

Σχεδίαση και Ανάλυση Αλγορίθμων

Ενότητα 7.0

Αλγόριθμοι Γραφημάτων

Διερεύνηση Γραφημάτων – BFS & DFS

Τοπολογική Ταξινόμηση

Συνεκτικές και Ισχυρές Συνεκτικές Συνιστώσες



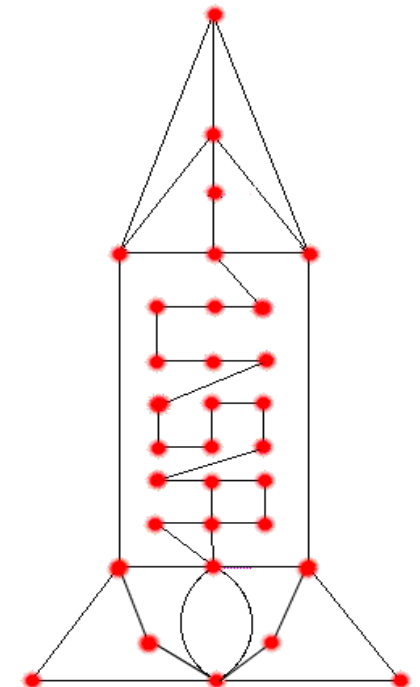
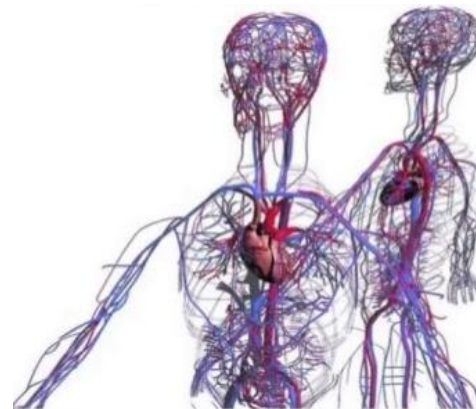
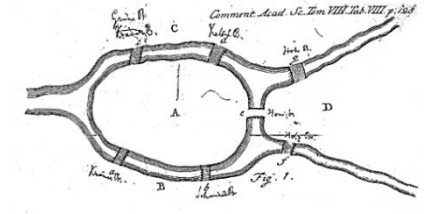
Σταύρος Δ. Νικολόπουλος | 2016-17

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

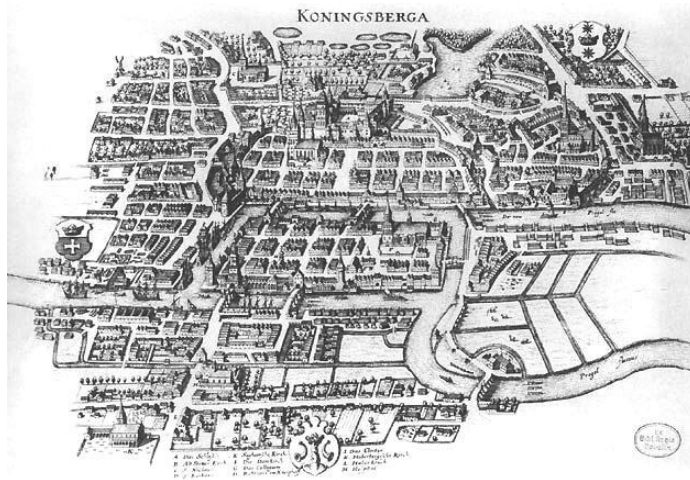
Πανεπιστήμιο Ιωαννίνων

Webpage: www.cs.uoi.gr/~stavros

Διερεύνηση Γραφημάτων



Γραφήματα - Πως ξεκίνησαν όλα !!!

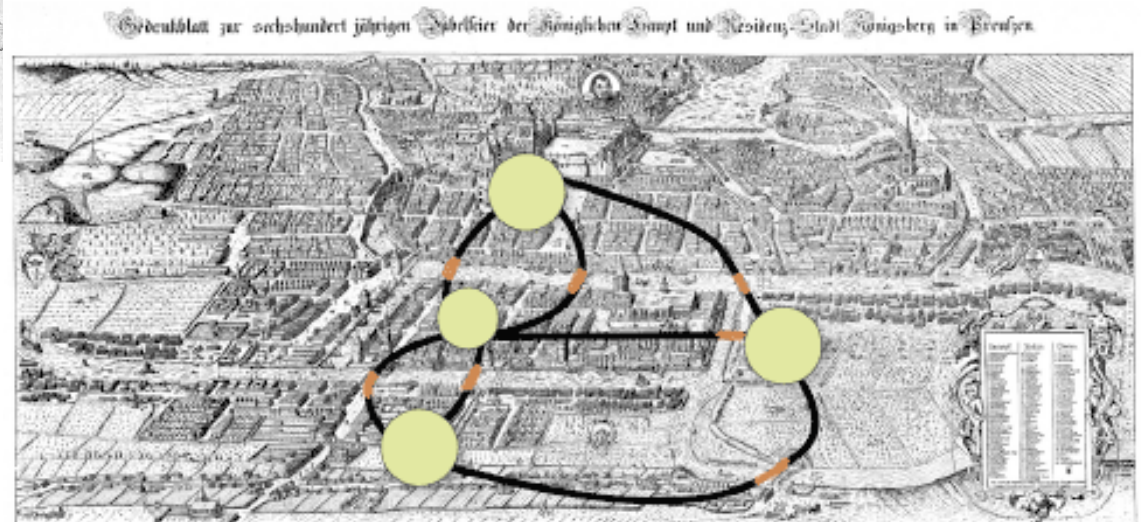
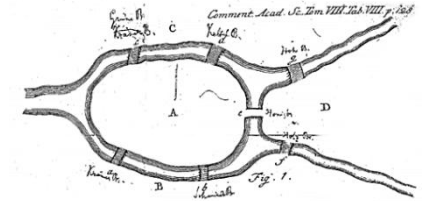


1736
Γέφυρες Königsberg

Σημερινό Ρωσικό Kaliningrad
(στη Βαλτική μεταξύ Λιθουανίας και Πολωνίας)

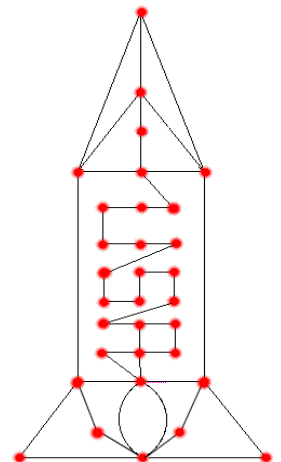
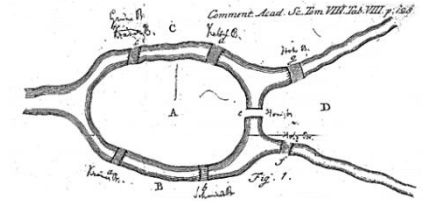


Leonhard Euler (1707-1783)

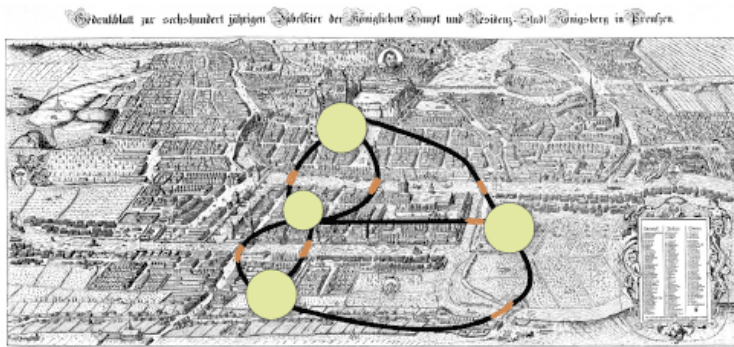


Γραφήματα - Εξέλιξη !!!

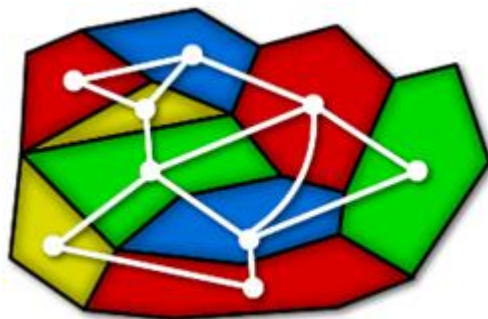
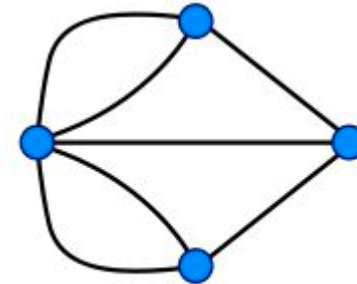
- 1736 Euler, γέφυρες Koenigsburg
- 1847 Kirchhoff, δένδρα, ηλεκτρικά δίκτυα
- 1847 Cayley, δένδρα, ισομερή υδρογονανθράκων C_nH_{2n+2}
- 1850 Cayley - De Morgan - Moebius, 4 χρωματισμός
- 1859 Hamilton, δωδεκάεδρο
- 1936 Το πρώτο βιβλίο
- Σήμερα ...



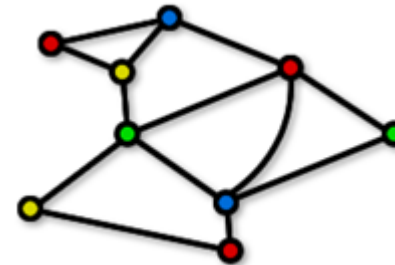
Μοντελοποίηση



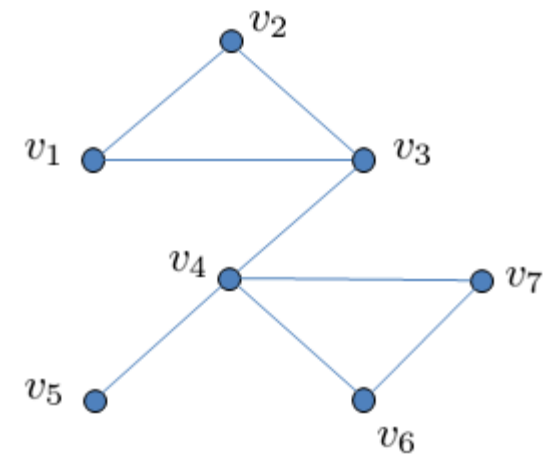
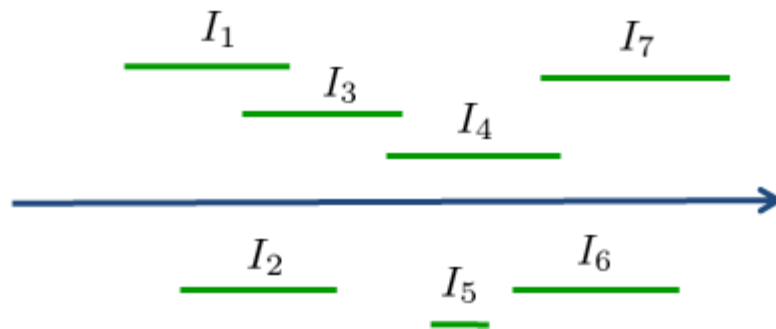
=



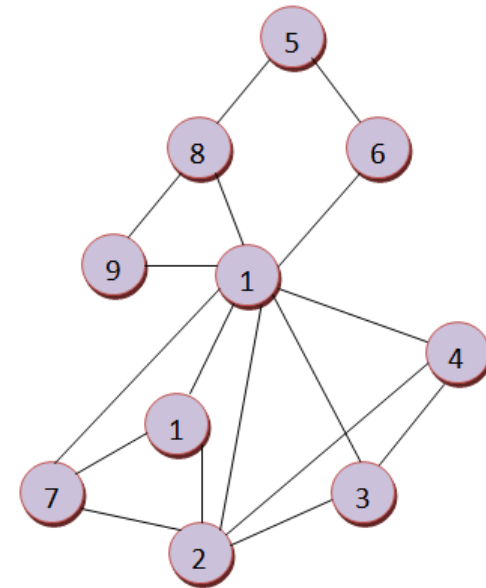
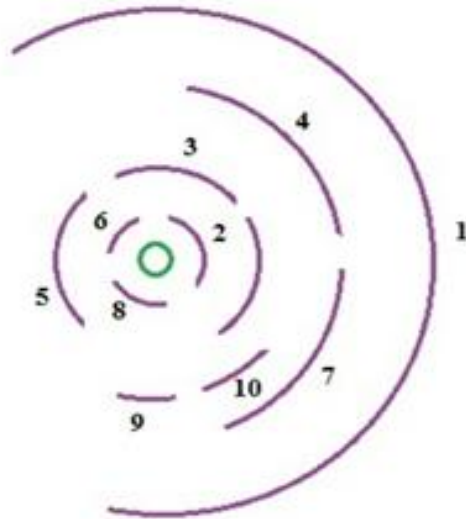
=



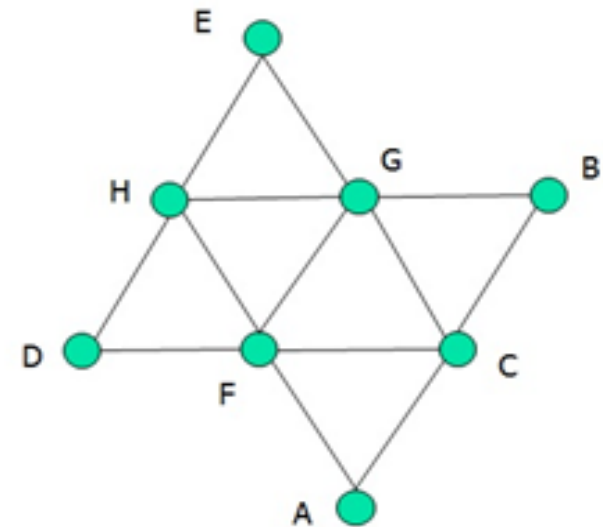
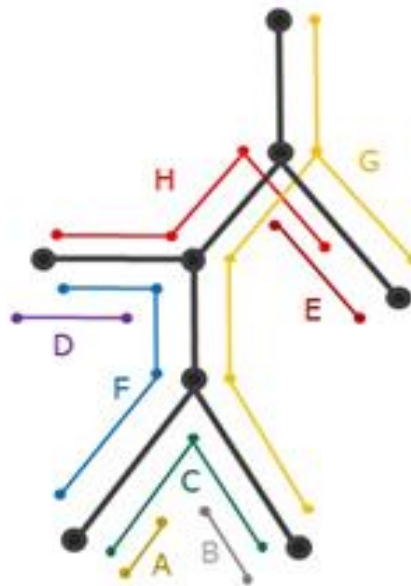
Μοντελοποίηση



Μοντελοποίηση

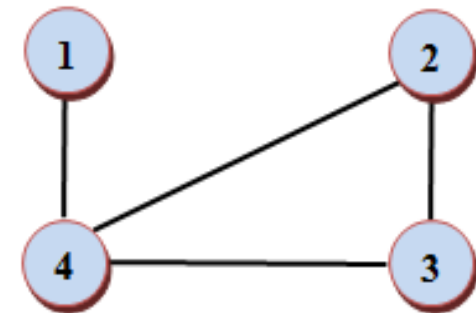
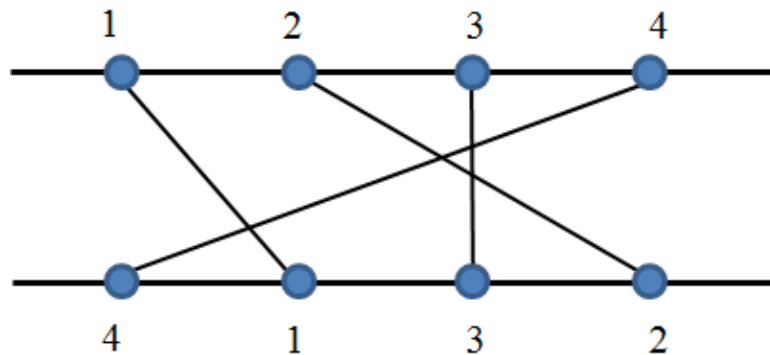


Μοντελοποίηση

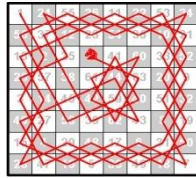
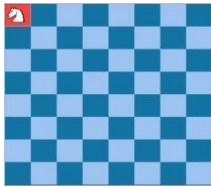


