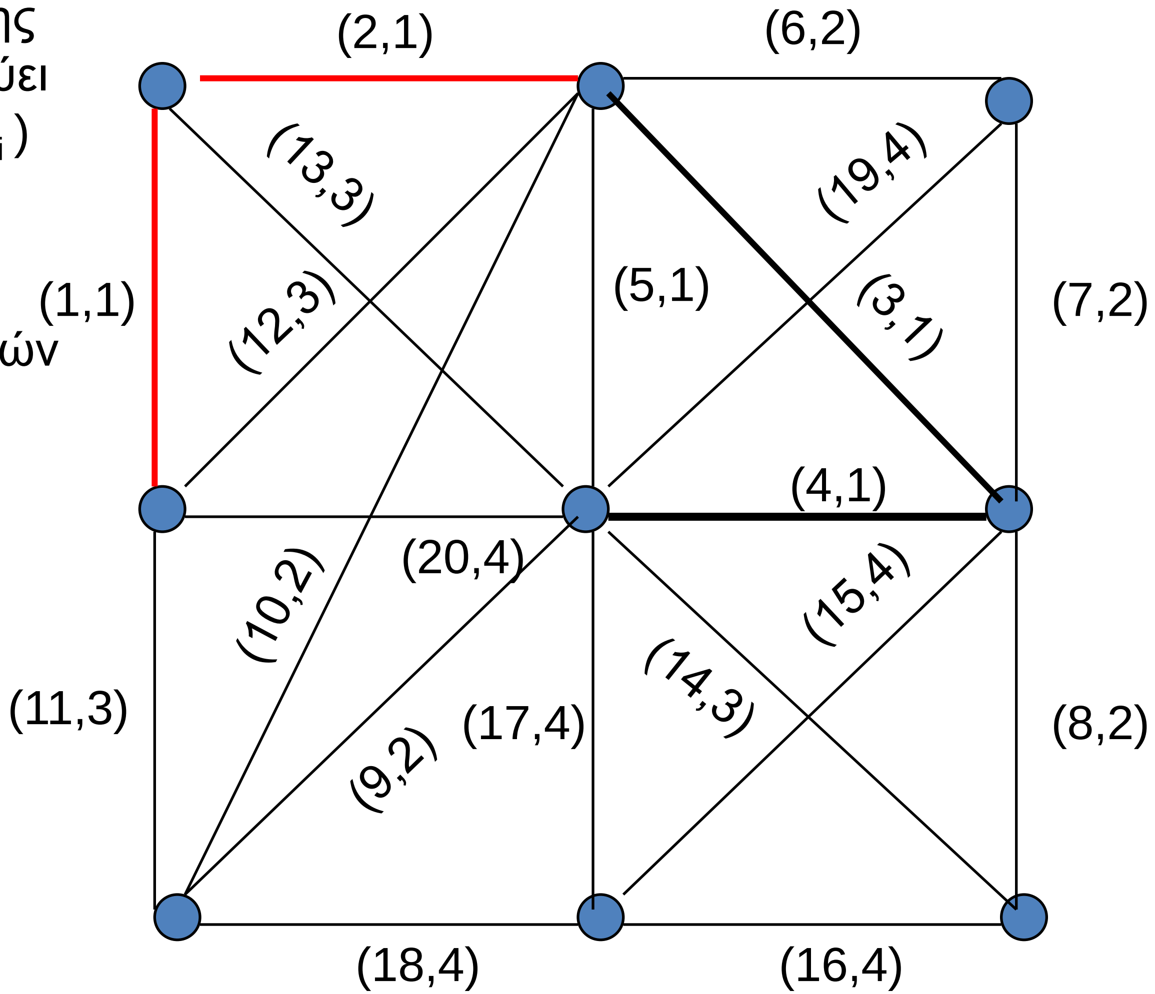
1. Το δένδρο έχει n-1 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Εφαρμογή

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9



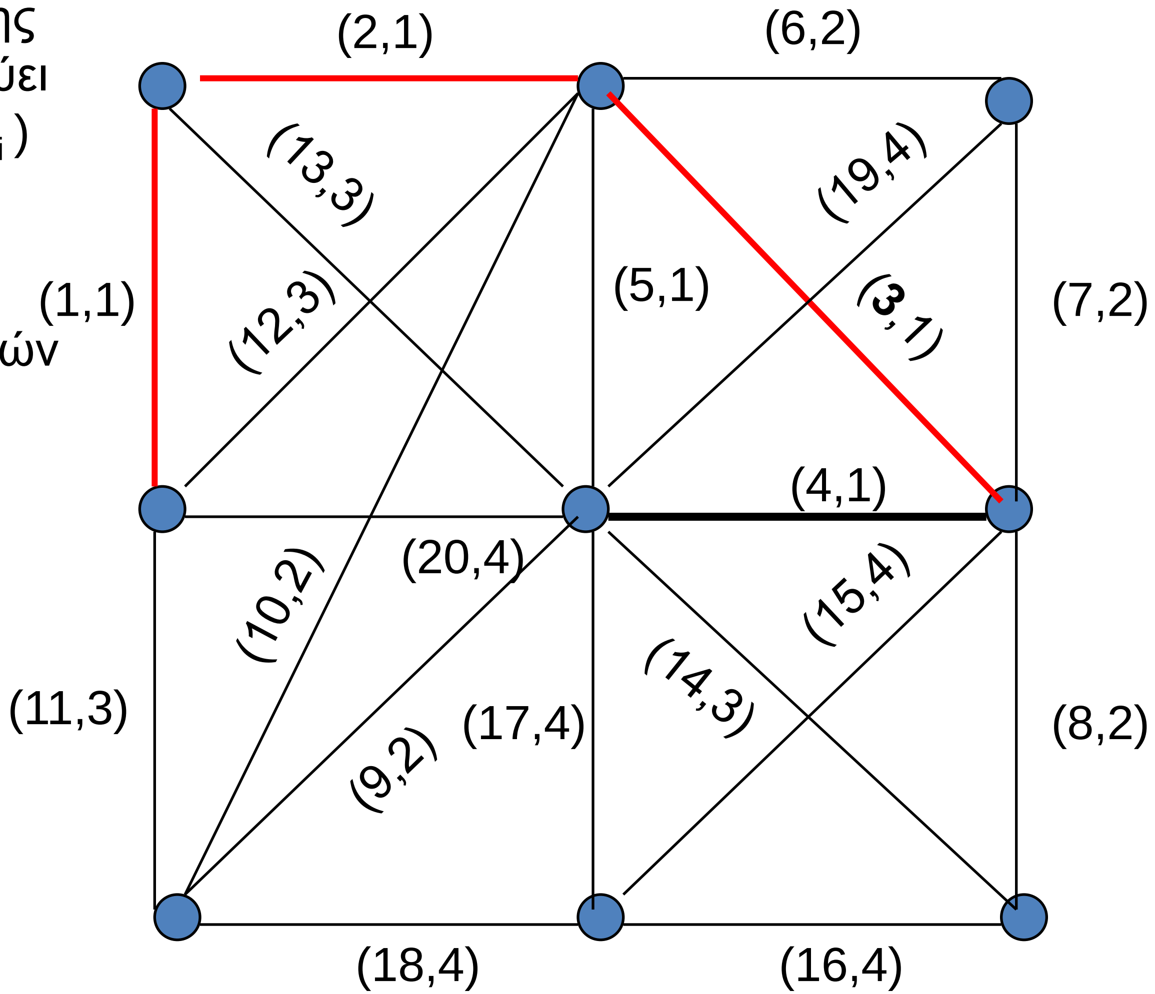
Έλεγχος:

1. Το δένδρο έχει n-1 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9



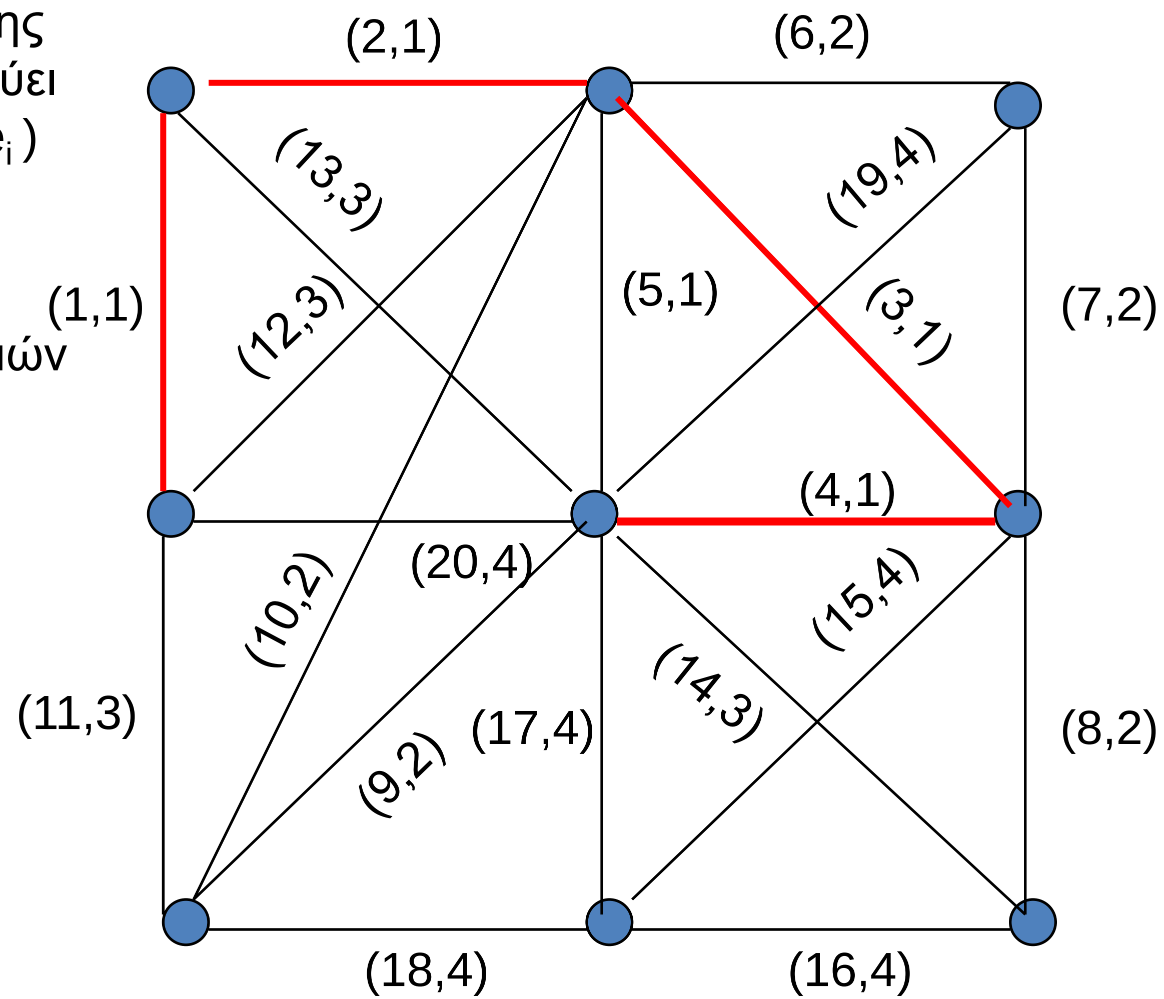
Έλεγχος:

1. Το δένδρο έχει n-1 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9

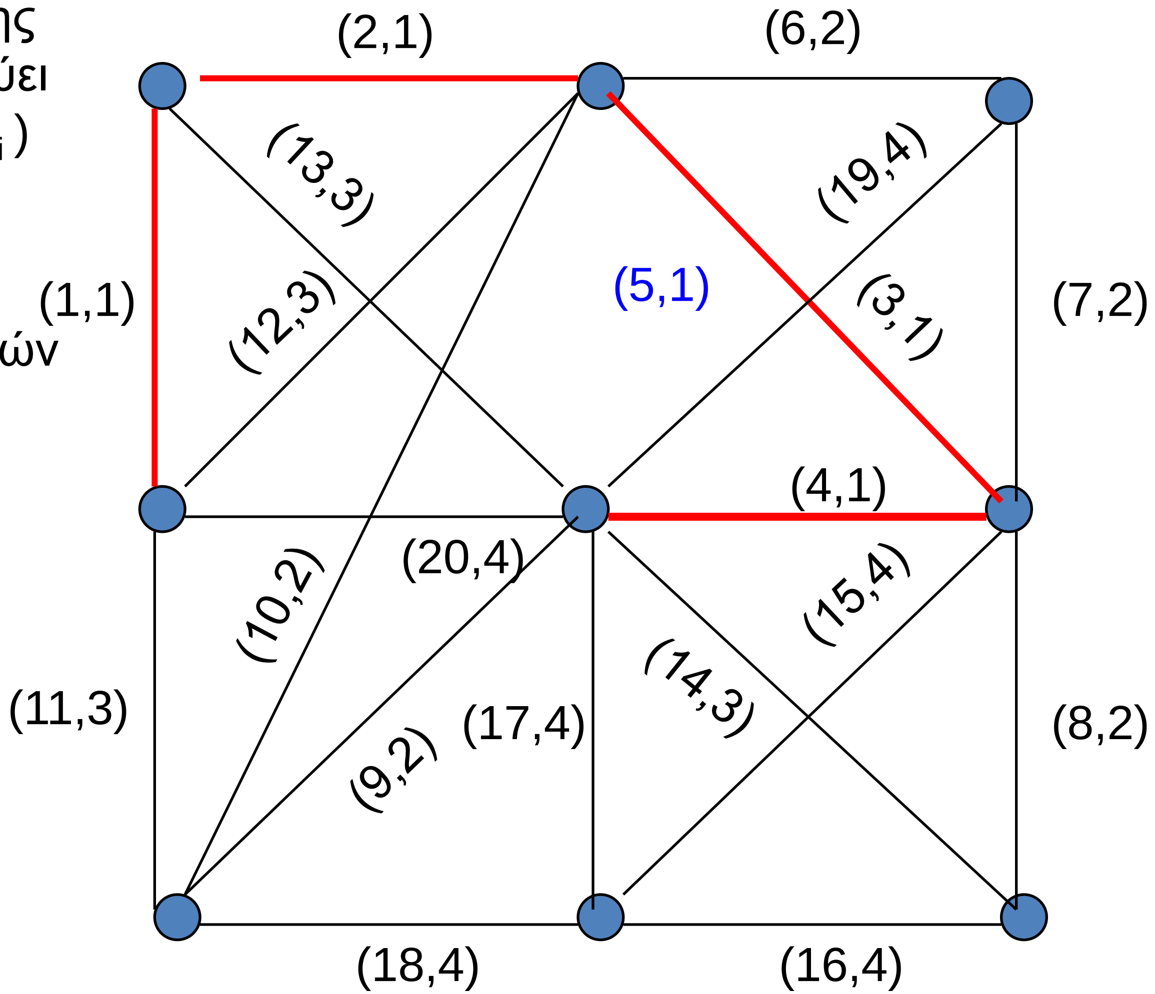


Έλεγχος:
1. Το δένδρο έχει n-1 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9

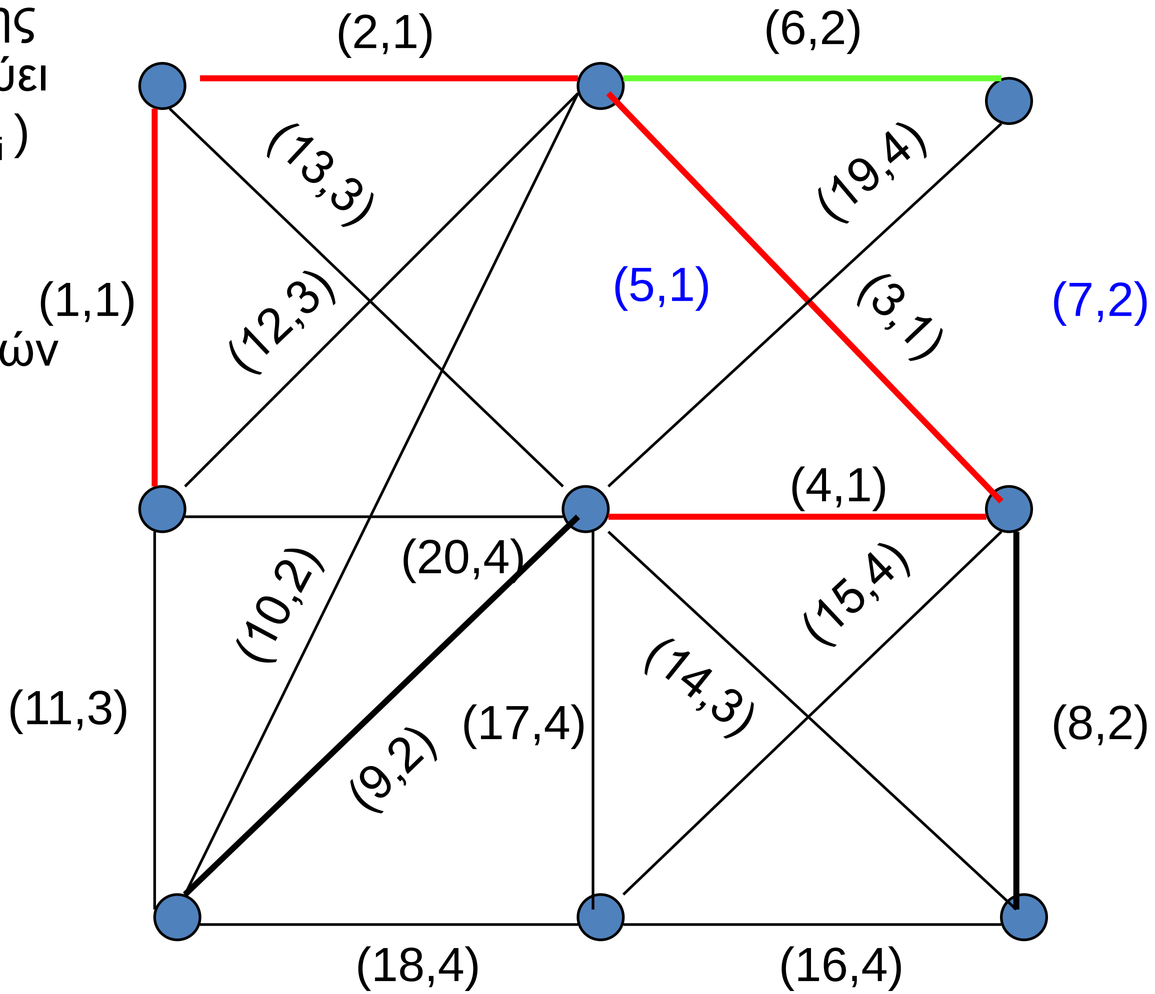


Έλεγχος:
1. Το δένδρο έχει n-1 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9