Μοντελοποίηση

$$\pi = (4, 1, 3, 2)$$



Εφαρμογές



Διαδρομές Hamilton

Χάρτες
Χρωματισμός

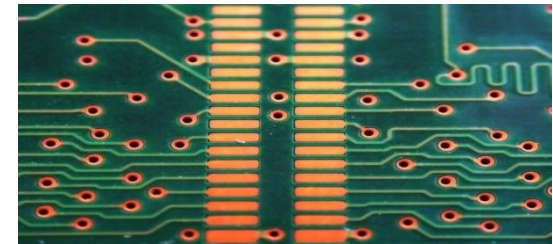


Συνδεσμικότητα



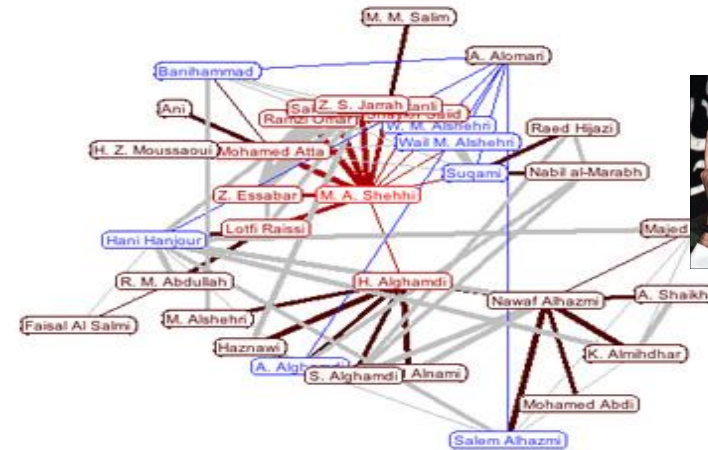
Συντομότερες διαδρομές

GPS – Navigation



Επιπεδικότητα

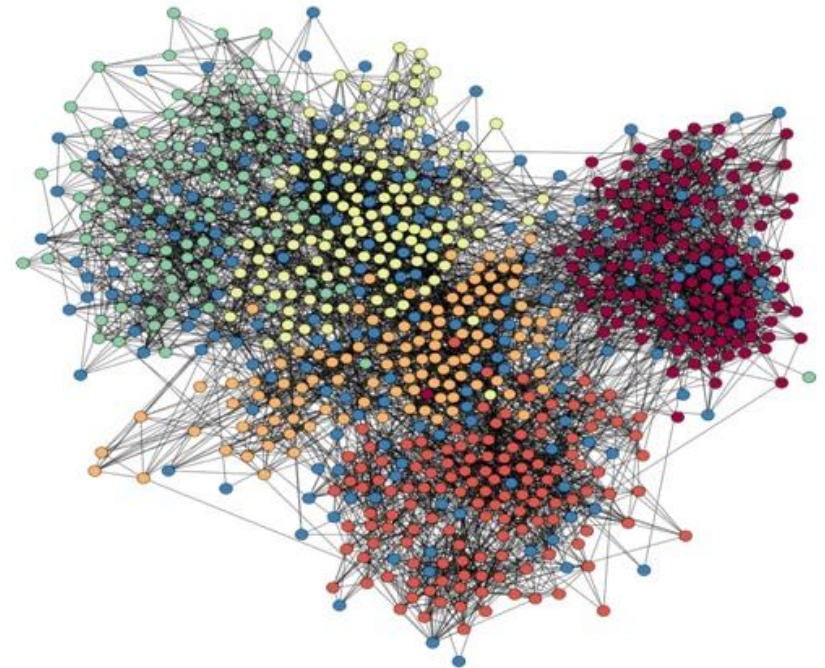
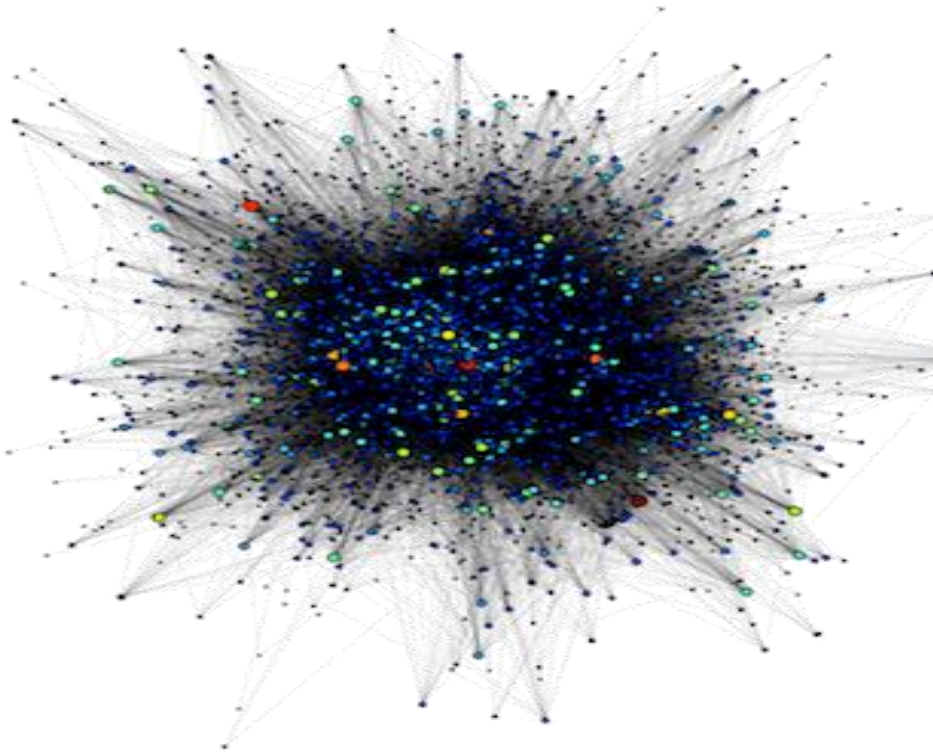
Εφαρμογές



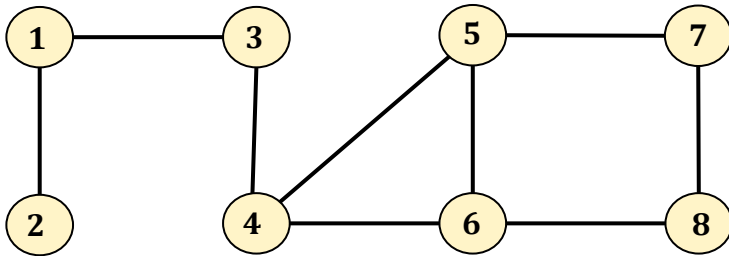
Κοινωνικά δίκτυα
(small-world Phenomena)

Ποιο δίκτυο διαδρομών είναι το ασφαλέστερο;

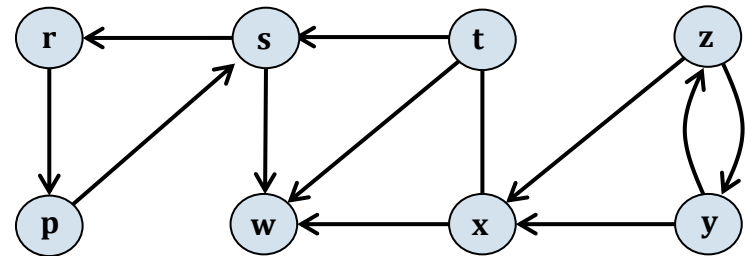
Μεγάλα Γραφήματα... Δίκτυα !!!



Ορισμός



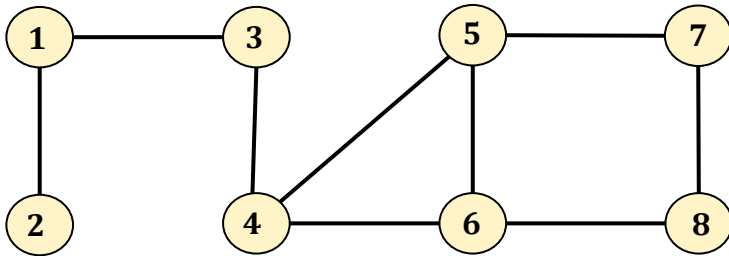
$$G_1 = (V_1, E_1)$$



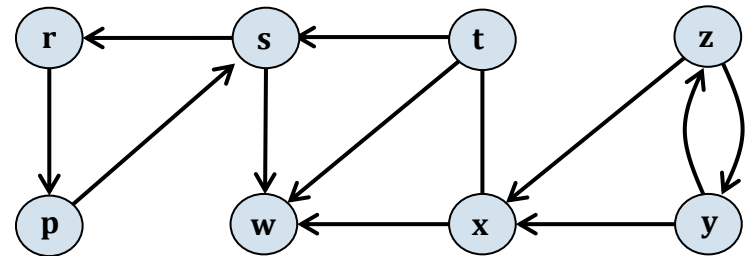
$$G_2 = (V_2, E_2)$$

- ✓ **Γράφημα (ή γράφος) :** ένα ζεύγος (V, E) , V ένα μη κενό σύνολο
 E διμελής σχέση πάνω στο V
- ✓ **Μη-κατευθυνόμενο γράφημα :** σχέση E συμμετρική
- ✓ V : κορυφές (vertices) ή κόμβοι (nodes) και E : ακμές (edges)

Απεικόνιση

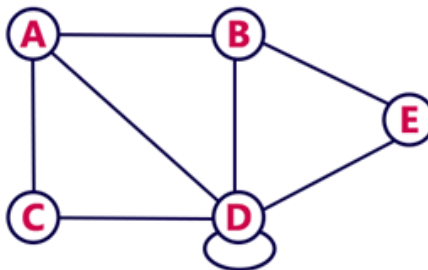


$$G_1 = (V_1, E_1)$$



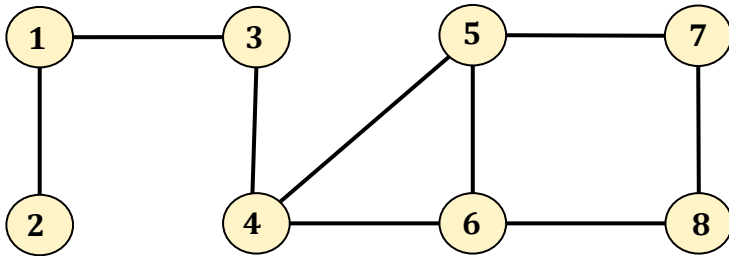
$$G_2 = (V_2, E_2)$$

✓ Πίνακας Γειτνίασης

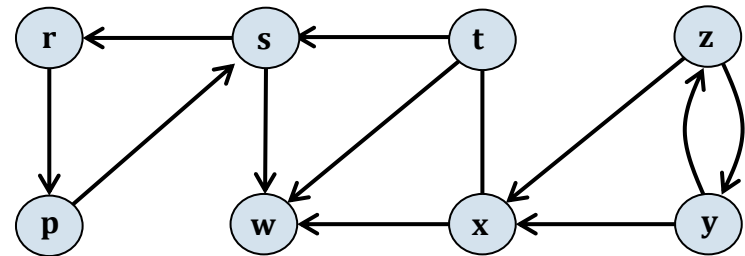


	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	1	1
C	1	0	0	1	0
D	1	1	1	1	1
E	0	1	0	1	0

Απεικόνιση

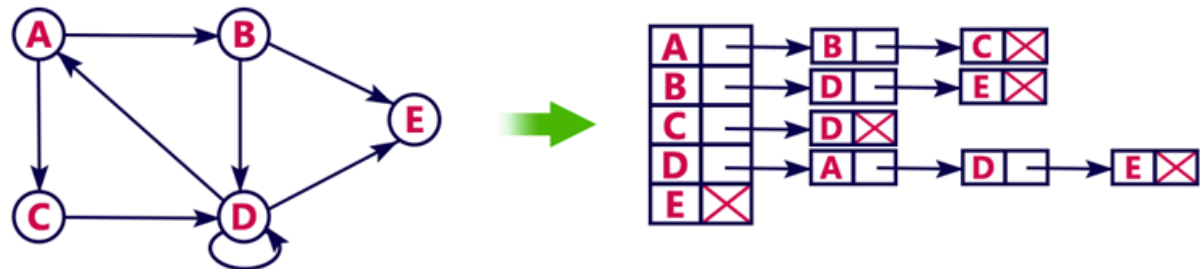


$$G_1 = (V_1, E_1)$$

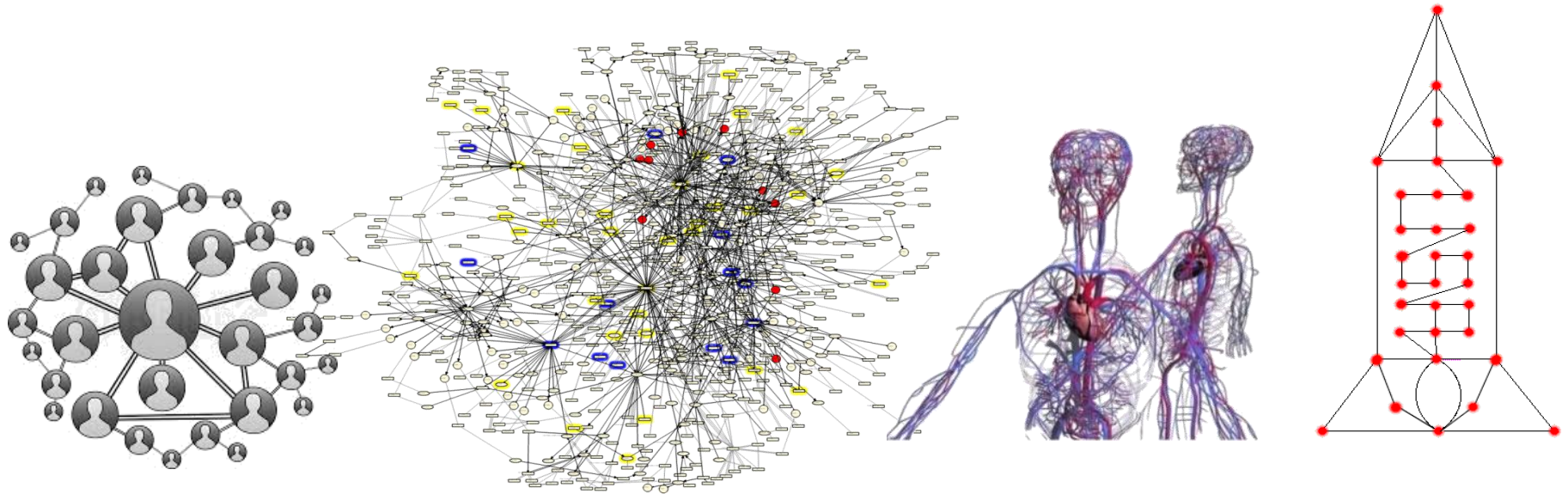
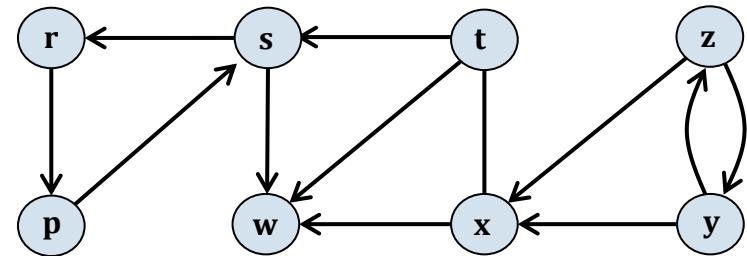
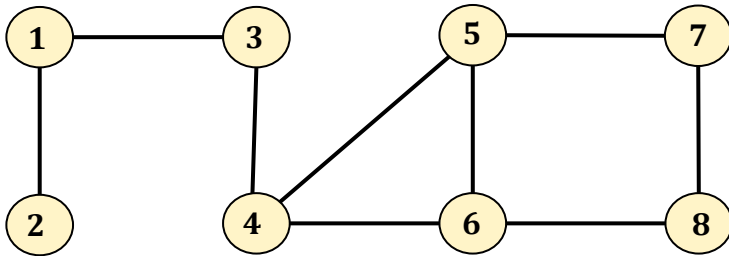


$$G_2 = (V_2, E_2)$$

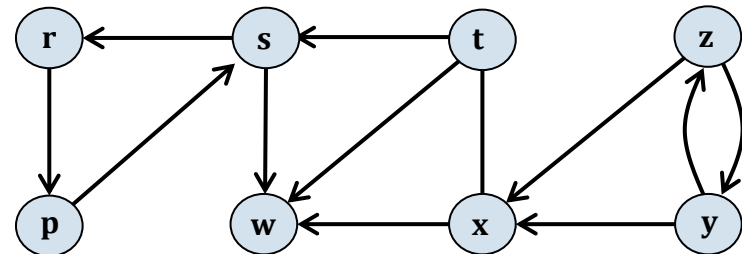
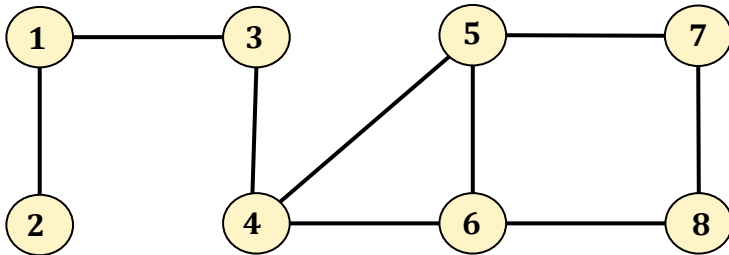
✓ **Λίστα
Γειτνίασης**



Τι θα θέλαμε... Πρωτίστως !!!



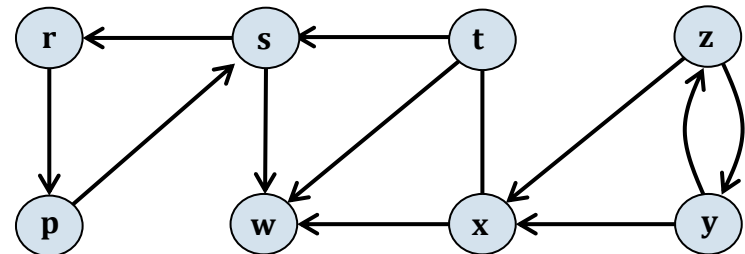
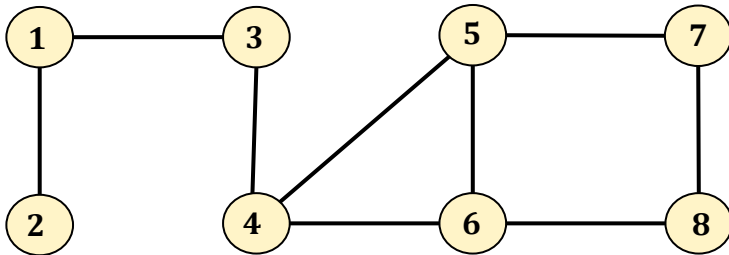
Τι θα θέλαμε... Πρωτίστως !!!



Αλγορίθμους Διερεύνησης ή Διαπέρασης !!!

Πρωτίστως θα θέλαμε μια διαδικασία - αλγόριθμο που θα μας εξασφάλιζε ότι, ξεκινώντας από ένα κόμβο, **θα «επισκεφθούμε» όλους** (θα «περάσουμε» από όλους) τους κόμβους του γραφήματος !!!

Τι θα θέλαμε... Πρωτίστως !!!



Αλγορίθμους Διερεύνησης ή Διαπέρασης !!!

Διερεύνηση Breadth-first Search (BFS)

Διερεύνηση Depth-first Search (DFS)

Διερεύνηση Κατά Πλάτος - Διερεύνηση Κατά Βάθος

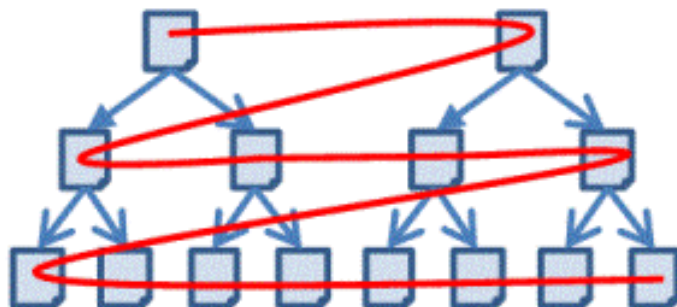
Breadth-first Search (BFS)

Depth-first Search (DFS)

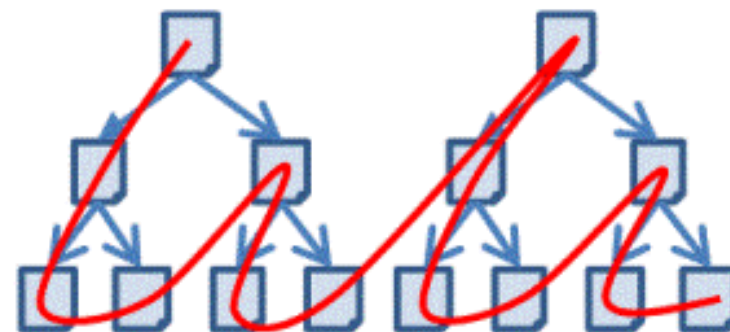


BFS vs DFS

Breadth First Search

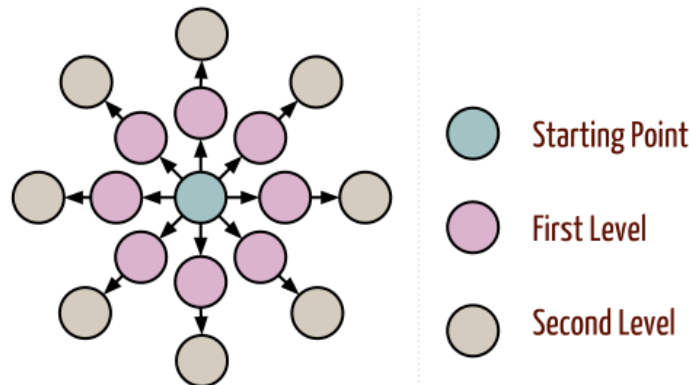


Depth-First Search

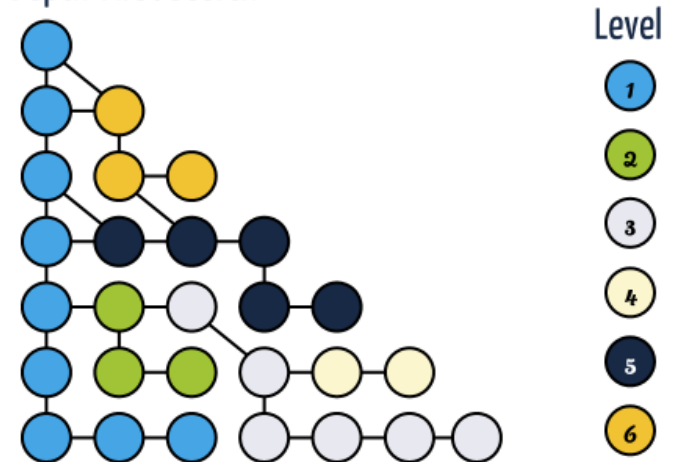


BFS vs DFS

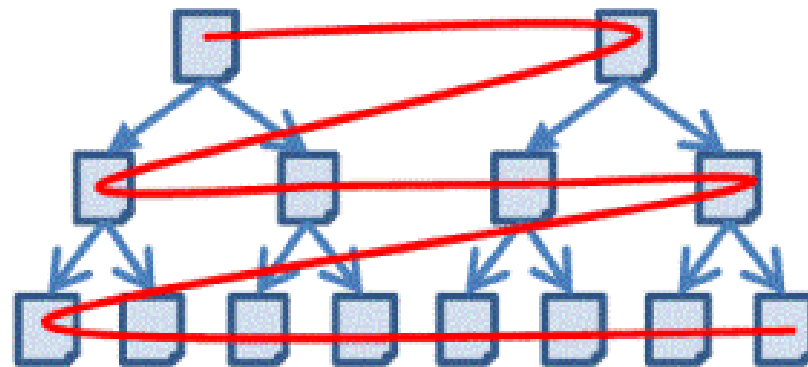
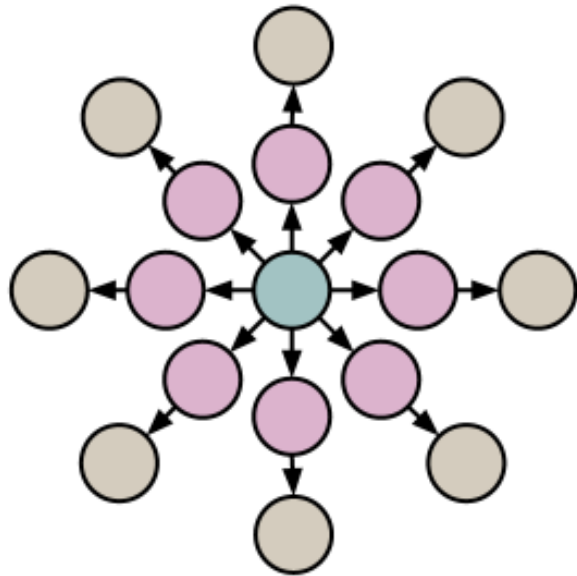
Breadth First Search



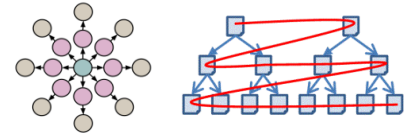
Depth-First Search



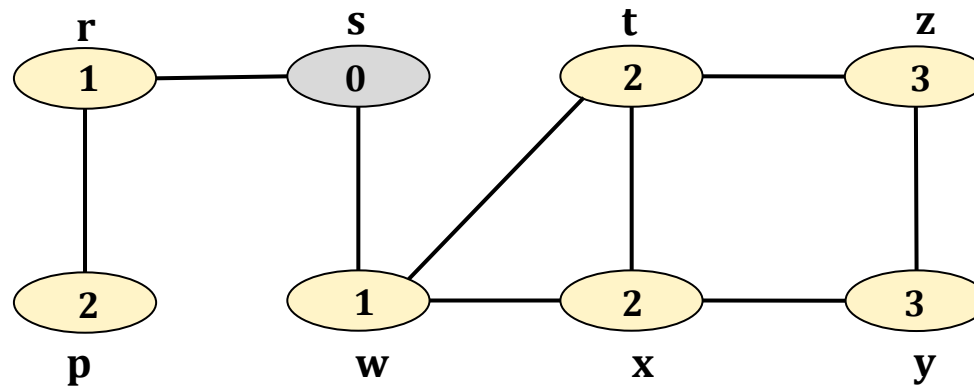
■ Βασική Ιδέα BFS !



■ Βασική Ιδέα BFS !



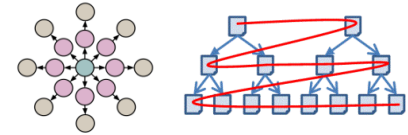
- Έστω $G = (V, E)$ ένα γράφημα και $s \in V$ ένας κόμβος εκκίνησης



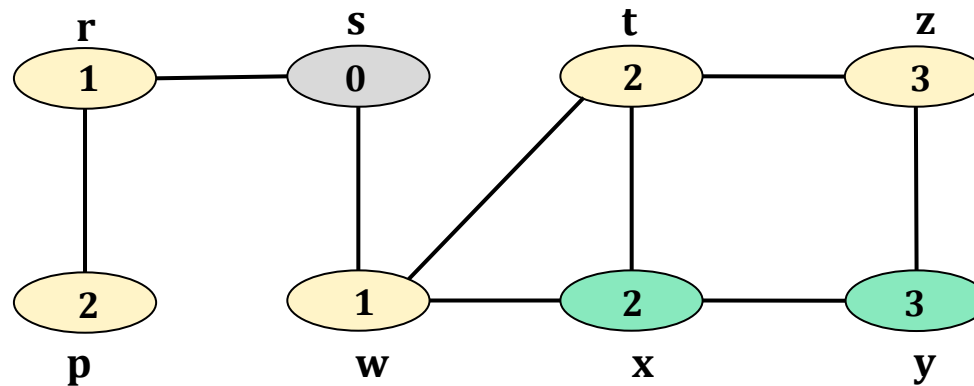
- $d(x)$: ελάχιστη απόσταση του κόμβου $x \in V$ από τον s

Διερεύνηση κατά Πλάτος - BFS

■ Βασική Ιδέα BFS !

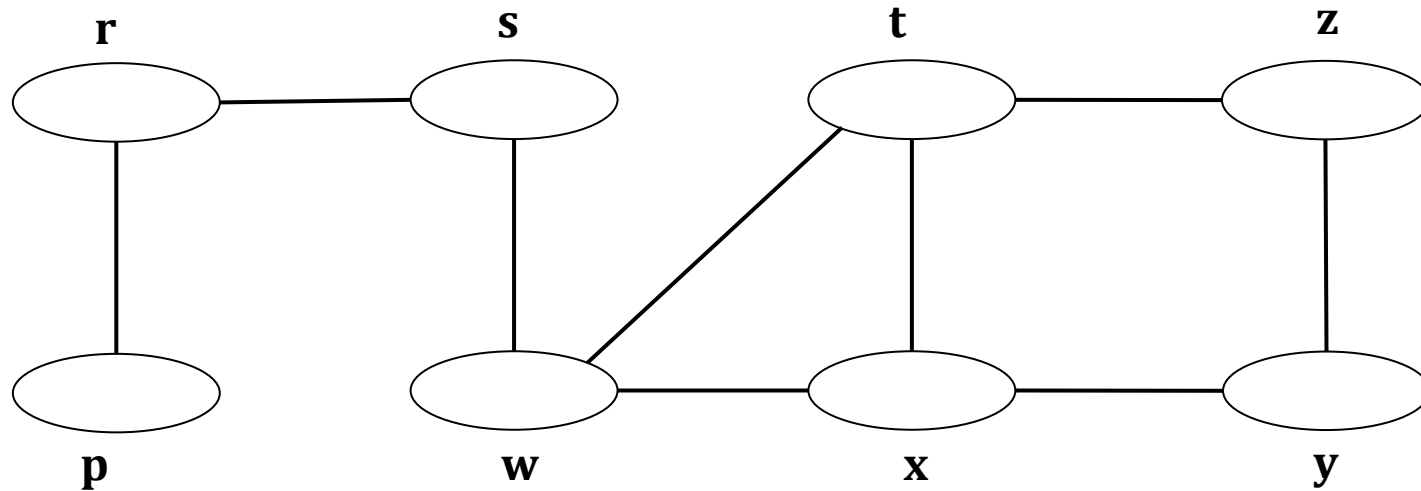


- Ο κόμβος x διερευνάται **πριν** τον κόμβο y σε μια διερεύνηση BFS, εάν $d(x) < d(y)$ από τον κόμβο εκκίνησης $s \in V$ του G

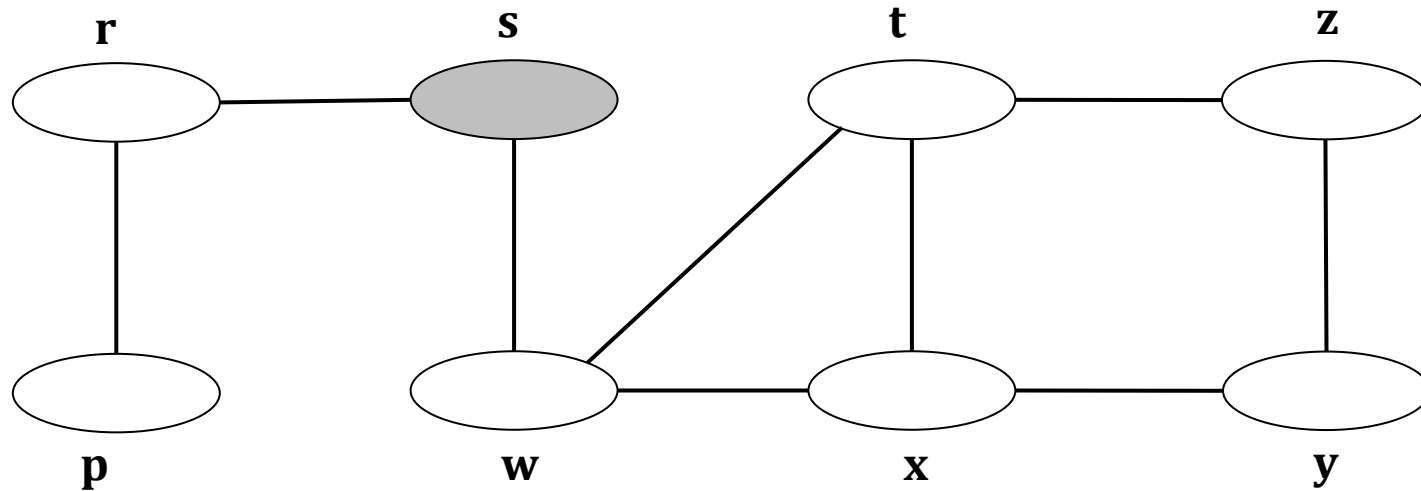


- Ο αλγόριθμος **BFS** διερευνά κάθε κόμβο x του G : $s \rightarrow x$

□ Παράδειγμα

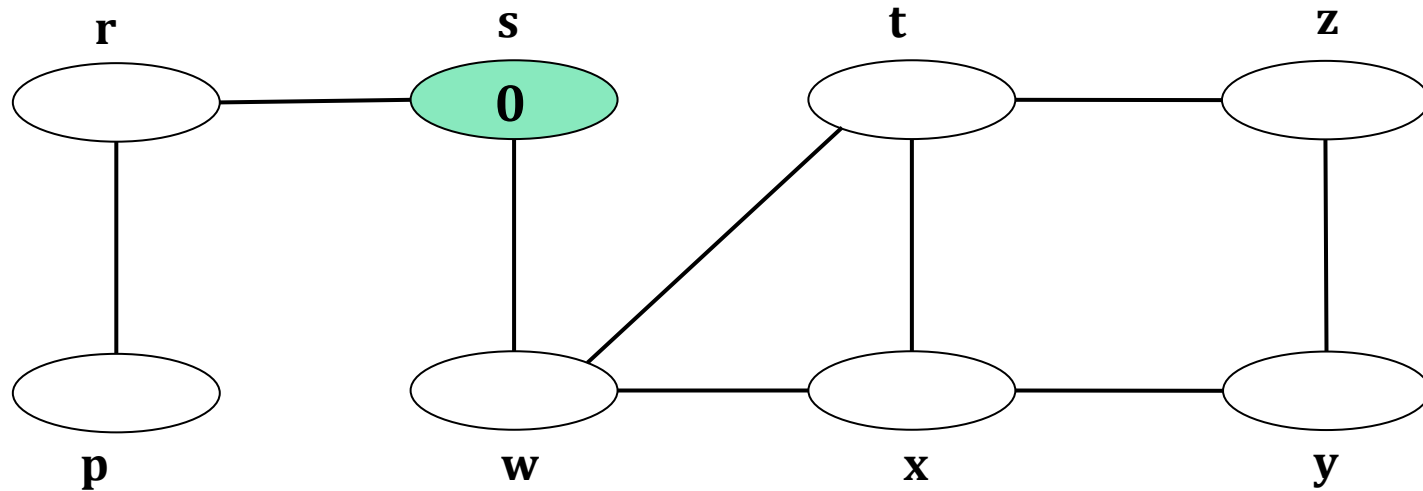


□ Παράδειγμα



Διερεύνηση κατά Πλάτος - BFS

□ Παράδειγμα



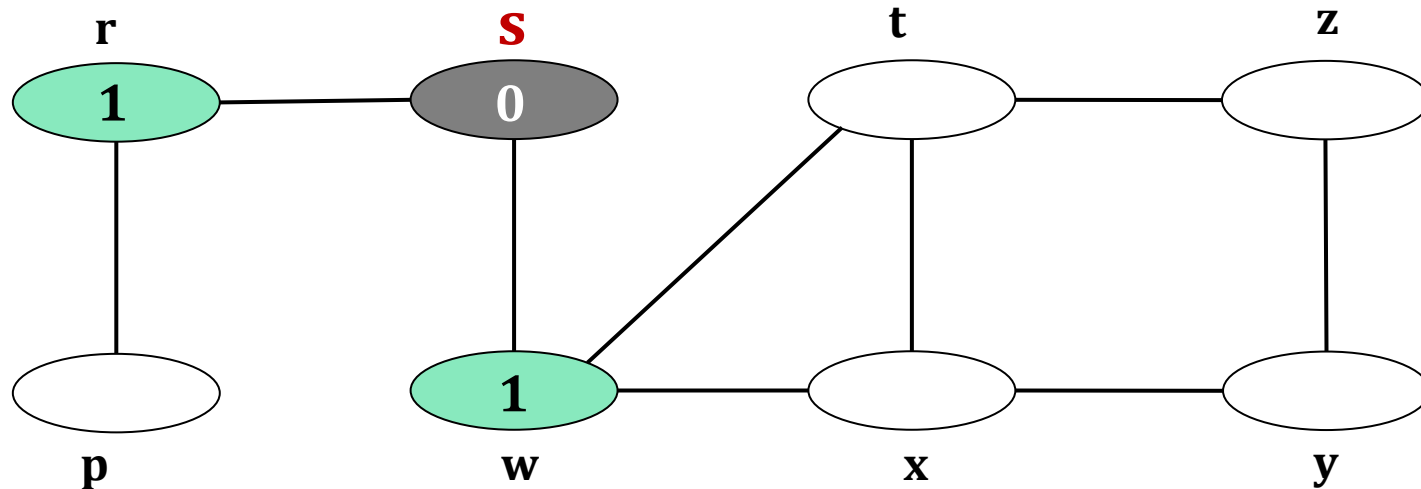
$\pi(s) = \text{nil}$

Q :