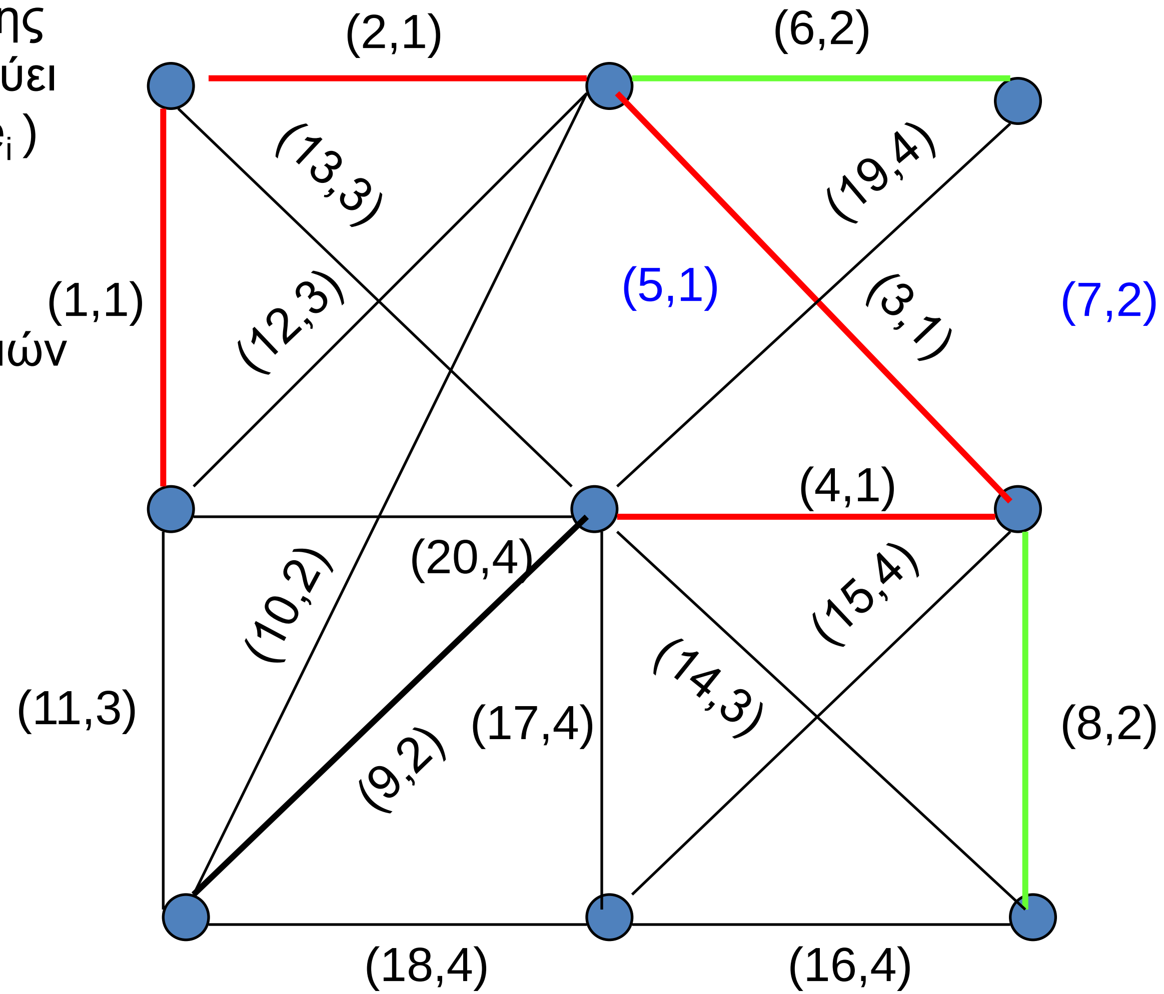
Έλεγχος:

1. Το δένδρο έχει 8 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9



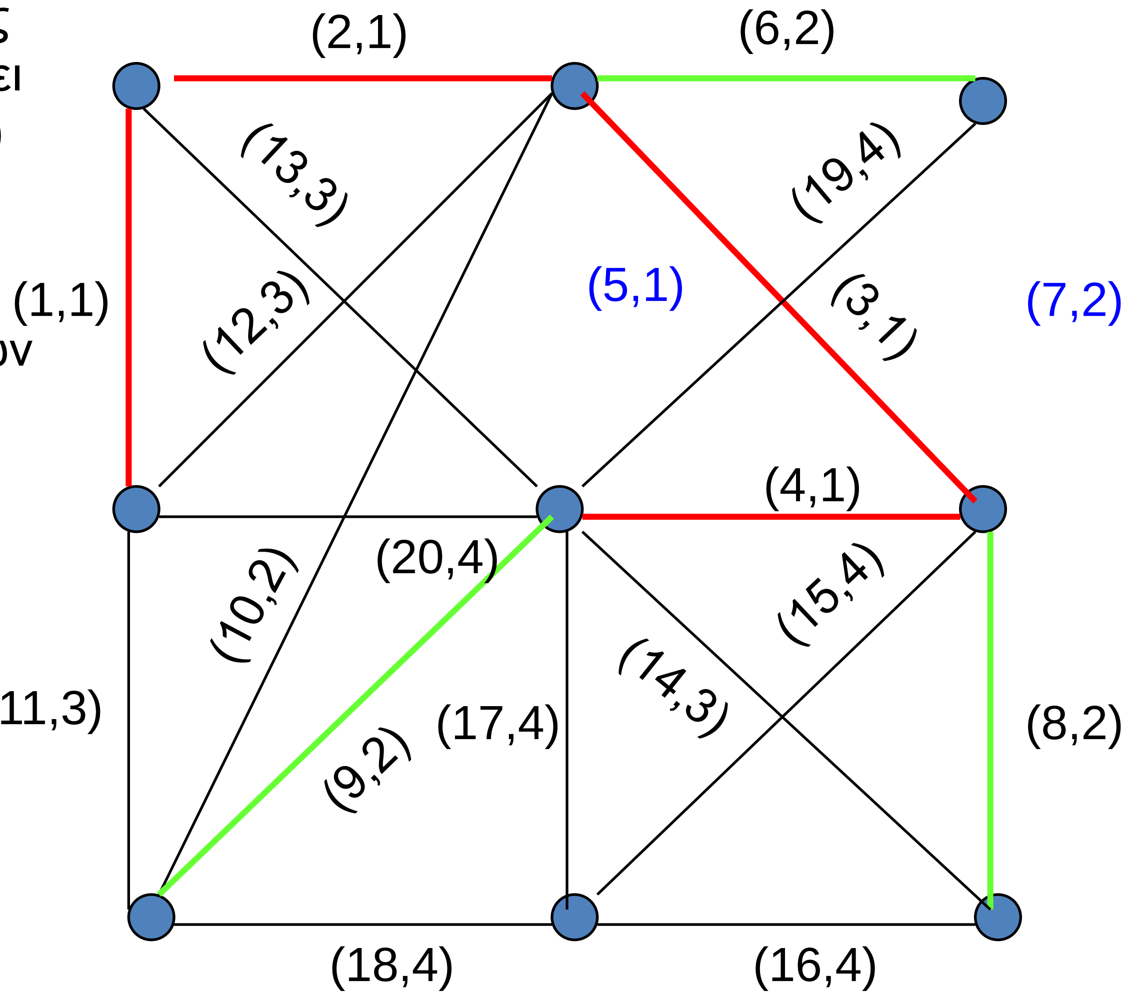
Έλεγχος:

1. Το δένδρο έχει 8 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9



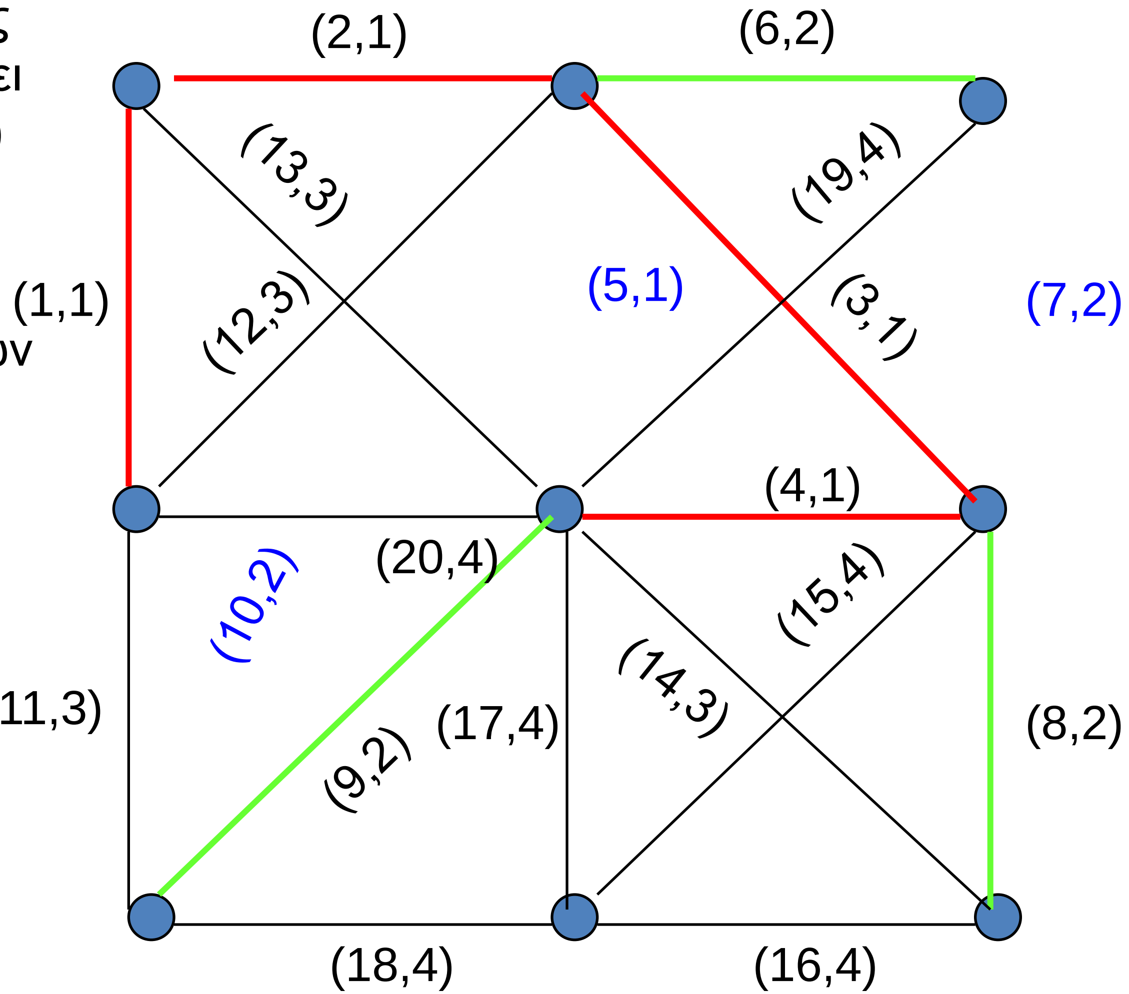
Έλεγχος:

1. Το δένδρο έχει 8 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9



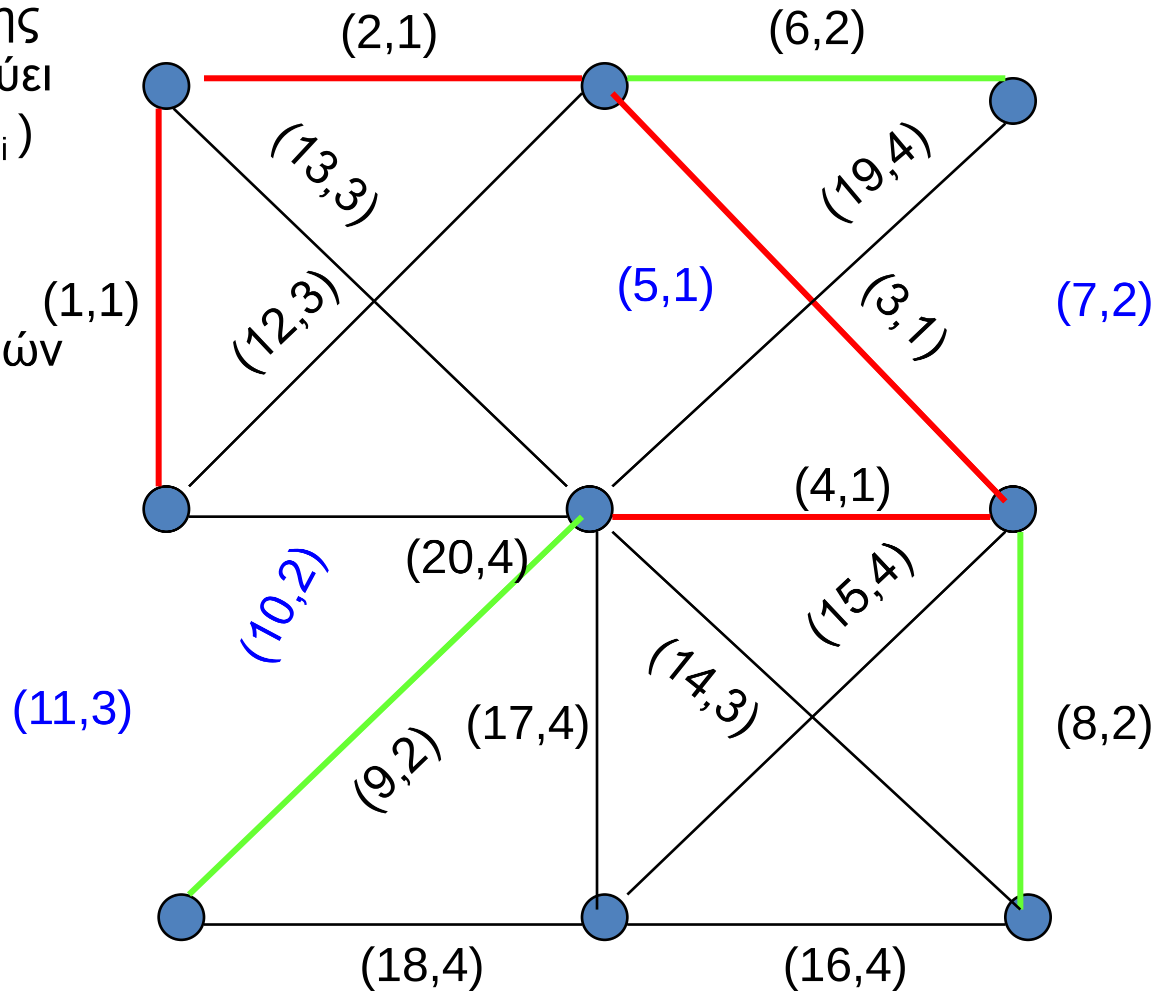
Έλεγχος:

1. Το δένδρο έχει 8 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9



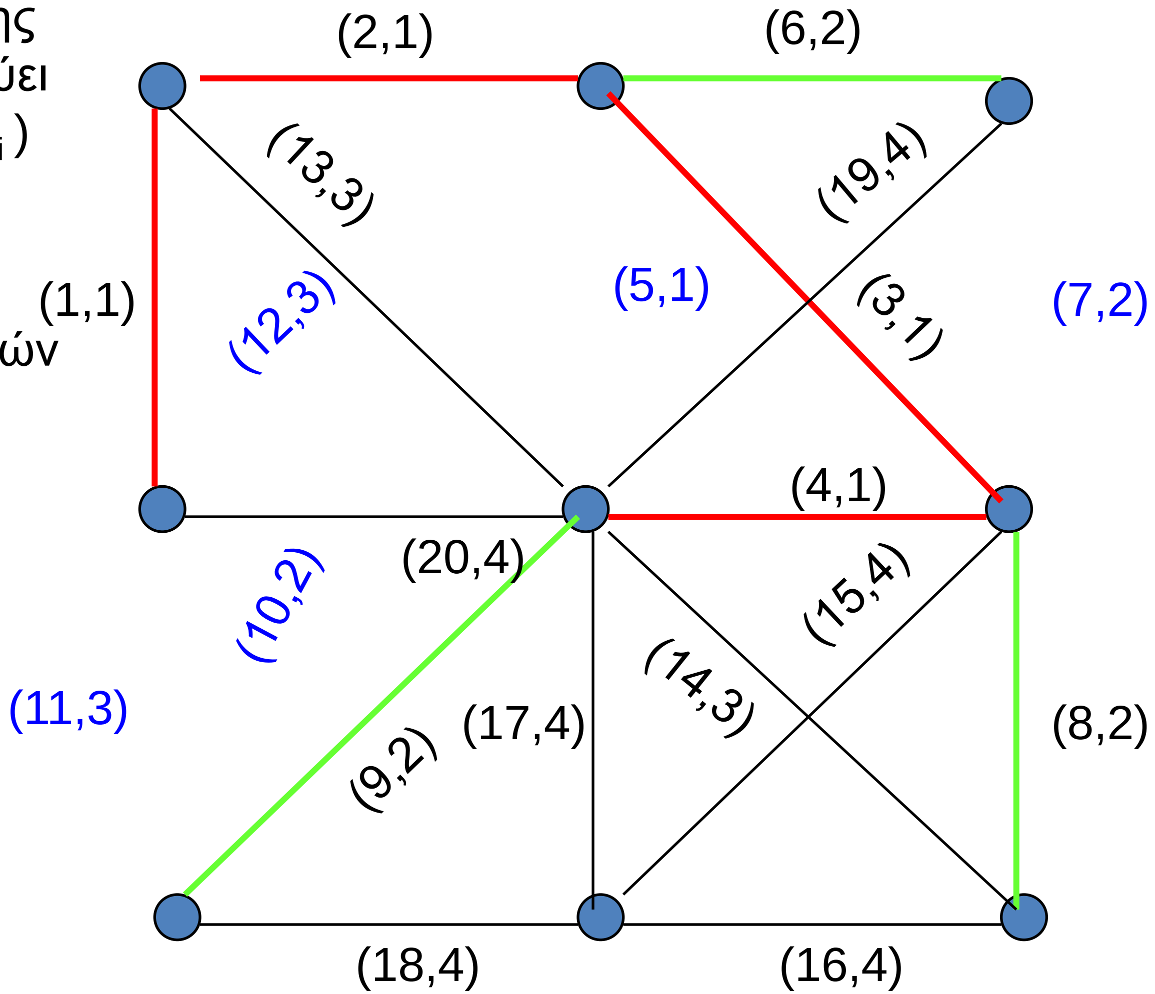
Έλεγχος:

1. Το δένδρο έχει 8 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9



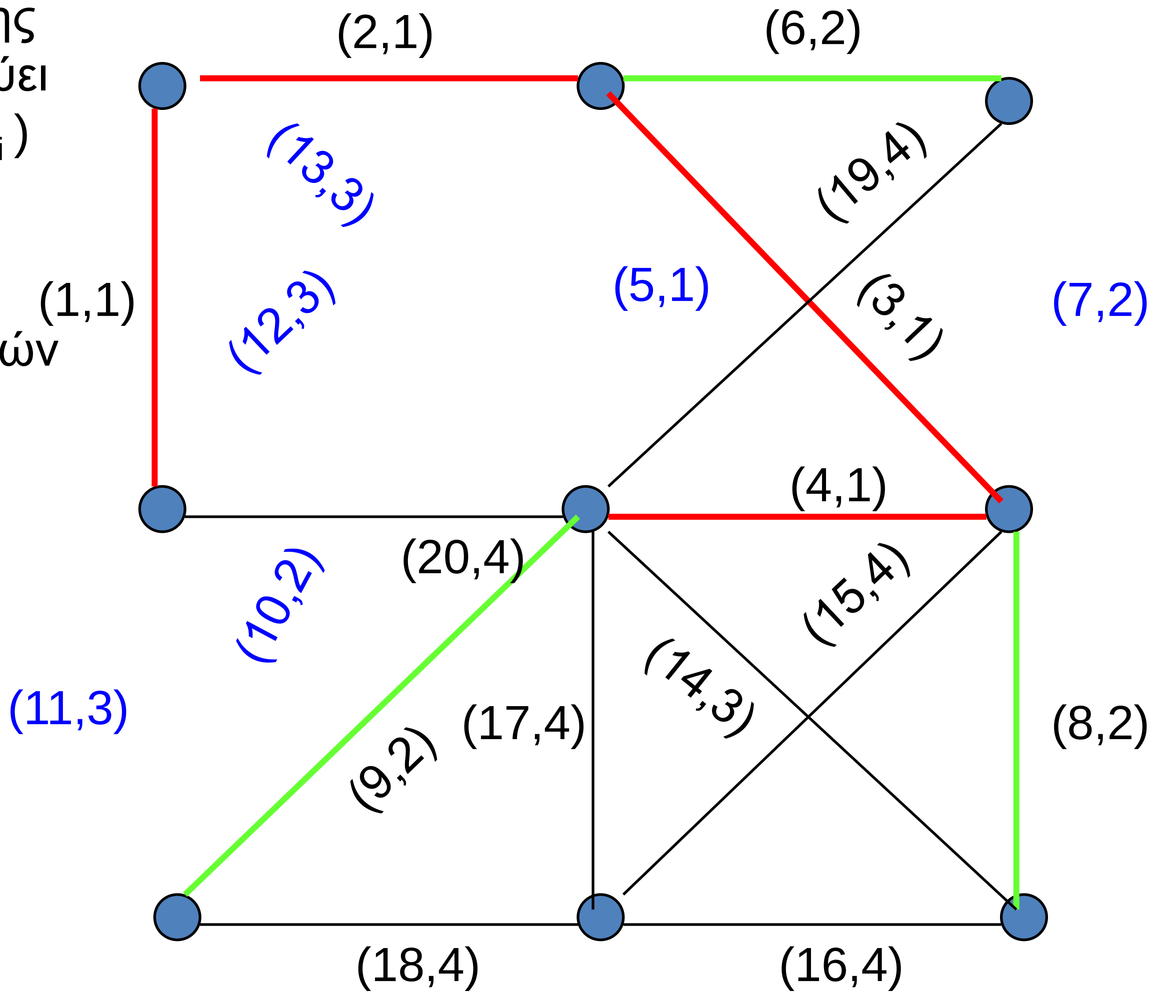
Έλεγχος:

1. Το δένδρο έχει 8 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9



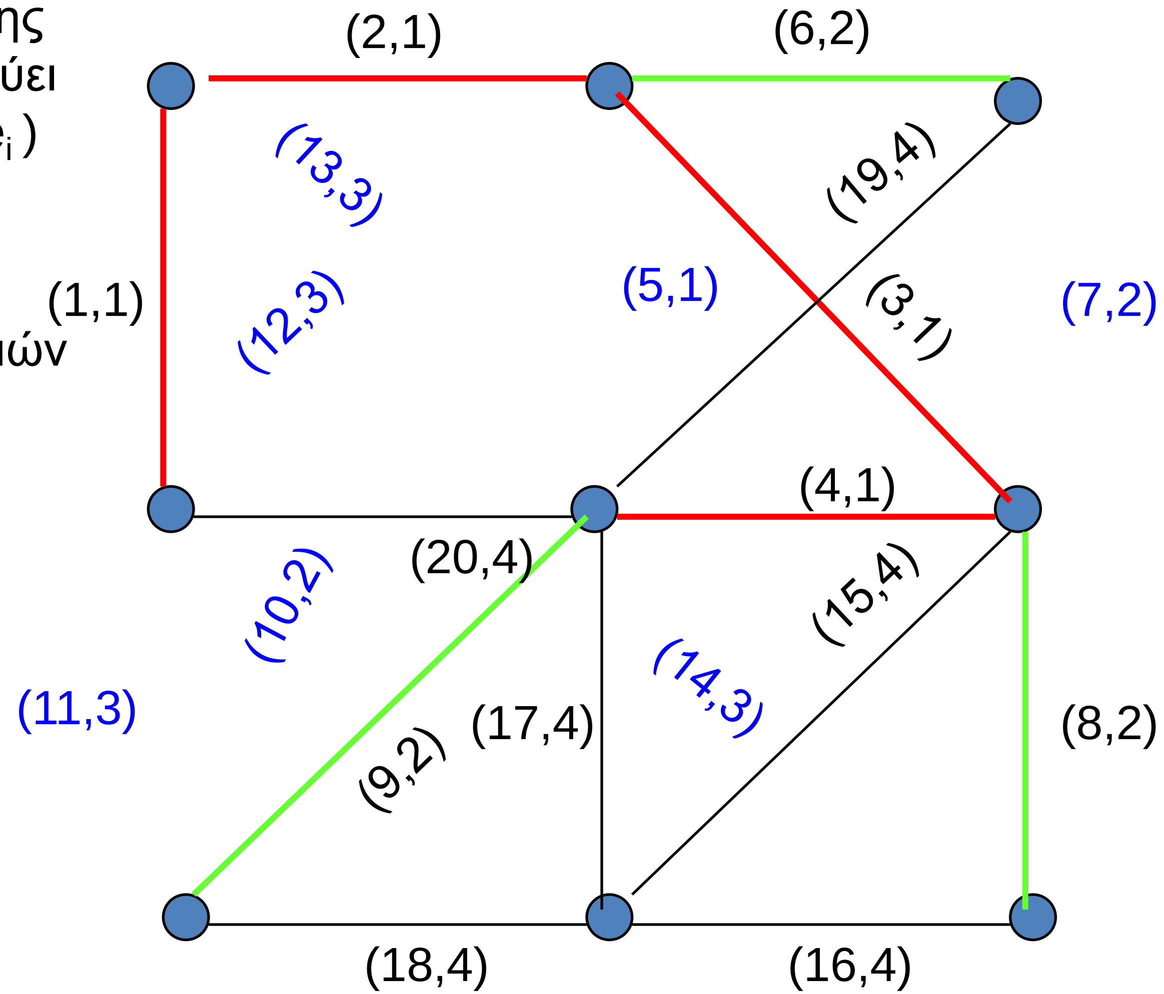
Έλεγχος:

1. Το δένδρο έχει 8 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9

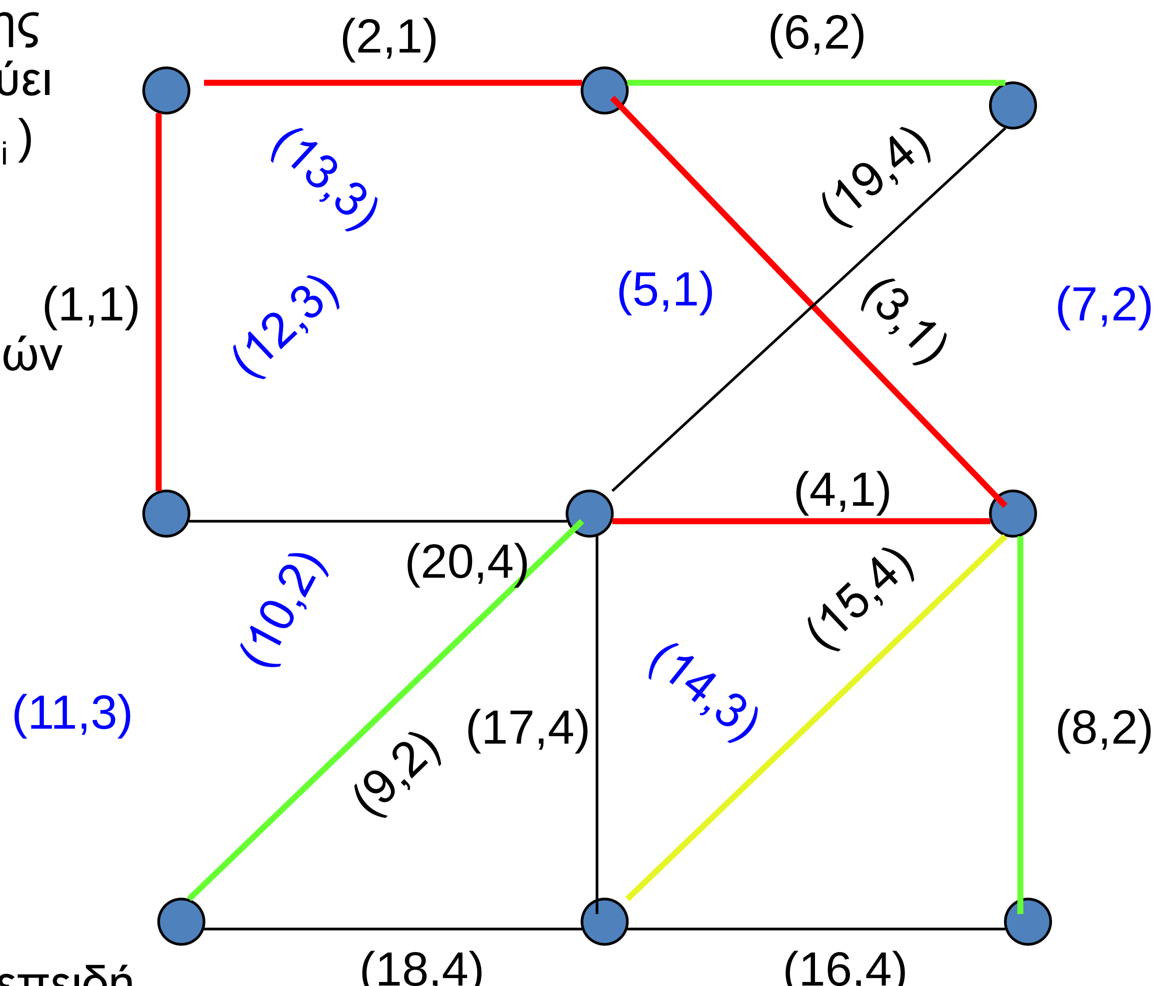


Έλεγχος:
1. Το δένδρο έχει 8 ακμές;
2. Οι νέες προσθήκες έκλεισαν κύκλωμα;