s

Διερεύνηση κατά Πλάτος - BFS

□ Παράδειγμα



$\pi(s) = \text{nil}$

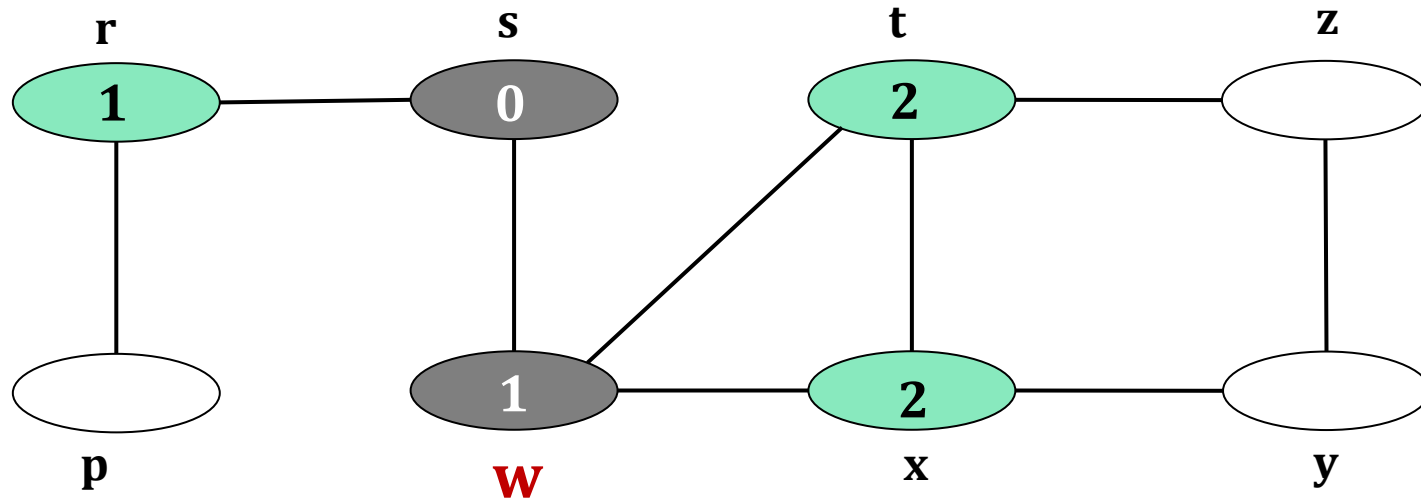
$\pi(w) = s$ $\pi(r) = s$

s $\leftarrow \text{head}(Q) :$

s w r

Διερεύνηση κατά Πλάτος - BFS

□ Παράδειγμα



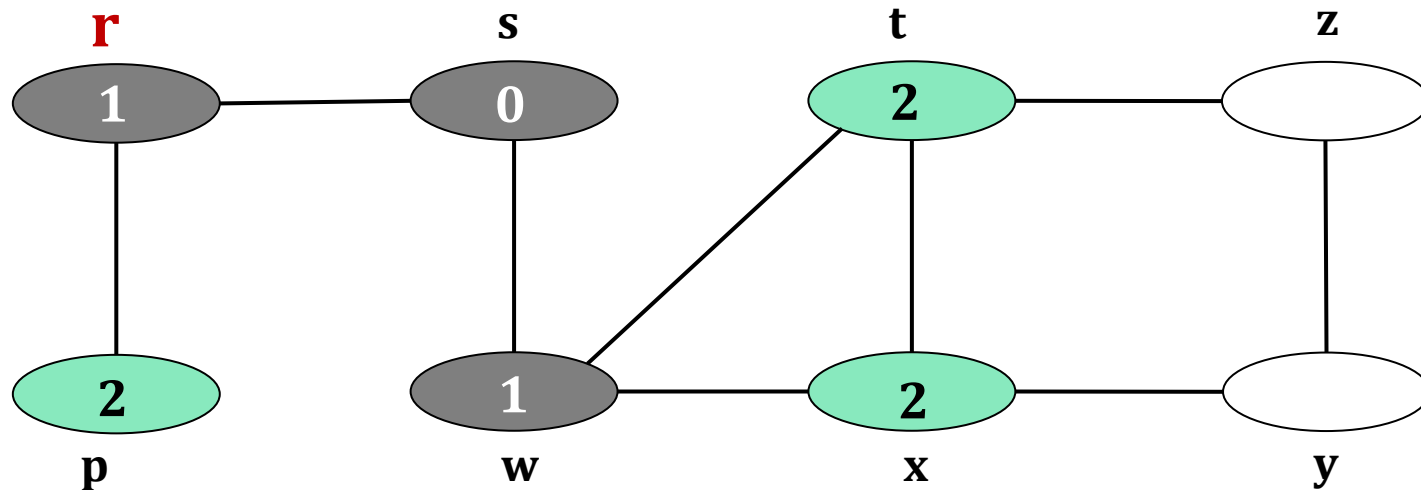
$\pi(s) = \text{nil}$

W $\leftarrow \text{head}(Q)$:

s w r **t** **x**

$\pi(w) = s$ $\pi(r) = s$ $\pi(\text{t}) = \text{w}$ $\pi(\text{x}) = \text{w}$

□ Παράδειγμα



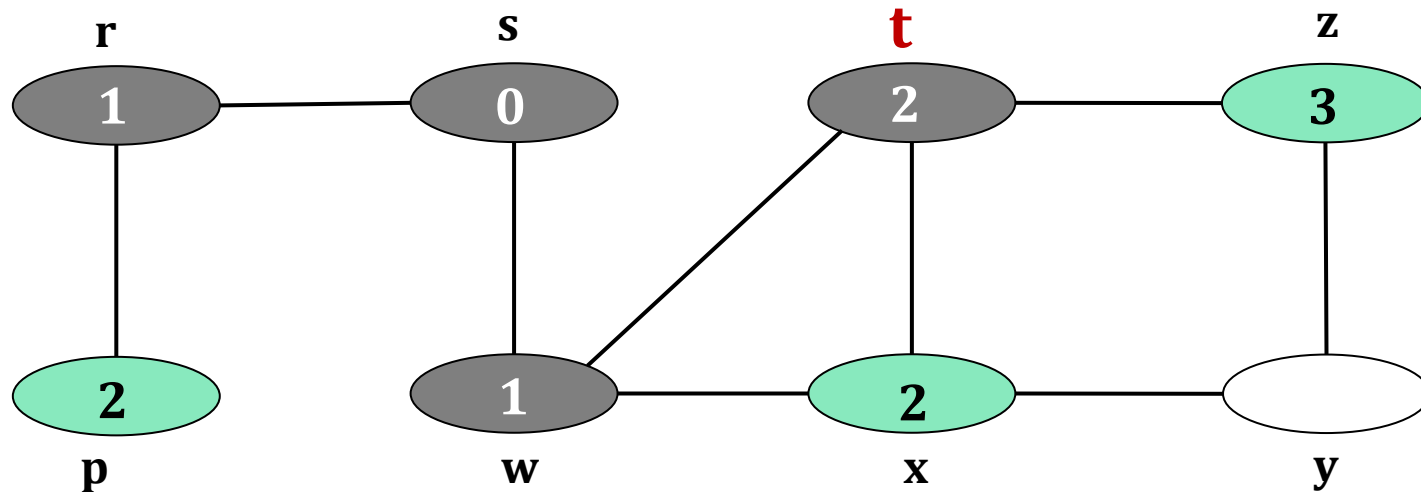
$\pi(s) = \text{nil}$

r $\leftarrow \text{head}(Q)$:

s w **r** t x **p**

$\pi(w) = s$ $\pi(r) = s$ $\pi(t) = w$ $\pi(x) = w$ $\pi(p) = \text{r}$

□ Παράδειγμα



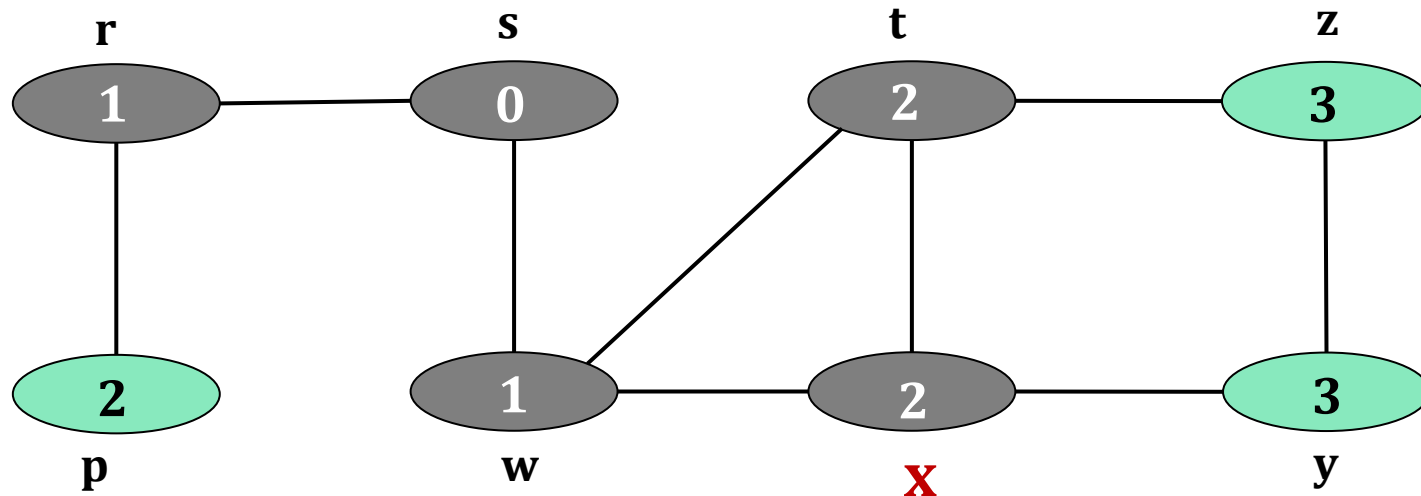
$\pi(s) = \text{nil}$

t $\leftarrow \text{head}(Q)$:

s w r **t** x p **z**

$\pi(w) = s$ $\pi(r) = s$ $\pi(t) = w$ $\pi(x) = w$ $\pi(p) = r$ $\pi(\text{z}) = \text{t}$

□ Παράδειγμα



$\pi(s)=\text{nil}$

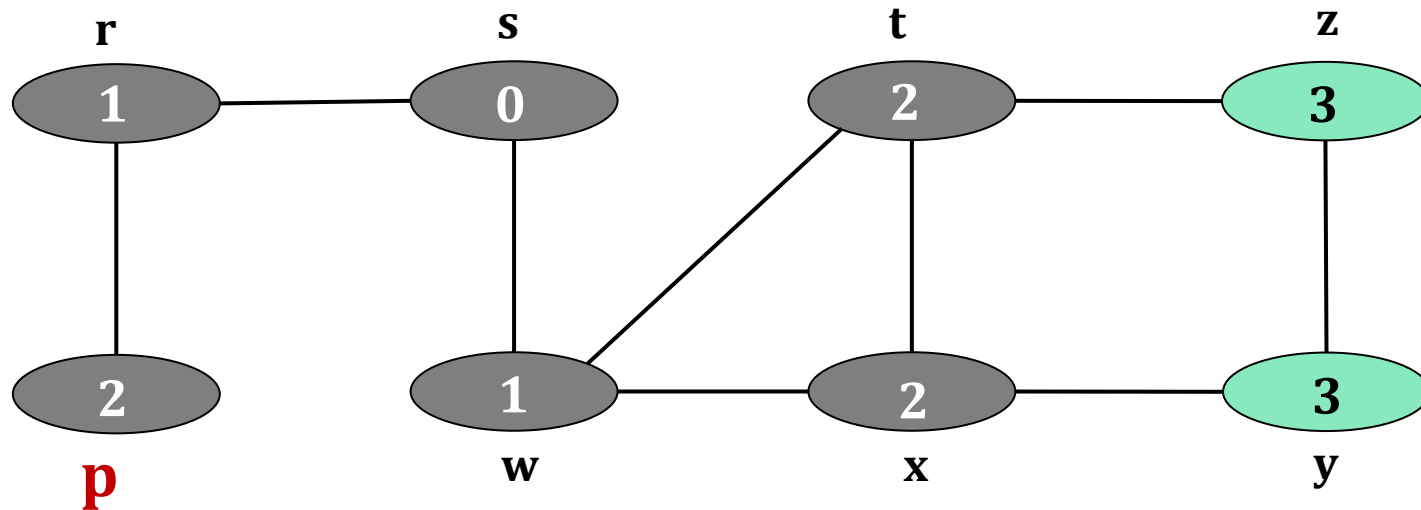
X $\leftarrow \text{head}(Q)$:

s	w	r	t	x	p	z	y
---	---	---	---	---	---	---	---

$\pi(w)=s$ $\pi(r)=s$ $\pi(t)=w$ $\pi(x)=w$ $\pi(p)=r$ $\pi(\text{z})=\text{t}$ $\pi(\text{y})=\text{x}$

Διερεύνηση κατά Πλάτος - BFS

□ Παράδειγμα



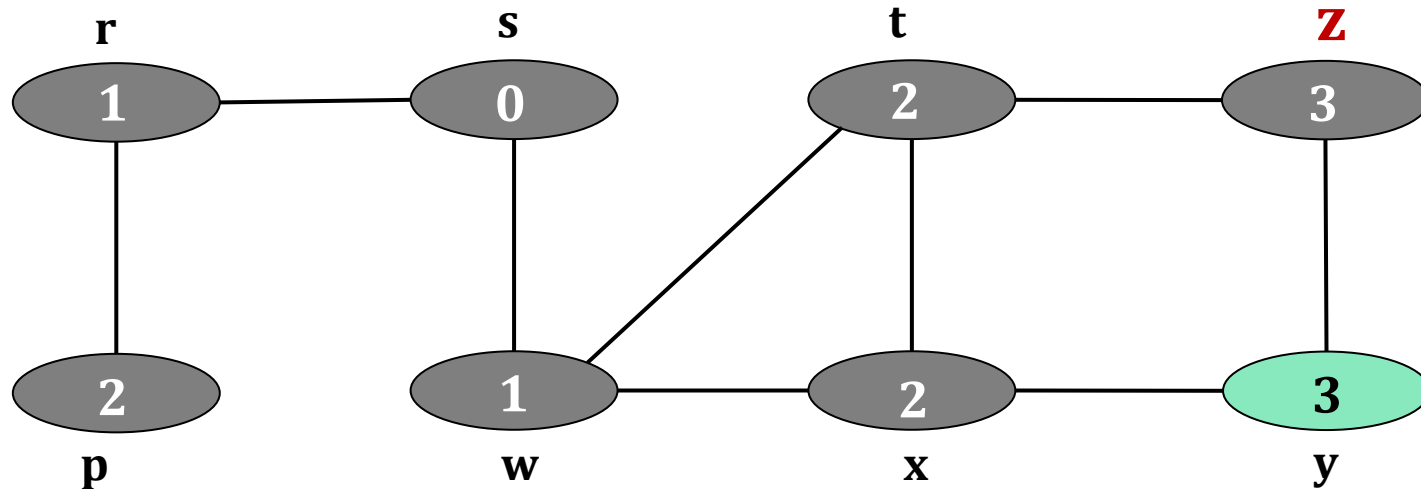
$\pi(s) = \text{nil}$

p $\leftarrow \text{head}(Q)$:

s	w	r	t	x	p	z	y
---	---	---	---	---	---	---	---

$\pi(w) = s$ $\pi(r) = s$ $\pi(t) = w$ $\pi(x) = w$ $\pi(p) = r$ $\pi(z) = t$ $\pi(y) = x$

□ Παράδειγμα



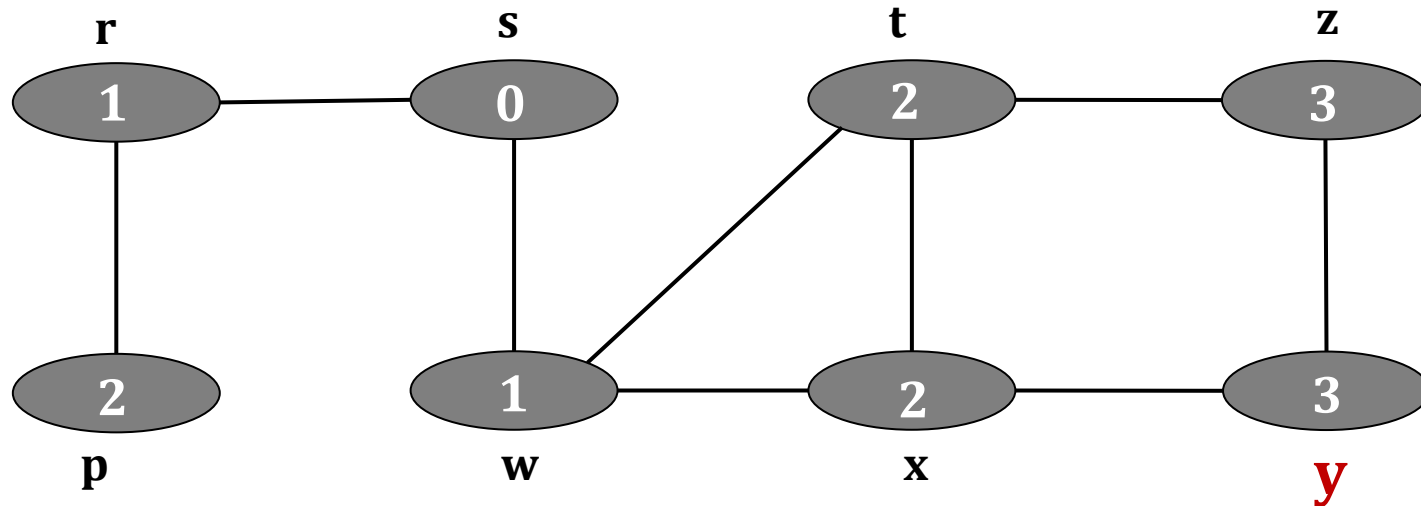
$\pi(s) = \text{nil}$

z $\leftarrow \text{head}(Q)$:

s w r t x p z y

$\pi(w) = s$ $\pi(r) = s$ $\pi(t) = w$ $\pi(x) = w$ $\pi(p) = r$ $\pi(z) = t$ $\pi(y) = x$

□ Παράδειγμα



$\pi(s) = \text{nil}$

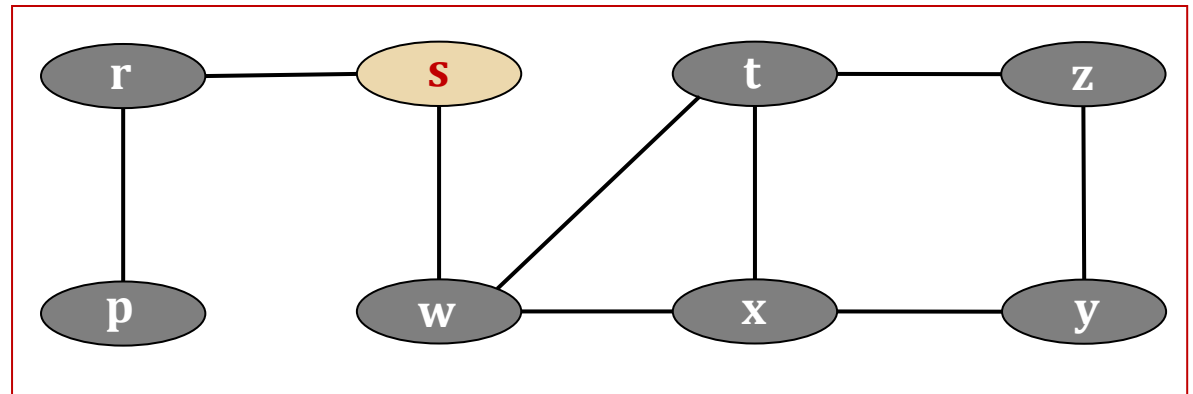
y $\leftarrow \text{head}(Q)$:

s w r t x p z y

$\pi(w) = s$ $\pi(r) = s$ $\pi(t) = w$ $\pi(x) = w$ $\pi(p) = r$ $\pi(z) = t$ $\pi(y) = x$

Διερεύνηση κατά Πλάτος - BFS

□ Παράδειγμα



BFS = (s, w, r, t, x, p, z, y)

$\pi(s) = \text{nil}$

$\pi(w) = s \quad \pi(r) = s \quad \pi(t) = w \quad \pi(x) = w \quad \pi(p) = r \quad \pi(z) = t \quad \pi(y) = x$

● Πολυπλοκότητα BFS

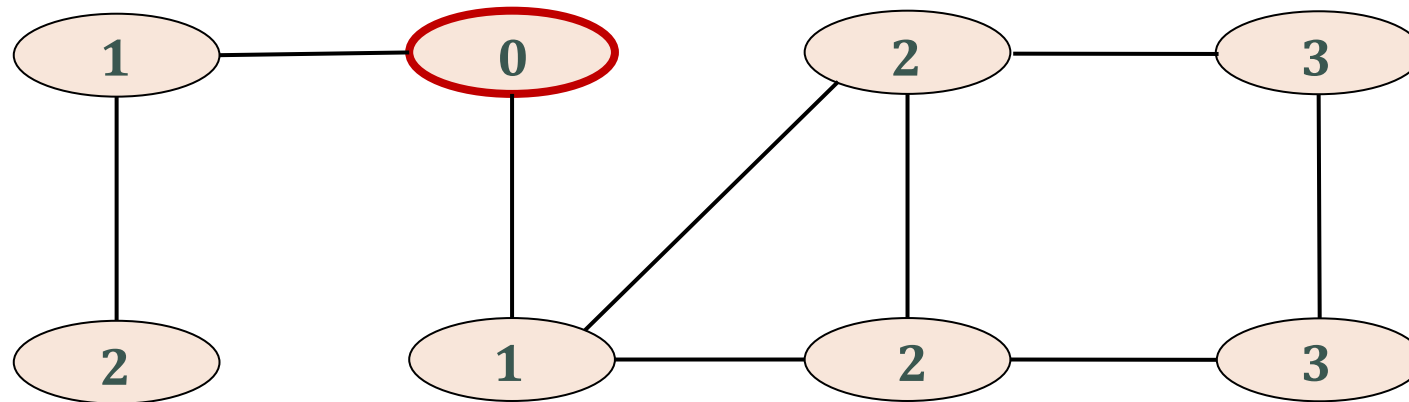
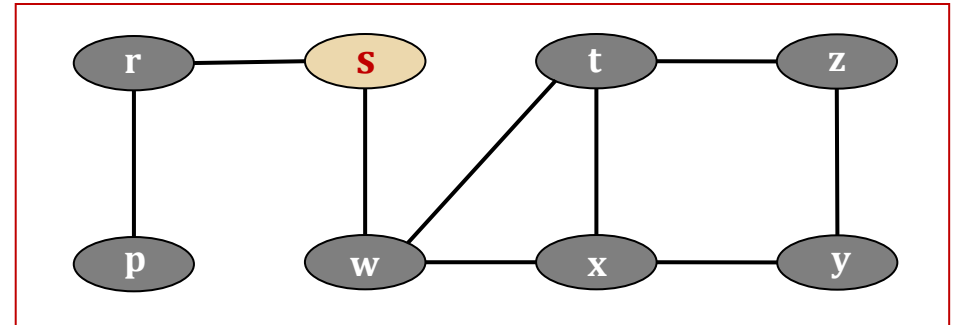
- ✓ Κάθε κόμβος $v \in V$ εισέρχεται στην ουρά Q το πολύ 1 φορά και διαγράφεται από αυτή το πολύ 1.
- ✓ Κάθε Εισαγωγή / Διαγραφή απαιτεί $O(1)$ χρόνο.
- ✓ Συνολικός χρόνος για όλες τις λειτουργίες της Q : $O(|V|) = O(n)$
- ✓ Η λίστα γειτνίασης $L(v)$ κάθε κόμβου $v \in V$ διερευνάτε 1 μόνο φορά όταν ο κόμβος v διαγράφεται από την Q .

Ισχύει: $\sum_{v \in V} |L(v)| = \Theta(|E|) \Rightarrow O(|E|) = O(m)$ χρόνος για διερεύνηση όλων των λιστών.

- ✓ Άρα, πολυπλοκότητα BFS : $O(|V| + |E|) = O(n + m)$

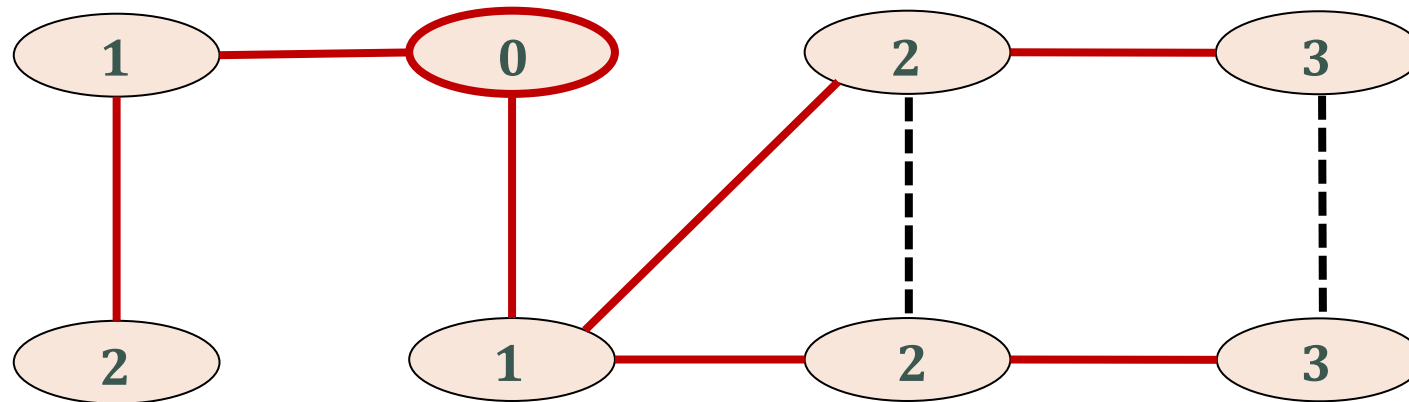
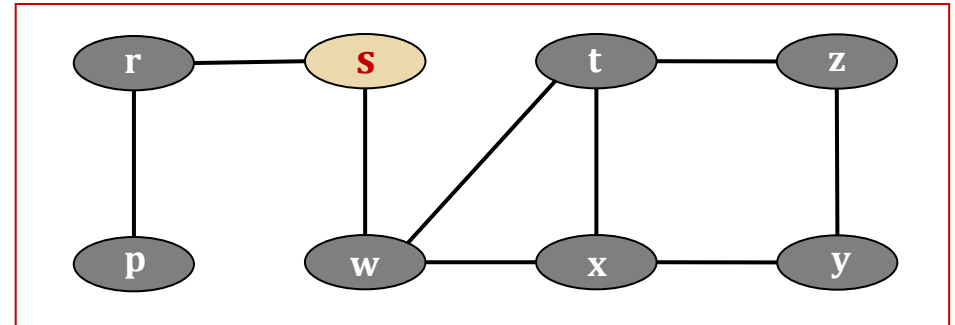
Διερεύνηση κατά Πλάτος - BFS

□ BFS-δένδρο (δάσος)



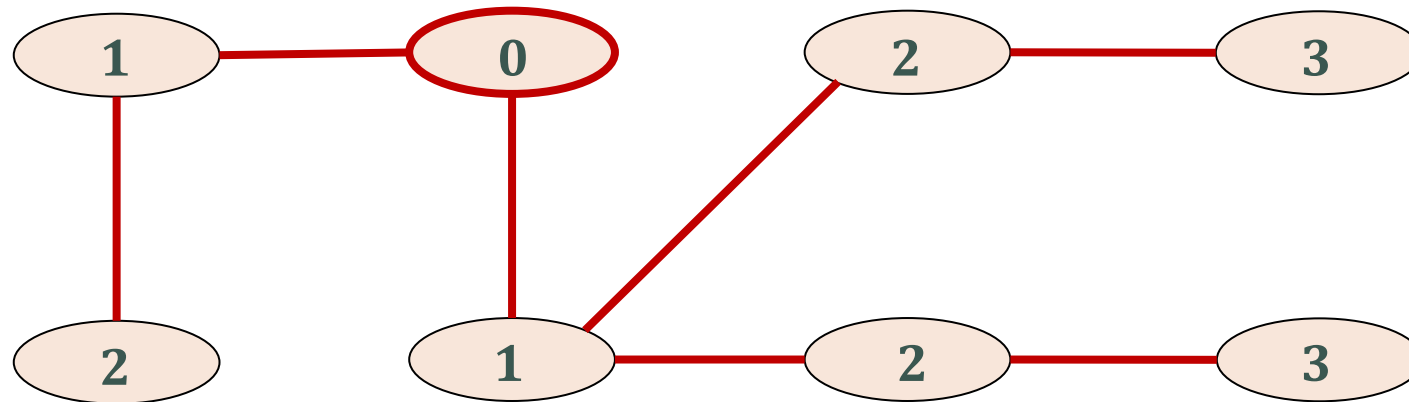
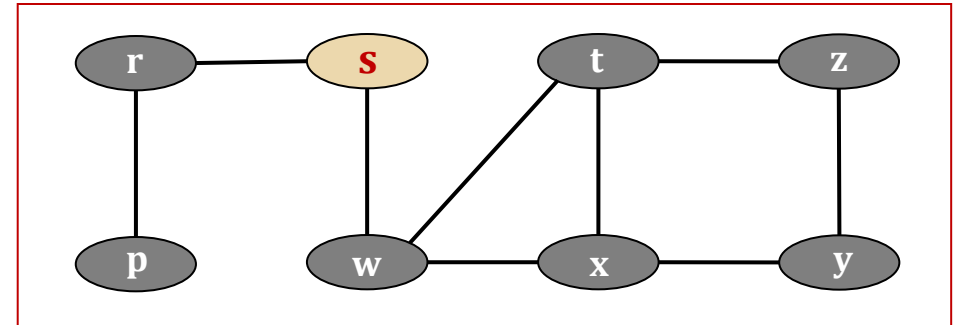
Διερεύνηση κατά Πλάτος - BFS

□ BFS-δένδρο (δάσος)



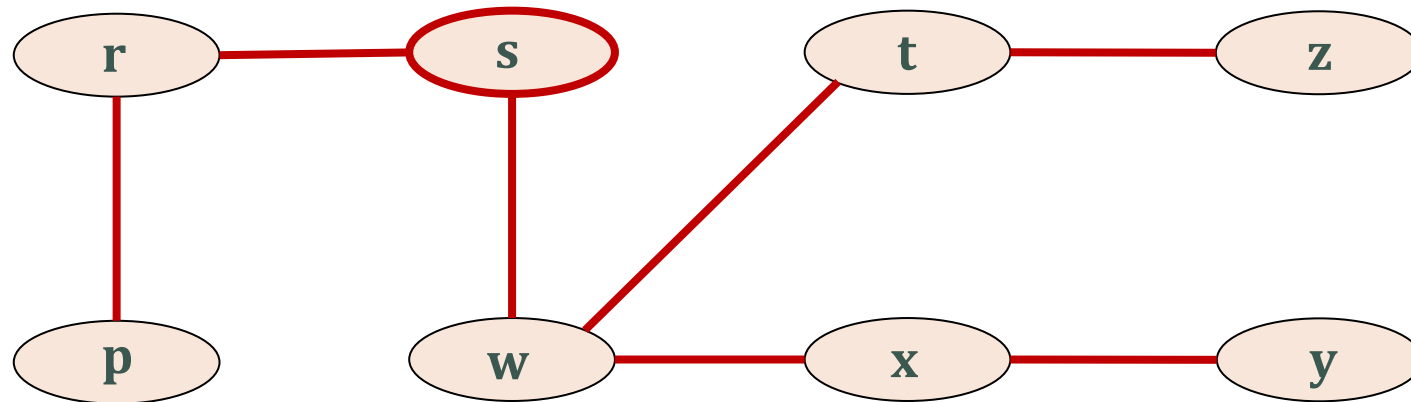
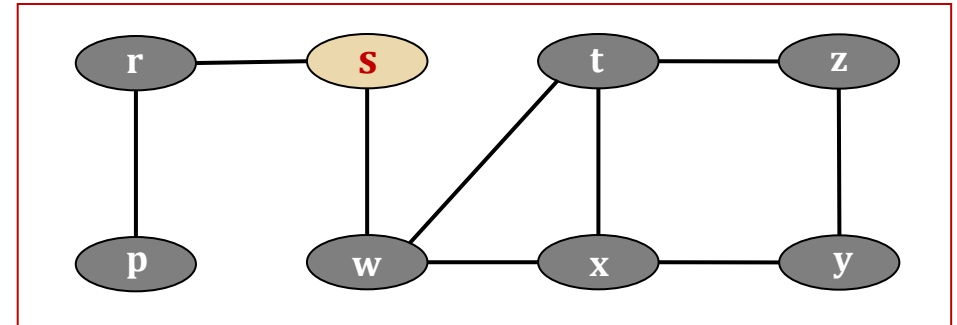
Διερεύνηση κατά Πλάτος - BFS

□ BFS-δένδρο (δάσος)

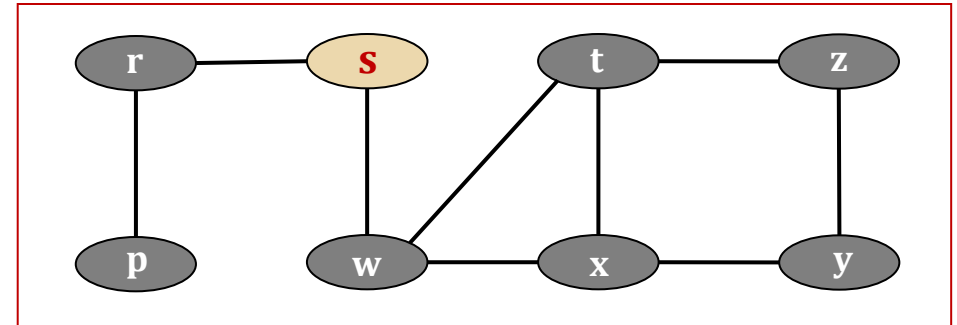
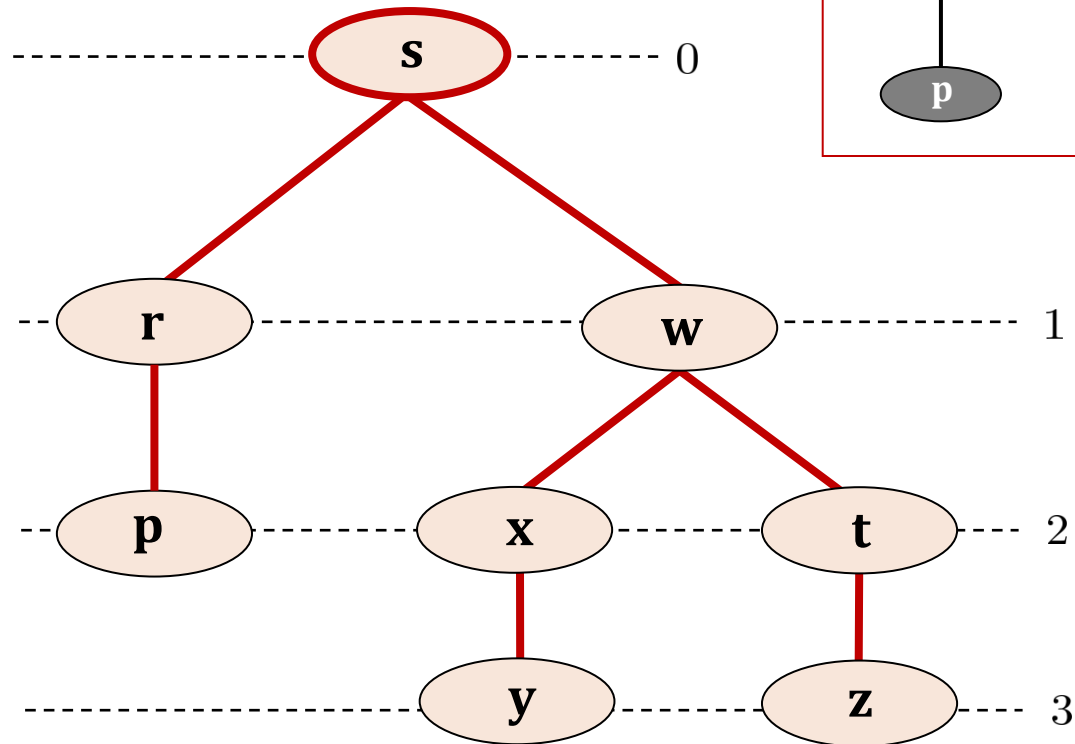


Διερεύνηση κατά Πλάτος - BFS

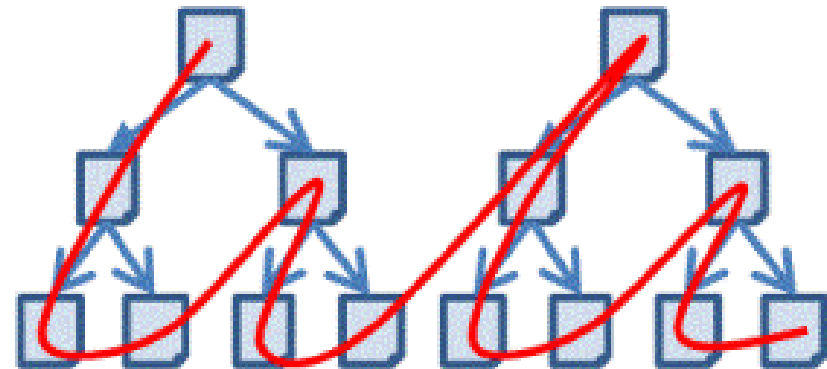
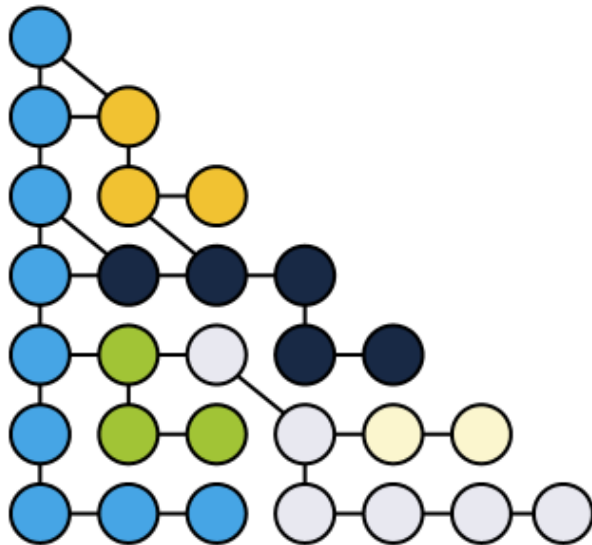
□ BFS-δένδρο (δάσος)



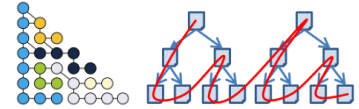
□ BFS-δένδρο (δάσος)



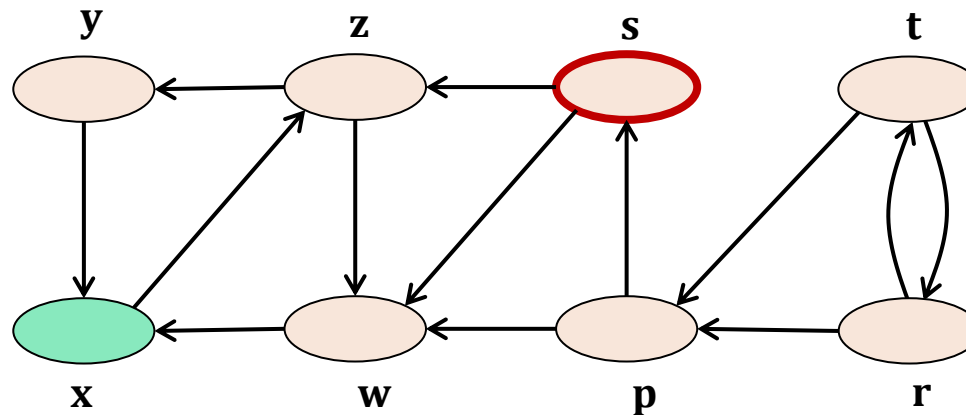
 Βασική Ιδέα DFS !