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9

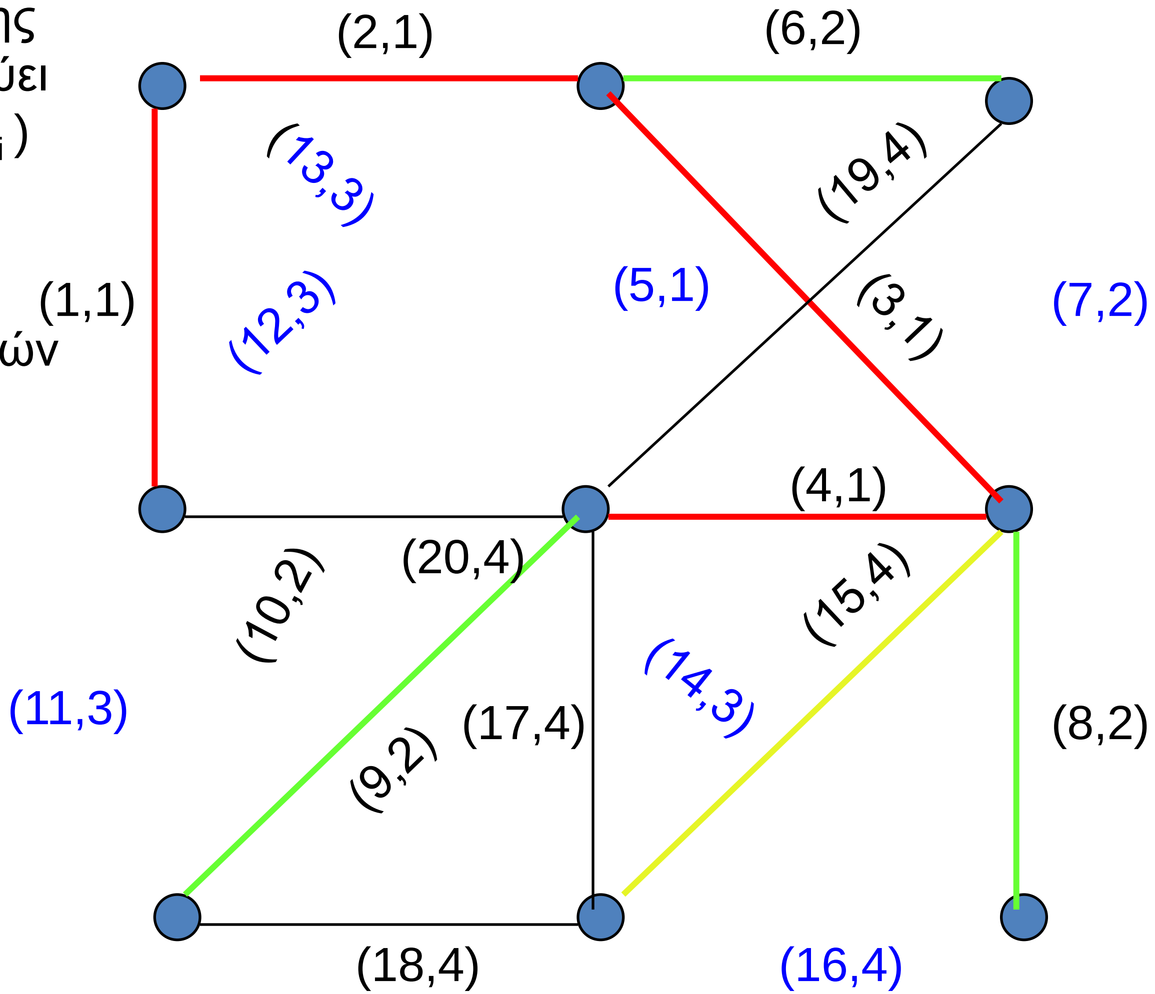


Στο σημείο αυτό επειδή
Το δένδρο έχει 8 ακμές
Σβήνουμε τις υπόλοιπες ακμές

Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

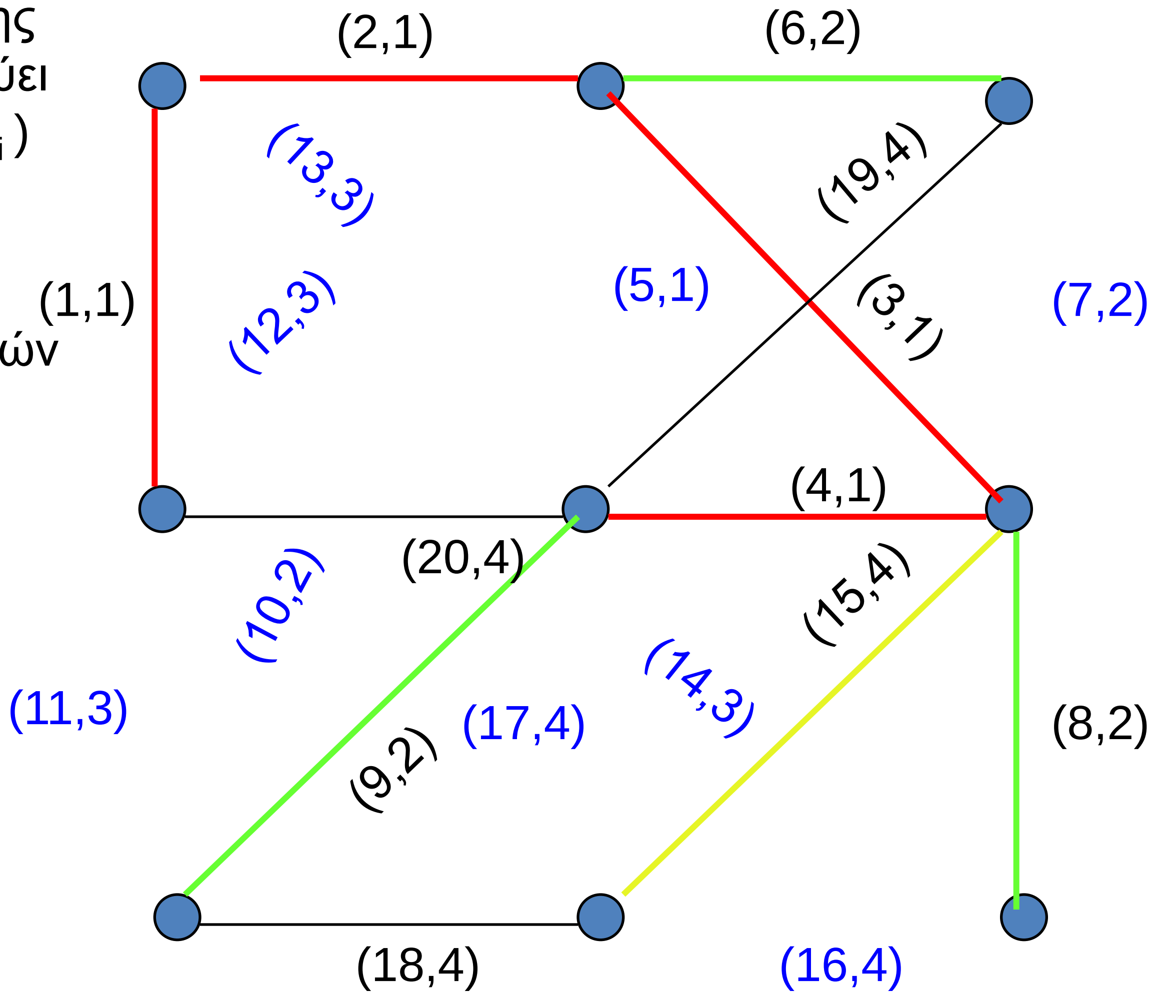
Αριθμός ακμών n=9



Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

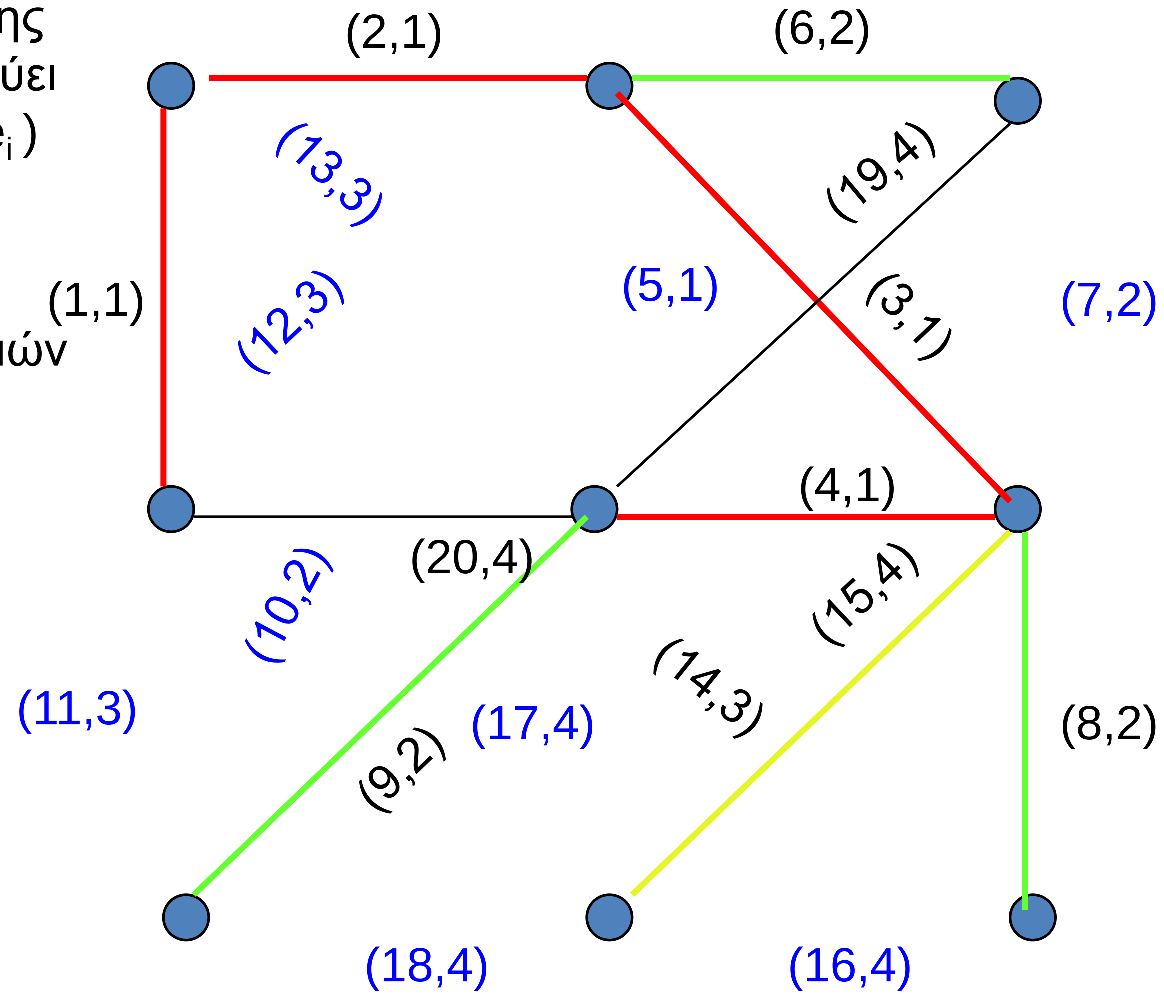
Αριθμός ακμών n=9



Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

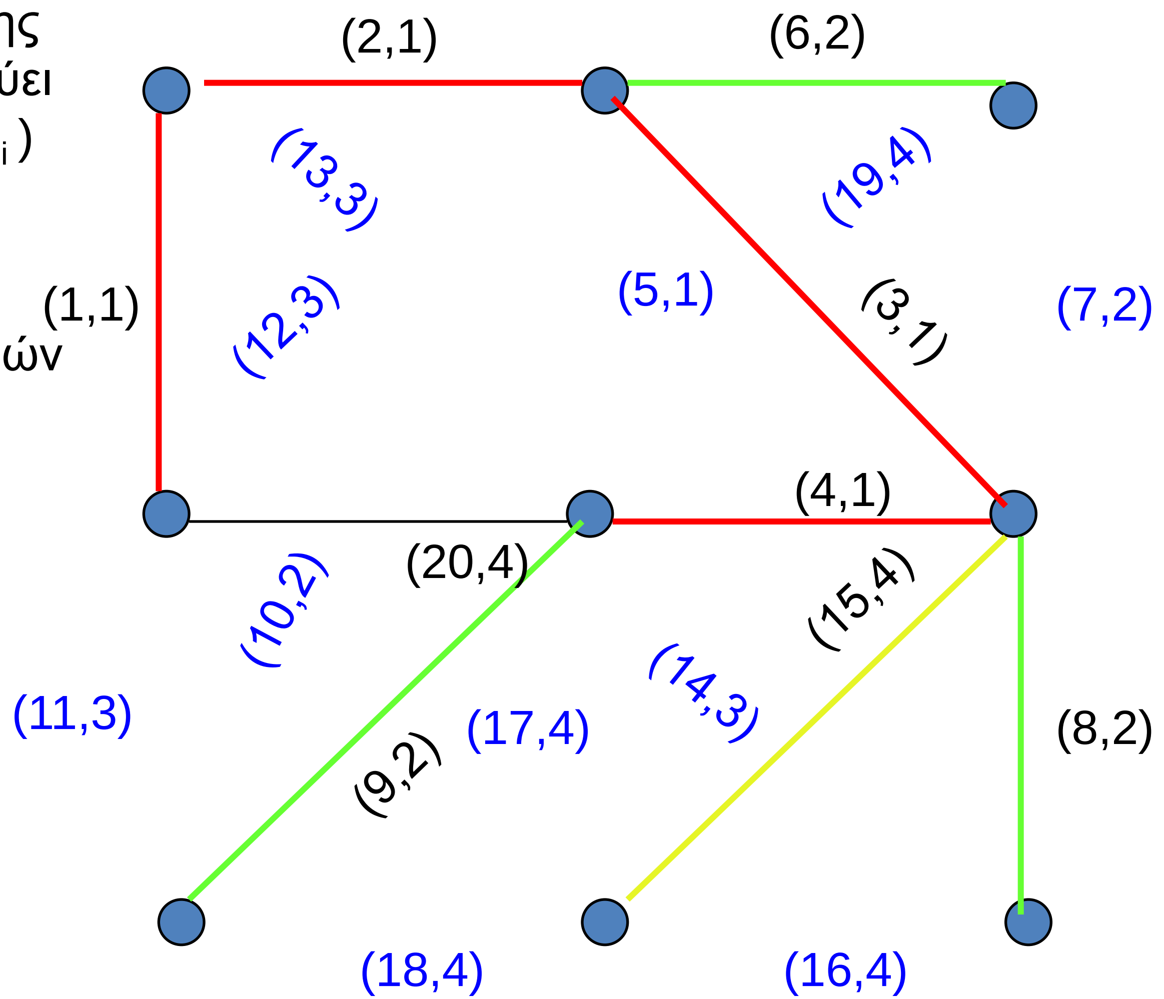
Αριθμός ακμών n=9



Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

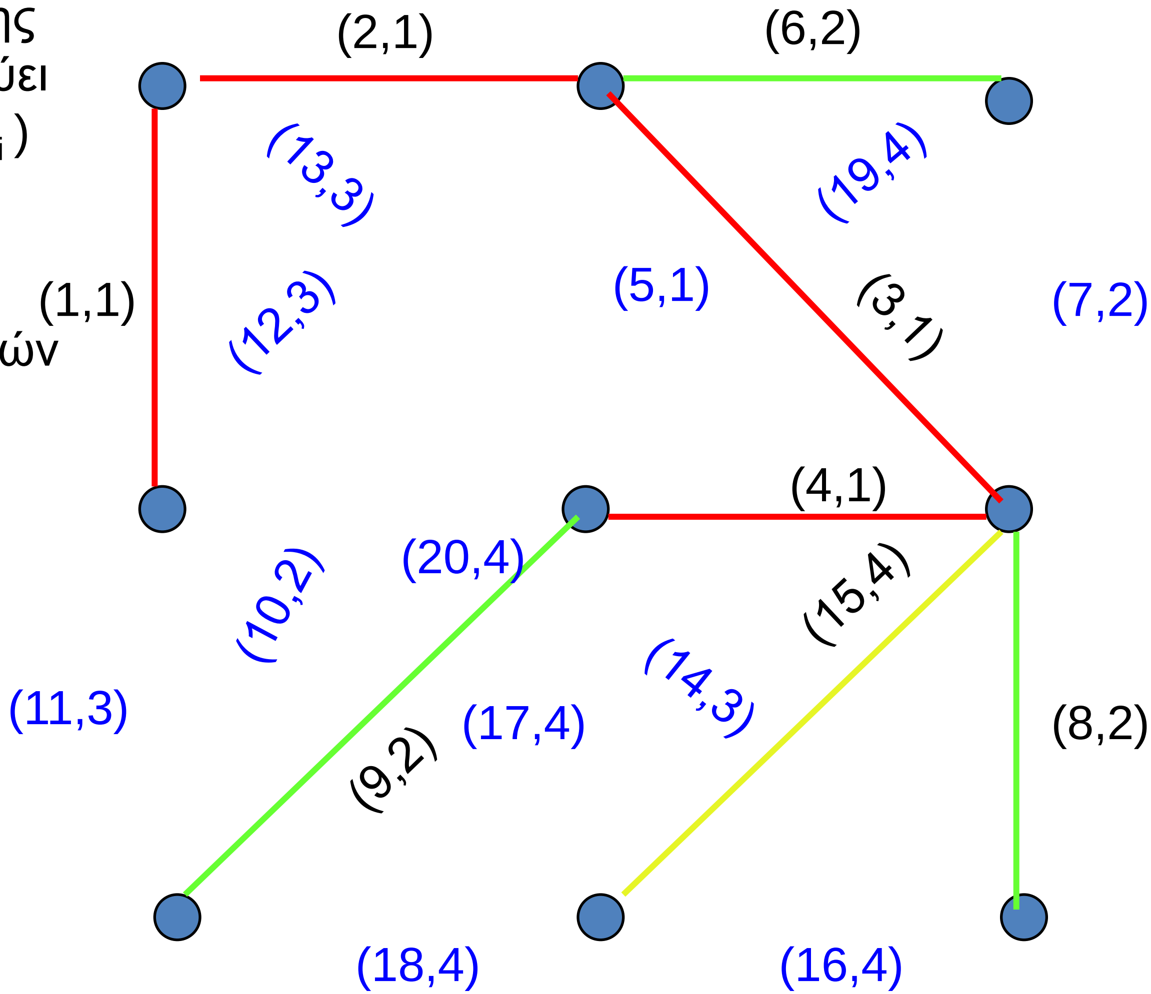
Αριθμός ακμών n=9



Πρόβλημα 2ο - Αλγόριθμος Kruskal

Στο γράφο της Εικόνας, ισχύει $w(e_{i+1}) \leq w(e_i)$

Αριθμός ακμών n=9



Υπολογισμός του πλήθους των δένδρων σύνδεσης ενός γράφου G

- Ο αριθμός t (G) των δέντρων που καλύπτουν ένα συνδεδεμένο γράφημα είναι μια σημαντική αναλλοίωτη.
- Σε ορισμένες περιπτώσεις, είναι εύκολο να υπολογίσει το t(G) άμεσα.