■ Βασική Ιδέα DFS !

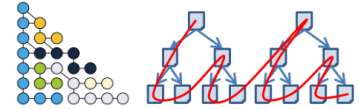


- Η DFS **διερευνά** πρώτα κόμβους που βρίσκονται σε όσο δυνατόν **μεγαλύτερη απόσταση** από τον κόμβο εκκίνησης $s \in V$

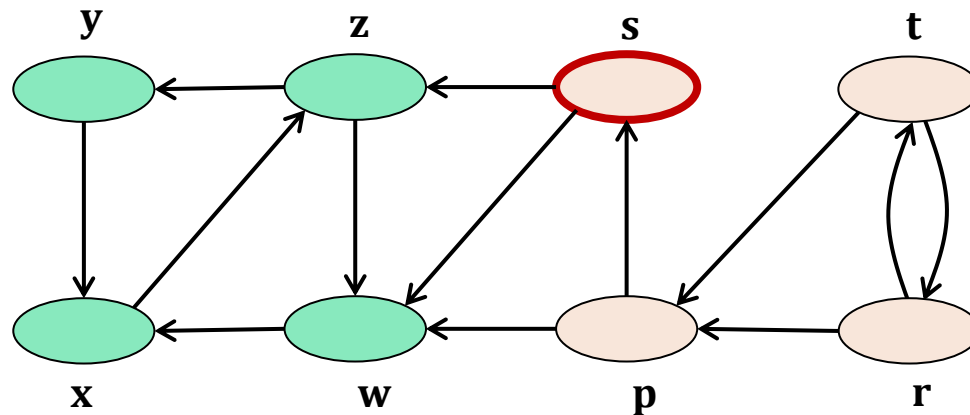


- Εάν όλοι οι γείτονες του κόμβου $v \in V$ έχουν διερευνηθεί, τότε ο DFS «επιστρέφει» στον $u \in V : (u, v) \in E$, και συνεχίζει σε αδιερεύνητους γείτονες του κόμβου u !

■ Βασική Ιδέα DFS !

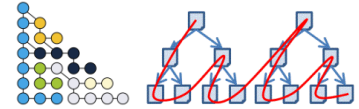


- Ο αλγόριθμος **DFS** σταματά όταν διερευνηθούν **όλοι οι κόμβοι x** του γραφήματος $G : s \rightarrow x$

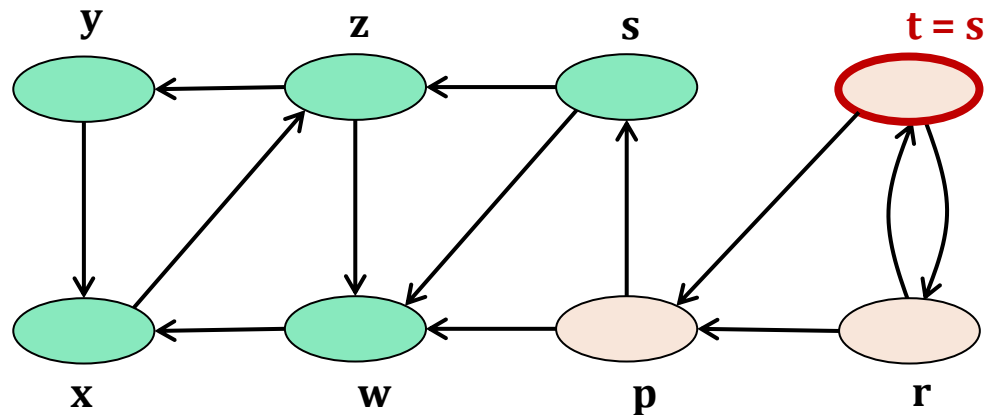


- Εάν $\exists$ κόμβοι που **δεν έχουν διερευνηθεί**, τότε επιλέγει ένα νέο μη-διερευνημένο $s' \in V$ και τον θέτει κόμβο εκκίνησης $s \leftarrow s'$

■ Βασική Ιδέα DFS !

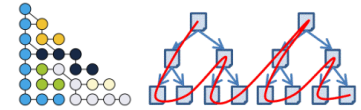


- Ο αλγόριθμος **DFS** σταματά όταν διερευνηθούν **όλοι οι κόμβοι x** του γραφήματος $G : s \rightarrow x$

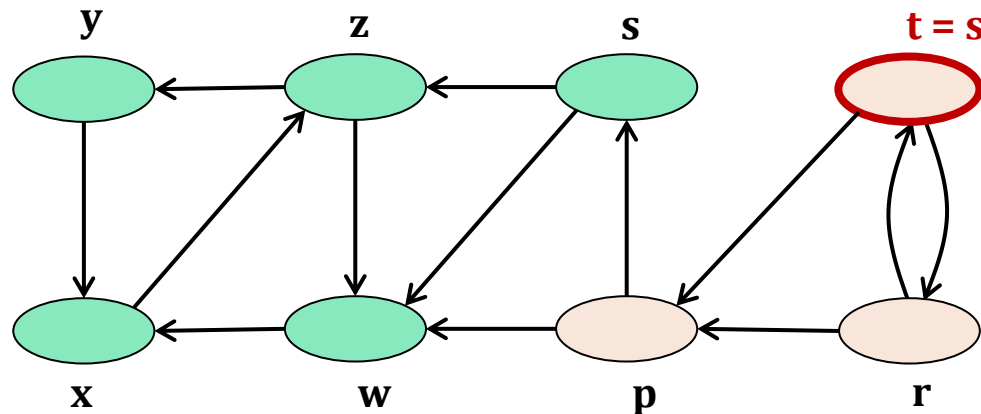


- Εάν $\exists$ κόμβοι που **δεν έχουν διερευνηθεί**, τότε επιλέγει ένα νέο μη-διερευνημένο $s' \in V$ και τον θέτει κόμβο εκκίνησης $s \leftarrow s'$

■ Βασική Ιδέα DFS !



- Ο αλγόριθμος **DFS** σταματά όταν διερευνηθούν **όλοι οι κόμβοι** **x** του γραφήματος $G : s \rightarrow x$

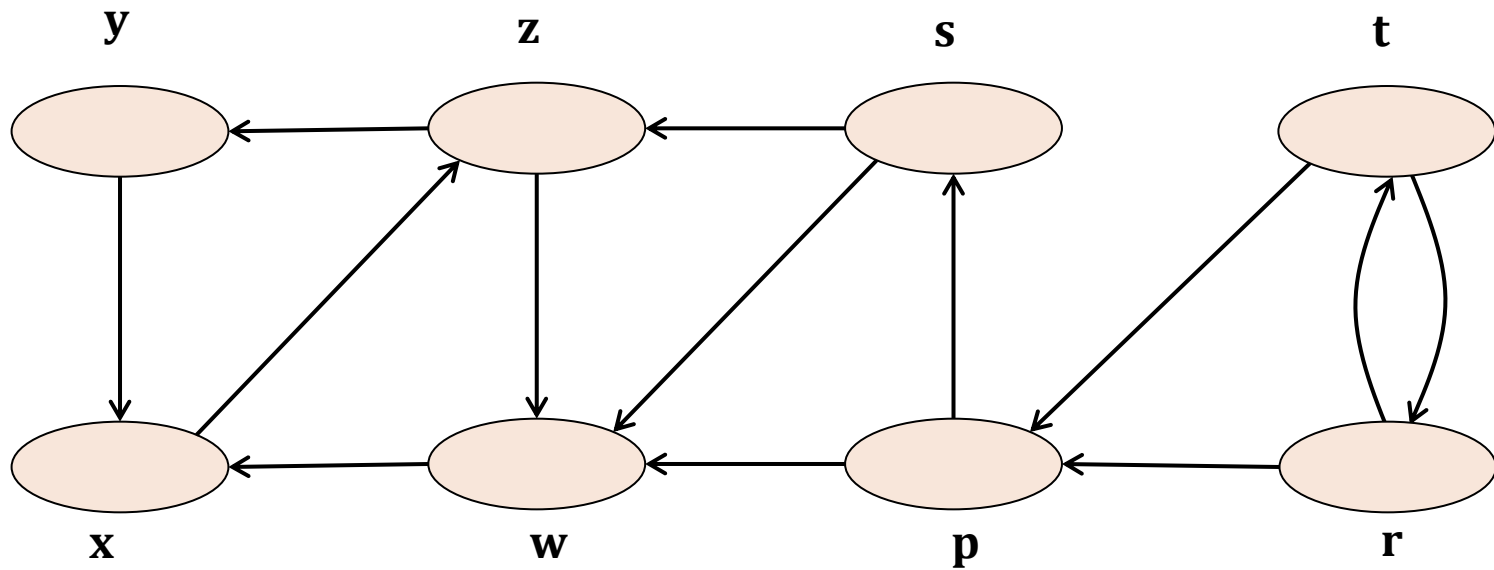


$a(x)/t(x)$

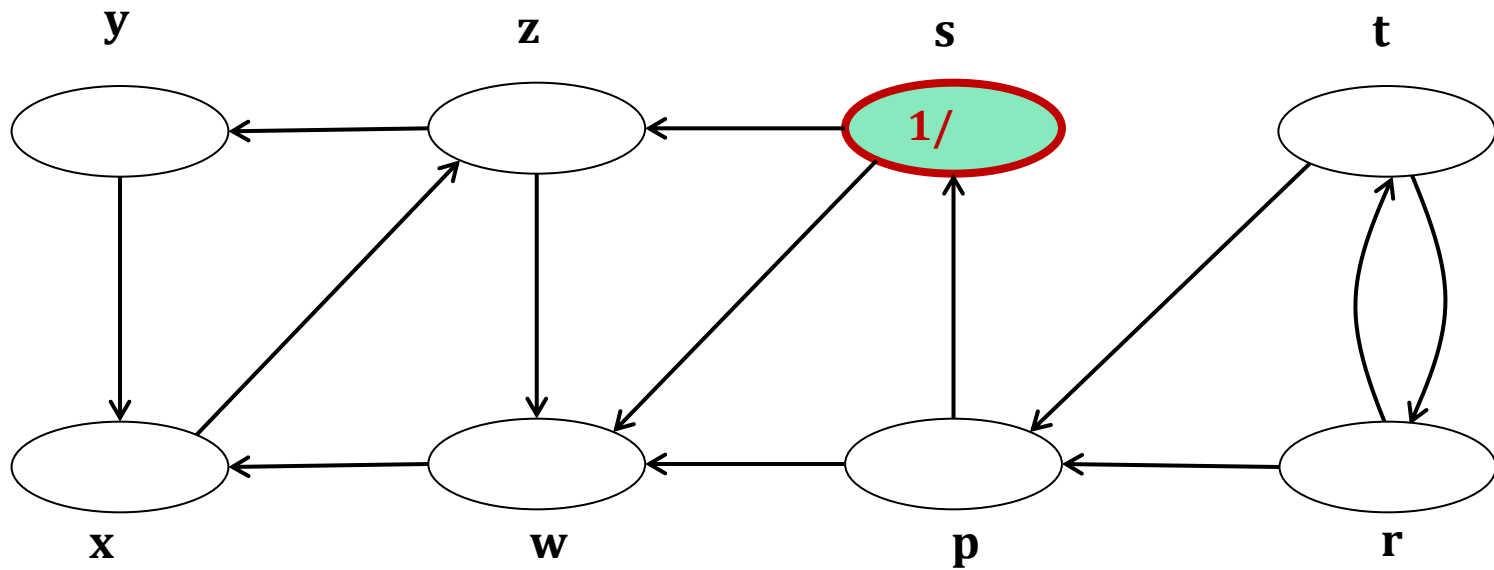
$a(x)$ χρόνος **αρχικής**
επίσκεψης, και
 $t(x)$ χρόνος **τελικής**
επίσκεψης του x

- Εάν $\exists$ κόμβοι που **δεν έχουν διερευνηθεί**, τότε επιλέγει ένα νέο μη-διερευνημένο $s' \in V$ και τον θέτει κόμβο εκκίνησης $s \leftarrow s'$

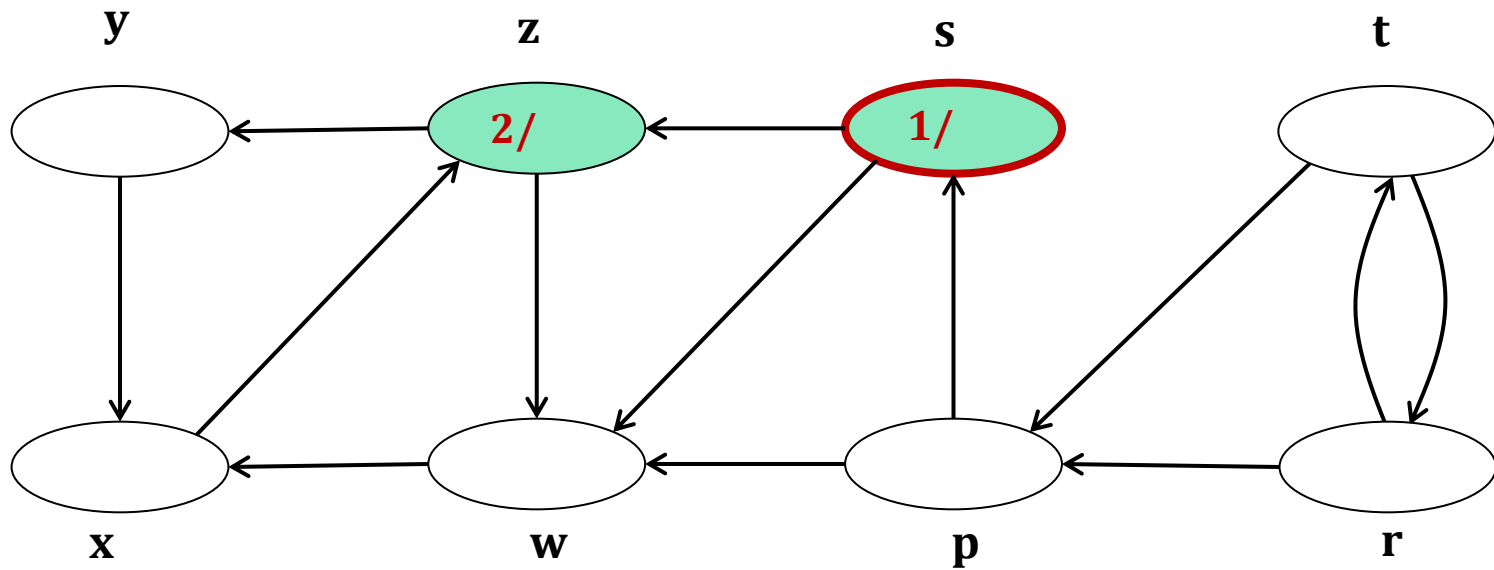
□ Παράδειγμα



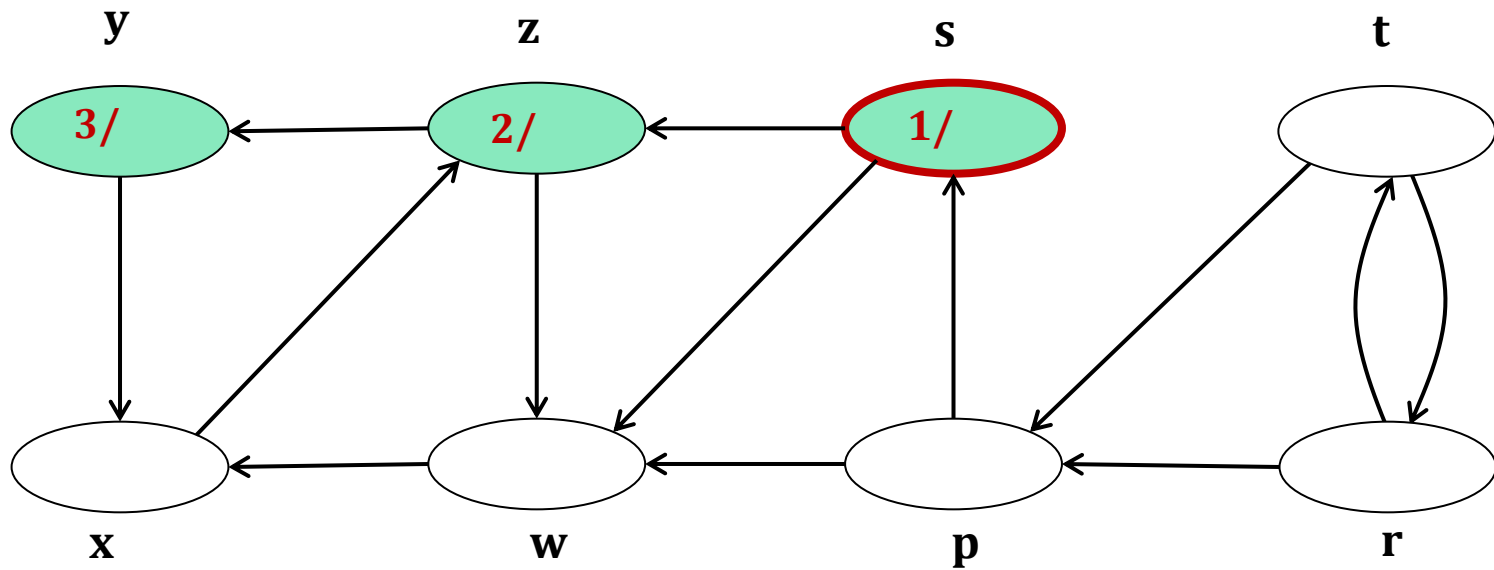
□ Παράδειγμα



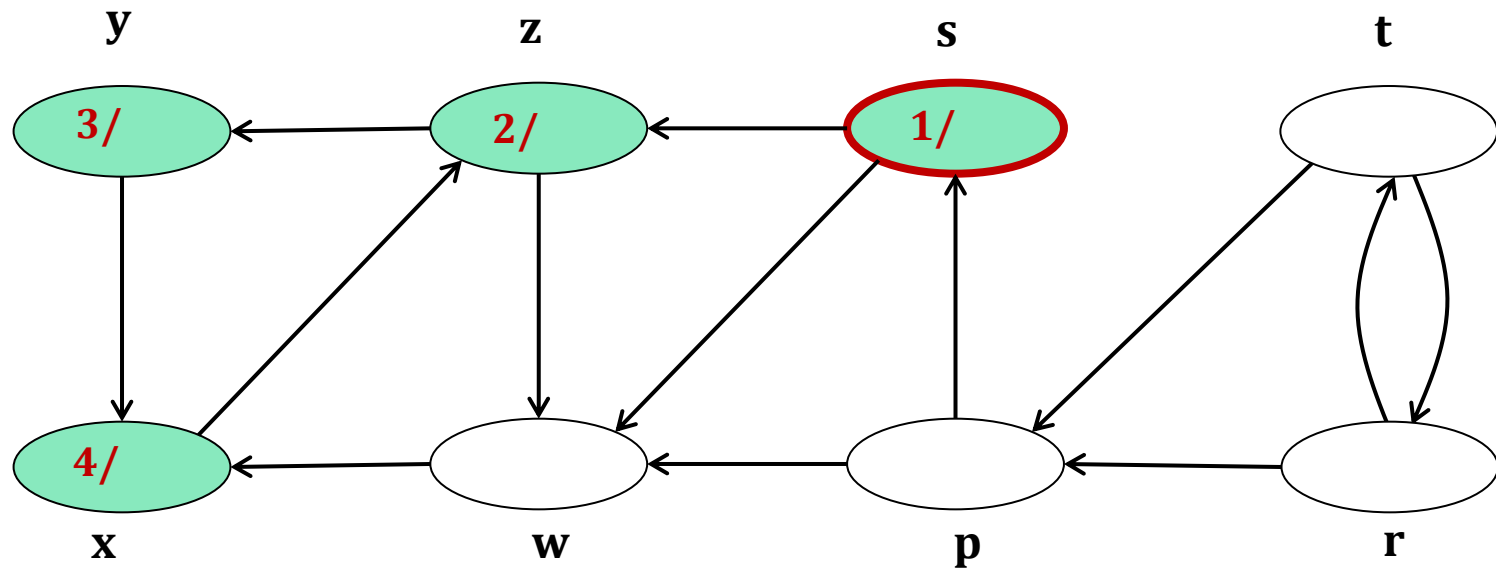
□ Παράδειγμα



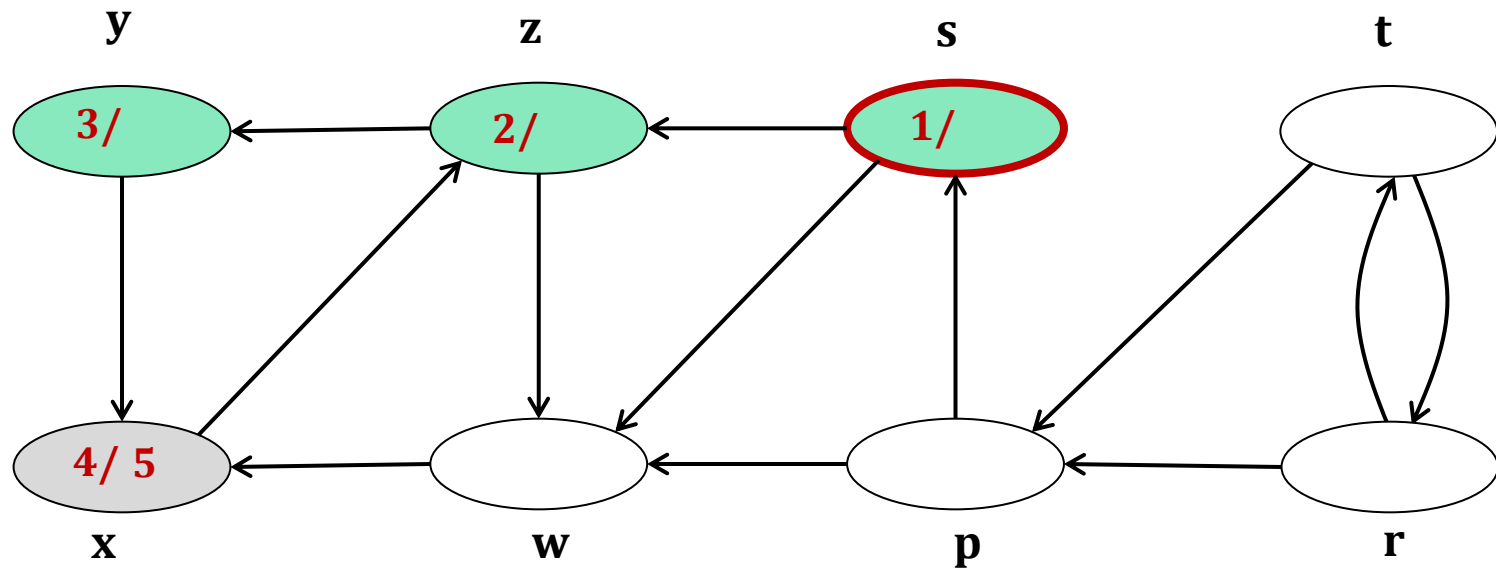
□ Παράδειγμα



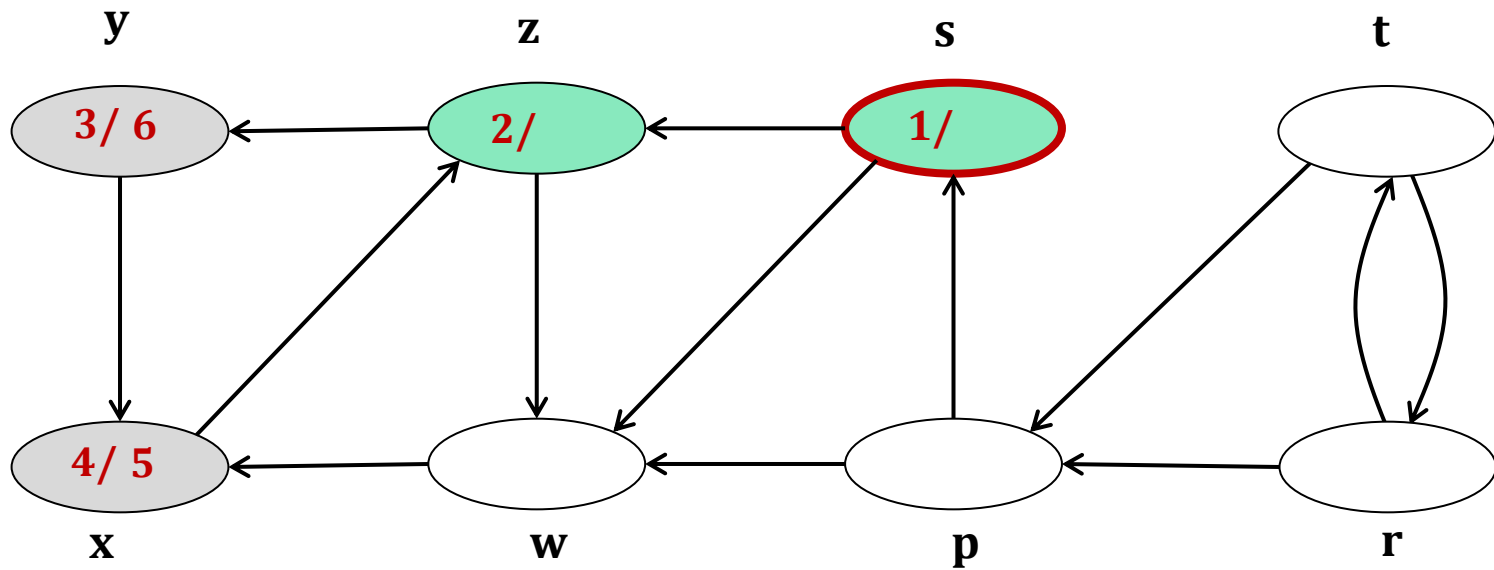
□ Παράδειγμα



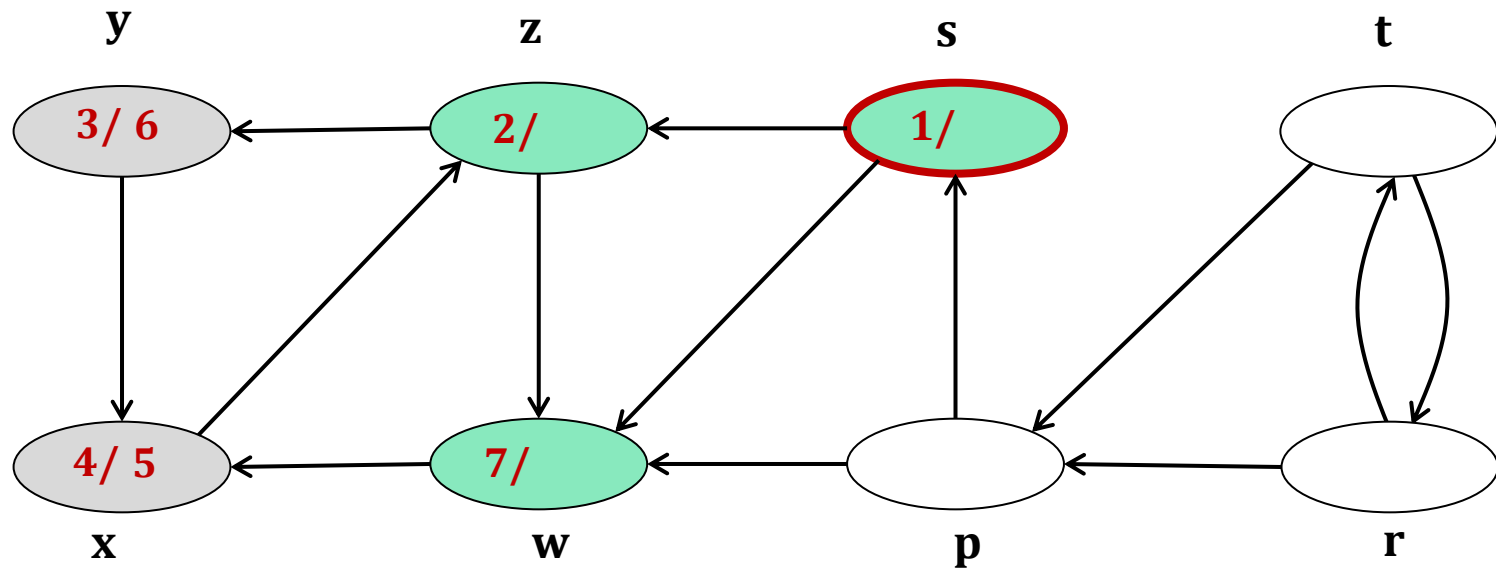
□ Παράδειγμα



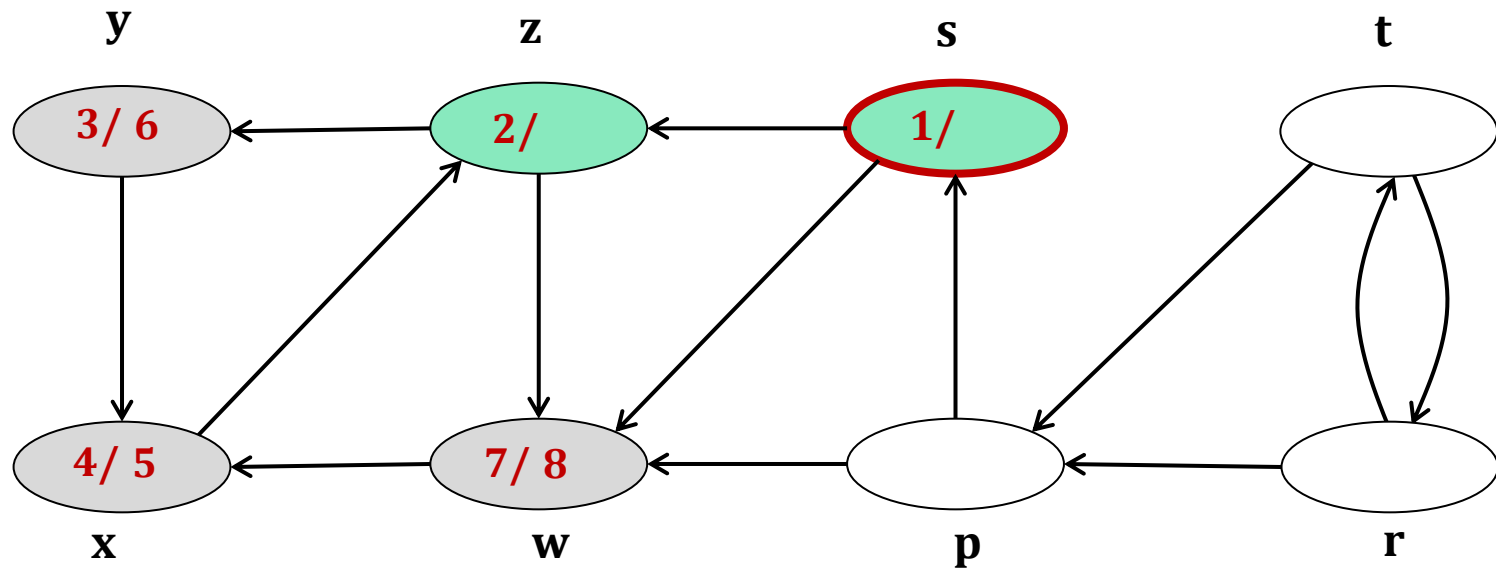
□ Παράδειγμα



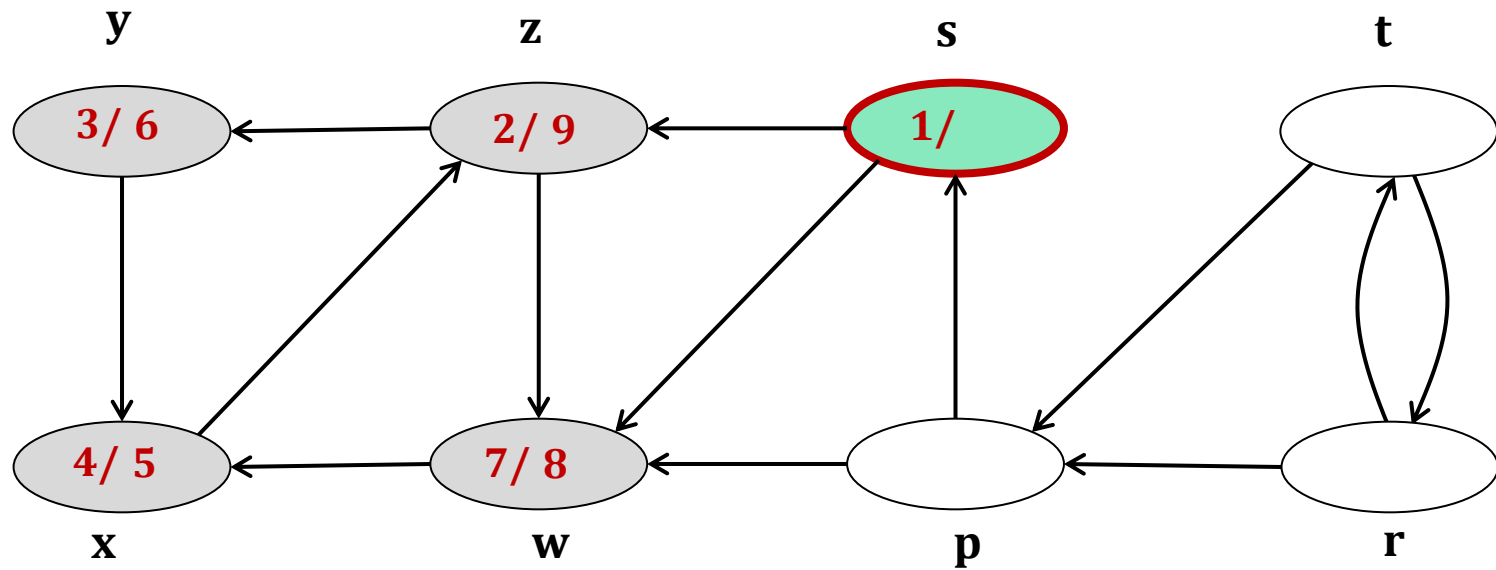
□ Παράδειγμα



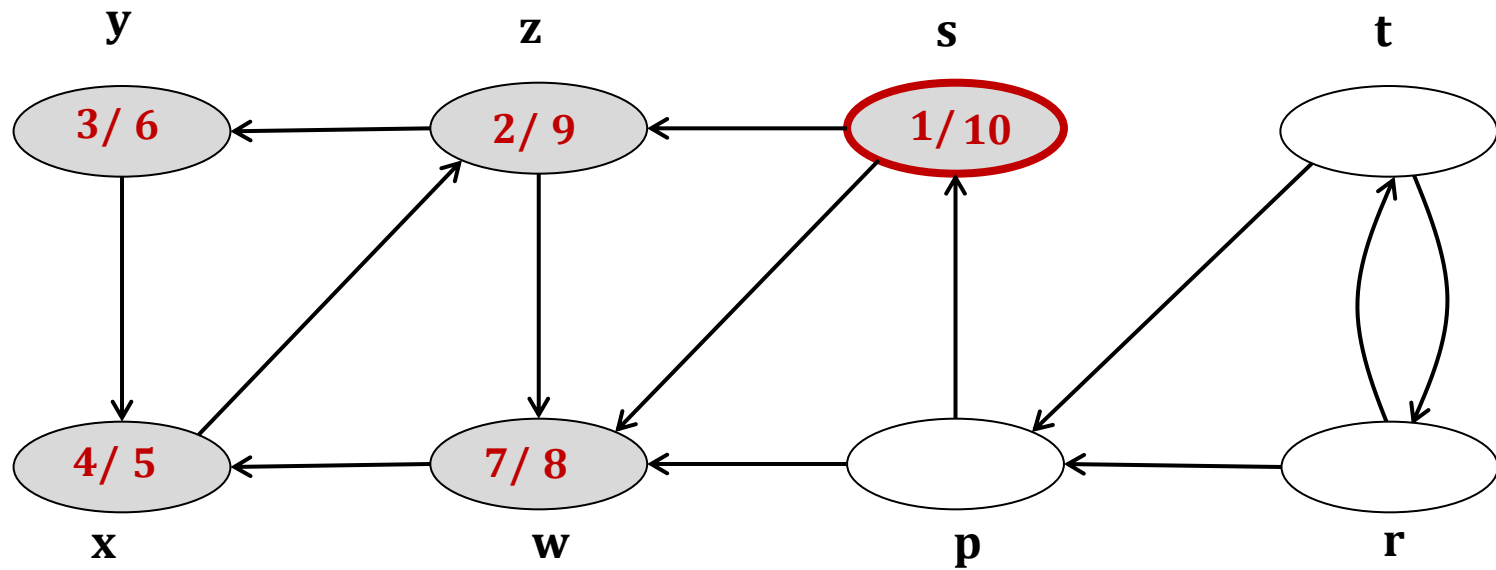
□ Παράδειγμα



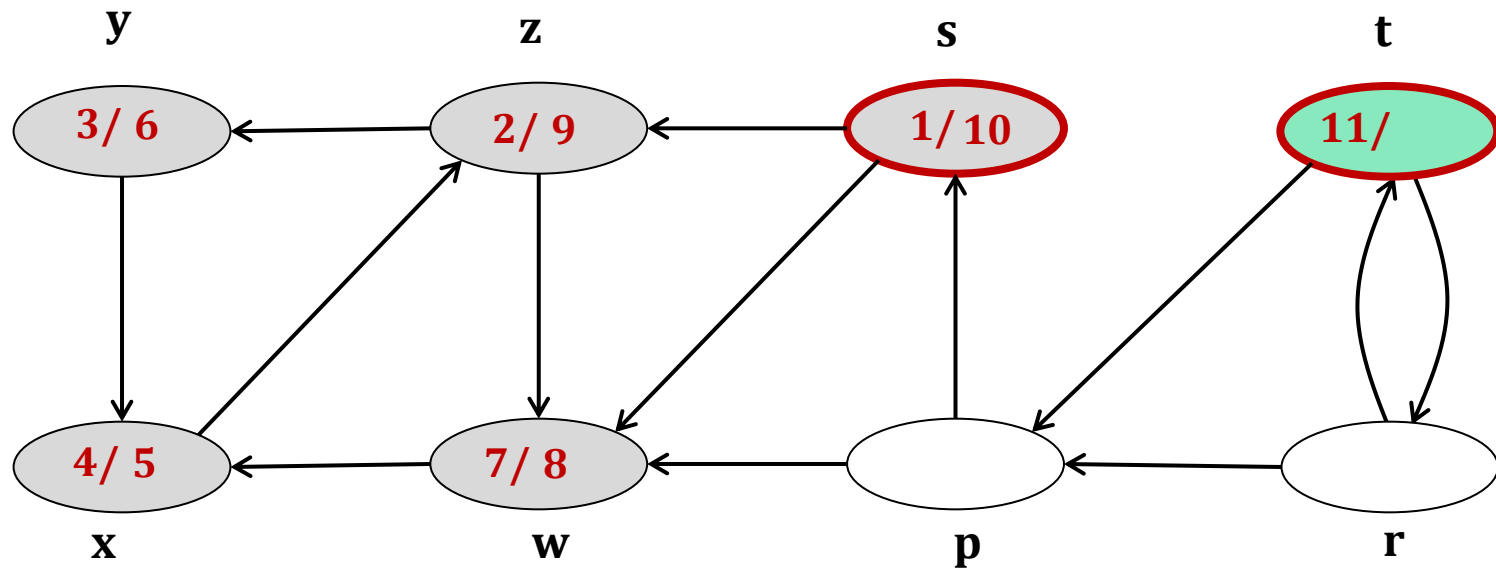
□ Παράδειγμα



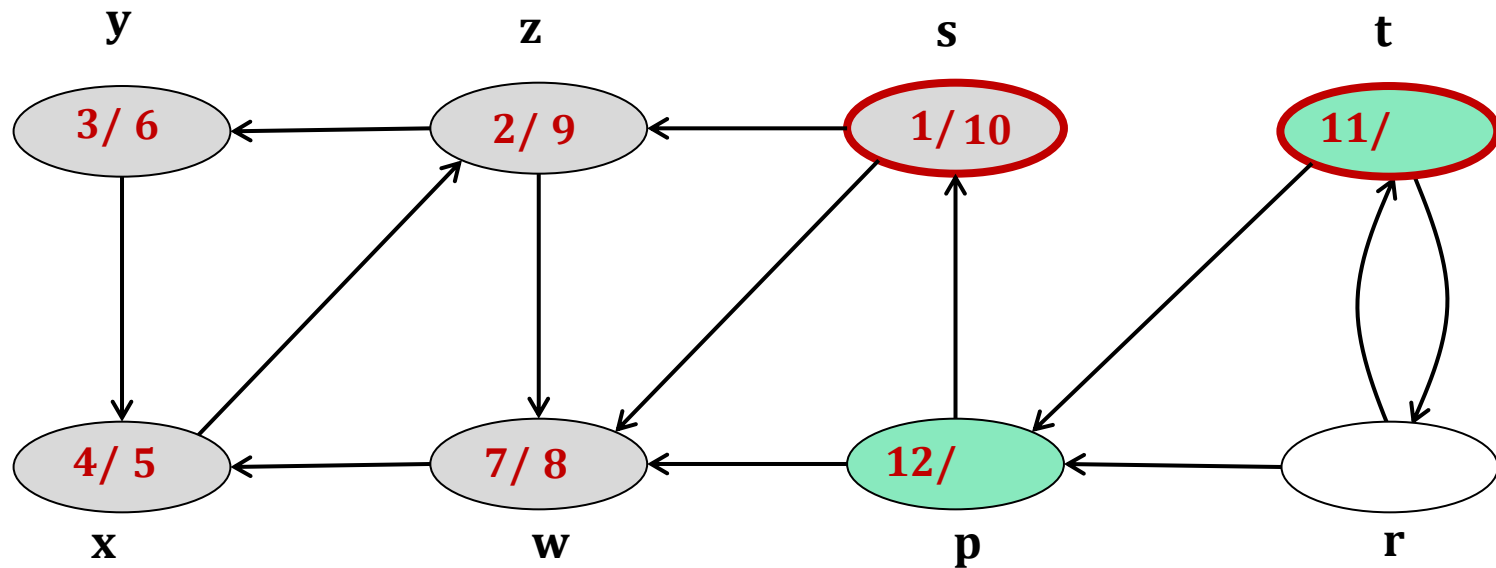
□ Παράδειγμα



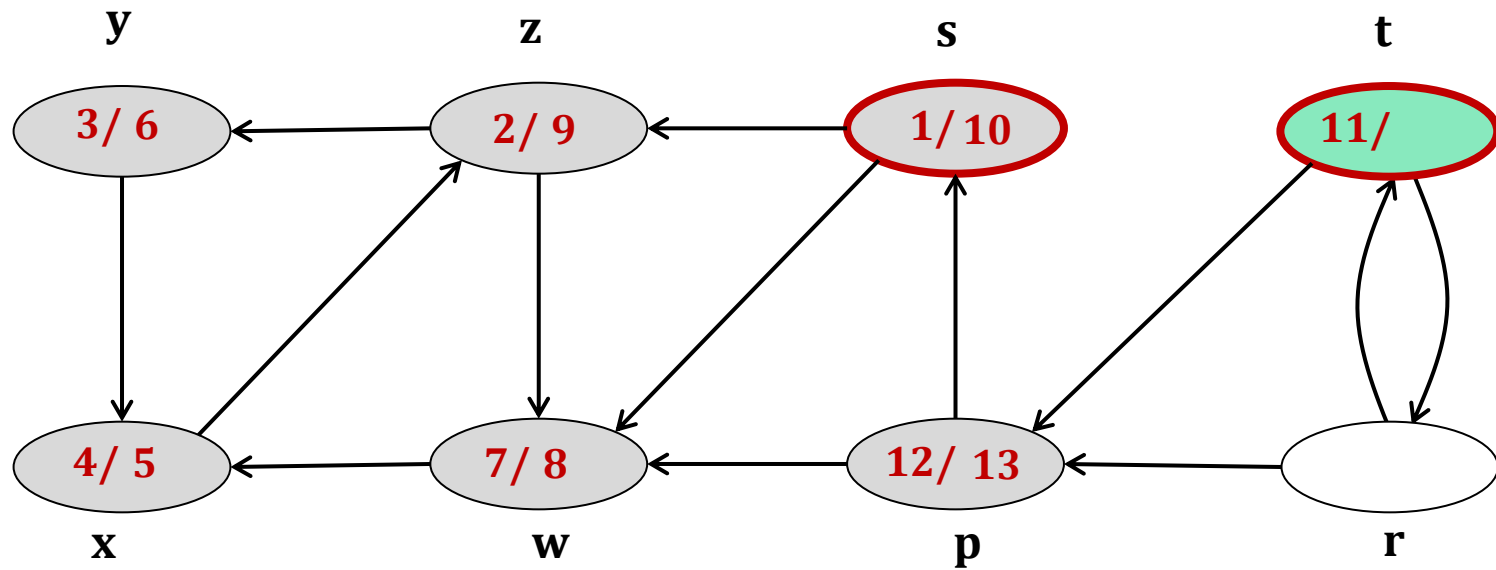
□ Παράδειγμα



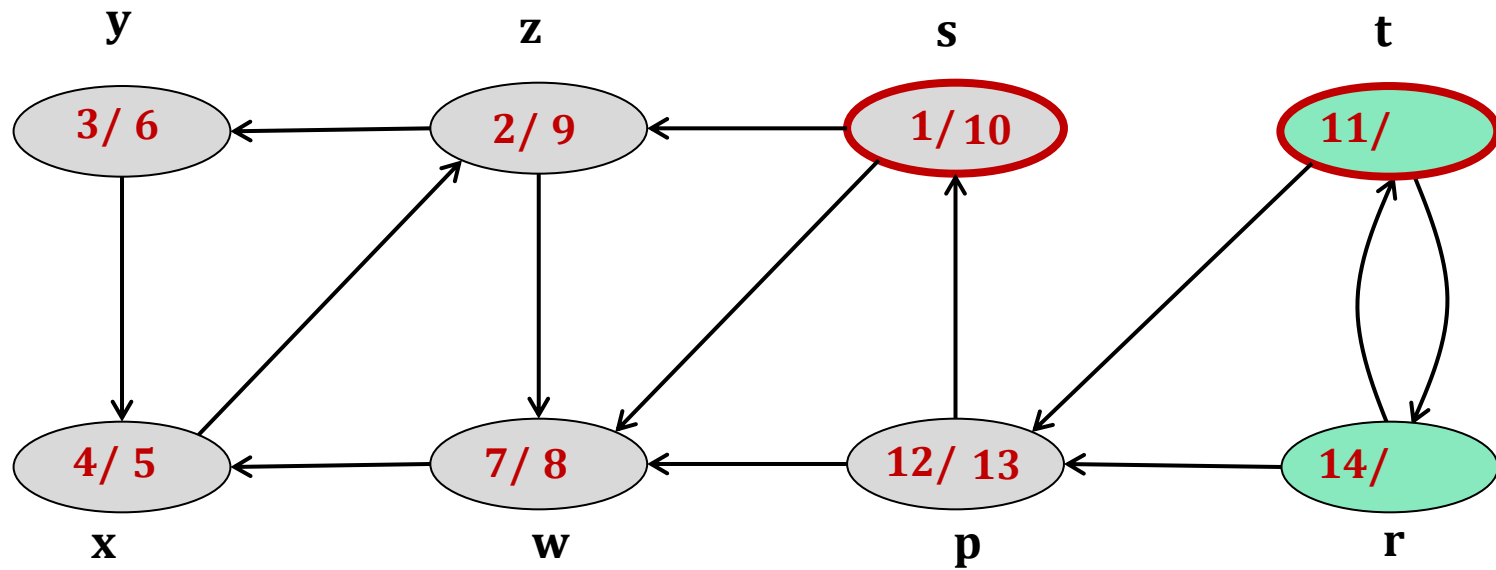
□ Παράδειγμα



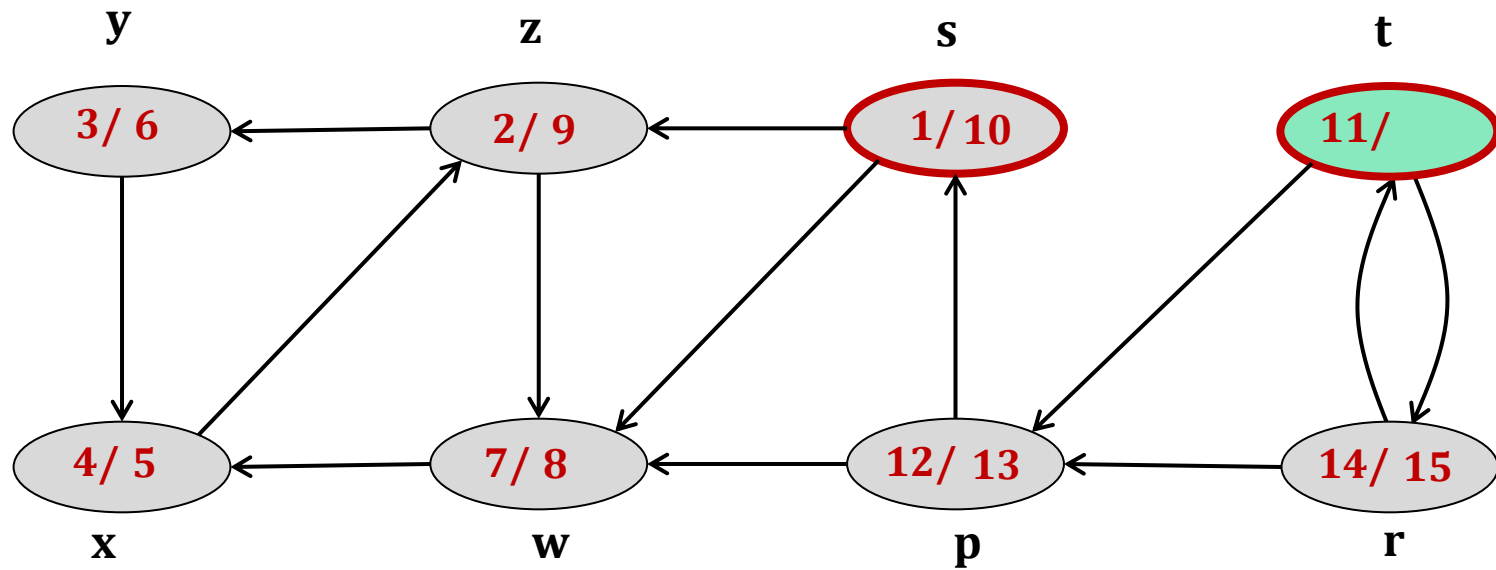
□ Παράδειγμα



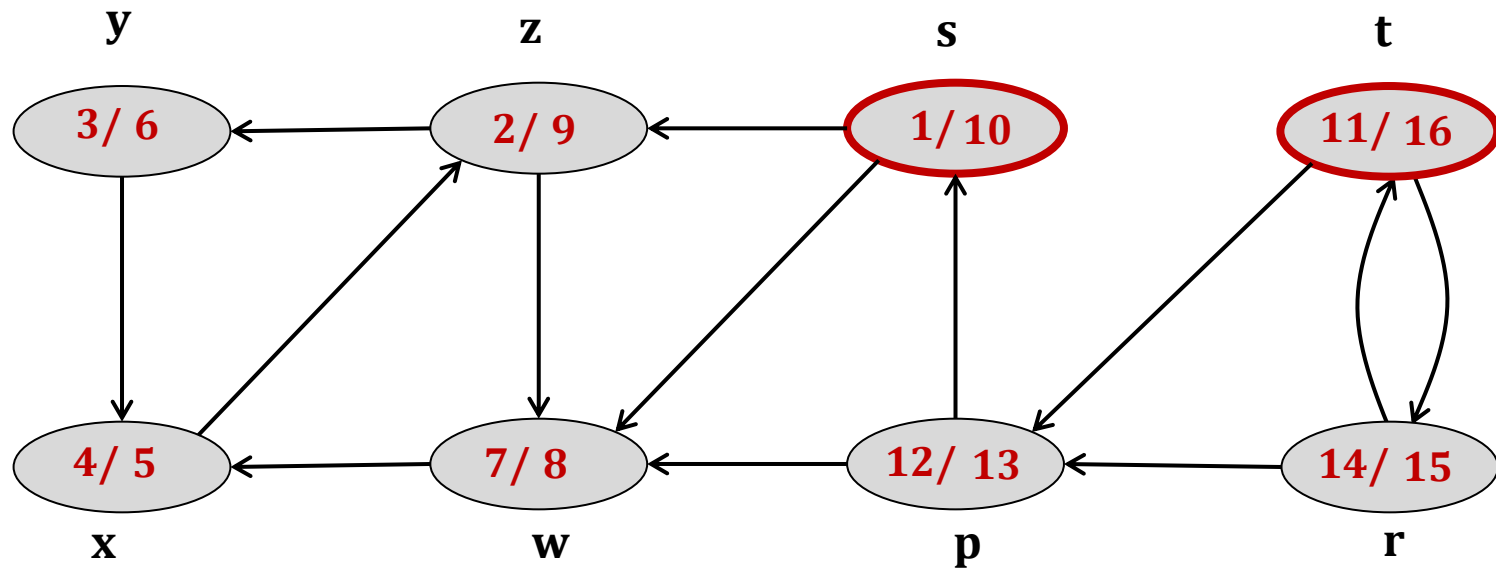
□ Παράδειγμα



□ Παράδειγμα



□ Παράδειγμα