- Χρησιμοποιείται ευρύτατα σε δομές δεδομένων και σε διάφορες γλώσσες υπολογιστή.
- Μερική περίπτωση: αν G είναι δέντρο, προκύπτει ότι, t(G) = 1, ενώ αν G είναι η C_n κυκλικός γράφος με n κορυφές, τότε t (G) = n. Για κάθε γράφο G, το t(G) μπορεί να υπολογιστεί χρησιμοποιώντας την ορίζουσα του προσκείμενου πίνακα του Kirchhoff.

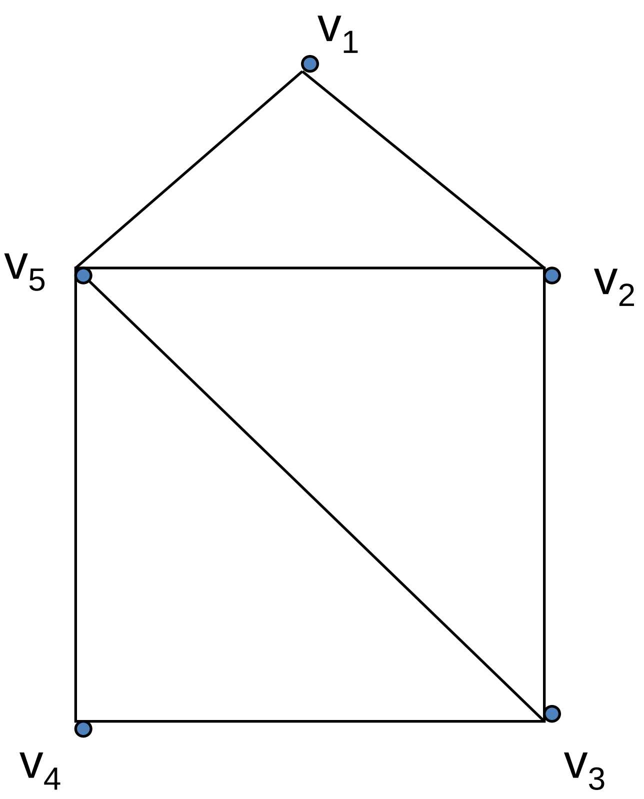
ΘΕΩΡΗΜΑ: Υπολογισμός του πλήθους t(G) των δένδρων σύνδεσης ενός γράφου G.

- Έστω A = $[\alpha_{ij}]_{n\times n}$ ο προσκείμενος πίνακας του γράφου G με κορυφές $v_1, v_2, \ldots, v_n$.
- Έστω B = $[b_{ij}]_{n\times n}$ ο πίνακας με στοιχεία $b_{ij} = -\alpha_{ij}$ όταν $i\neq j$ και $b_{ii} = \deg(v_i)$.
- Έστω S ο $[n-1] \times [n-1]$ πίνακας που αντιστοιχεί στο στοιχείο b_{11}
- Τότε, det [S] είναι το πλήθος των δένδρων σύνδεσης του γράφου

Παράδειγμα

- Ο προσκείμενος A είναι

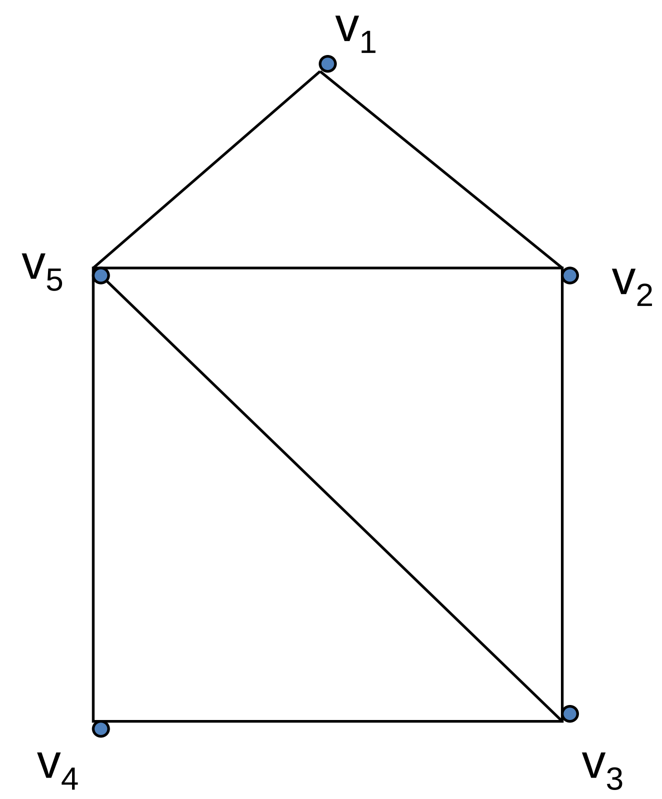
$$\begin{matrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{matrix}$$



Παράδειγμα

- Ο B είναι

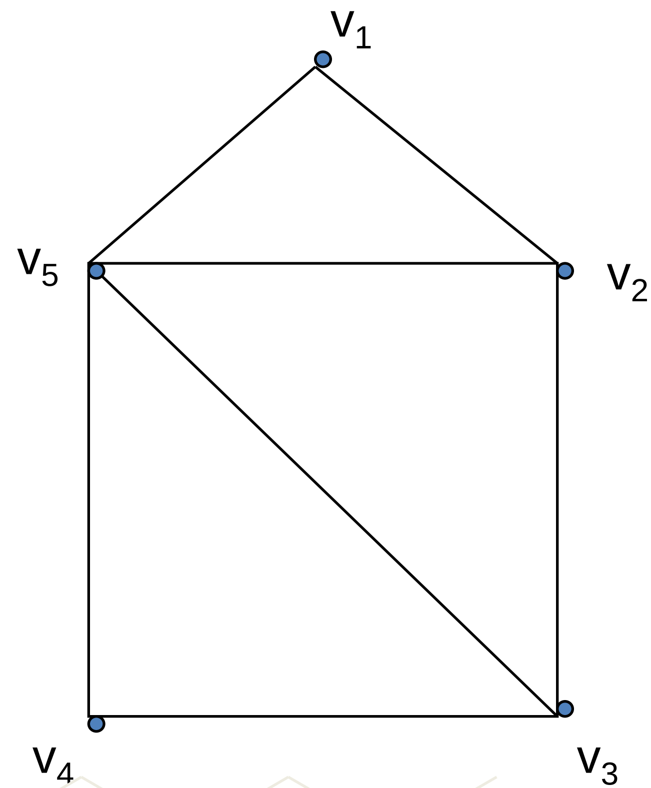
2	-1	0	0	-1
-1	3	-1	0	-1
0	-1	3	-1	-1
0	0	-1	2	-1
-1	-1	-1	-1	4



Παράδειγμα

- Ο υποπίνακας S, που αντιστοιχεί στο $b_{5,5}$ είναι

2	-1	0	0
-1	3	-1	0
0	-1	3	-1
0	0	-1	-2



2	-1	0	0
0	5/2	-1	0
0	0	13/5	-1
0	0	0	21/13

- Det(S)=21

Υπολογισμός του πλήθους των δένδρων σύνδεσης ενός γράφου

- Τύπος Cayley είναι μια φόρμουλα για τον αριθμό των δένδρων που καλύπτουν το πλήρες K_n γράφημα με n κορυφές. Ο τύπος αναφέρει ότι η $t(K_n) = n^{n-2}$. Ένας άλλος τρόπος δηλώνοντας τύπος Cayley είναι ότι υπάρχουν ακριβώς n^{n-2} επισημαίνοντα δέντρα με n κορυφές. Τύπος Cayley μπορεί να αποδειχθεί με τη βοήθεια πίνακα του Kirchhoff, ή μέσω του κώδικα Prüfer.

- Αν G είναι πλήρης διμερής $K_{p,q}$, τότε $t(G) = p^{q-1} q^{p-1}$.

- Αν G είναι υπερκύβος διάστασης n Q_n, τότε

- Οι τύποι αυτοί επίσης συνέπειες του προηγούμενου Θεωρήματος.