DFS = (x, y, w, z, s, p, r, t)

■ Πολυπλοκότητα DFS

- ✓ Τα βήματα 1 και 3 εκτελούνται σε χρόνο $O(n)$
- ✓ Το βήμα 2 σε $O(1)$
- ✓ Η διαδικασία DFS-VISIT() καλείται 1 φορά για $\forall v \in V$
- ✓ Η DFS-VISIT(v) στο Βήμα 3 εκτελείται $|adj(v)|$ φορές.
- ✓ Άρα, $\sum_{v \in V} |adj(v)| = O(m)$

Πολυπλοκότητα DFS: $O(n + m)$

1. Για κάθε κόμβο $x \in V$ επανάλαβε
 $Color(x) \leftarrow A$ και $\pi(x) \leftarrow NIL$
2. $Time \leftarrow 0$
3. Εφόσον η **υπάρχουν A κόμβοι** στο G επανάλαβε
επίλεξε ένα A κόμβο εκκίνησης $s \in V$ και
κάλεσε την διαδικασία **DFS-VISIT(s)**

Διαδικασία DFS-VISIT(u)

1. $Color(u) \leftarrow \Pi$
2. $A(u) \leftarrow Time \leftarrow Time + 1$
3. Για κάθε κόμβο $x \in adj(u)$ επανάλαβε
εάν $Color(x) = A$ τότε
 $\pi(x) \leftarrow u$ και DFS-VISIT(x)
4. $Color(u) \leftarrow M$
5. $T(u) \leftarrow Time \leftarrow Time + 1$

■ Κατηγορίες Ακμών κατά την DFS

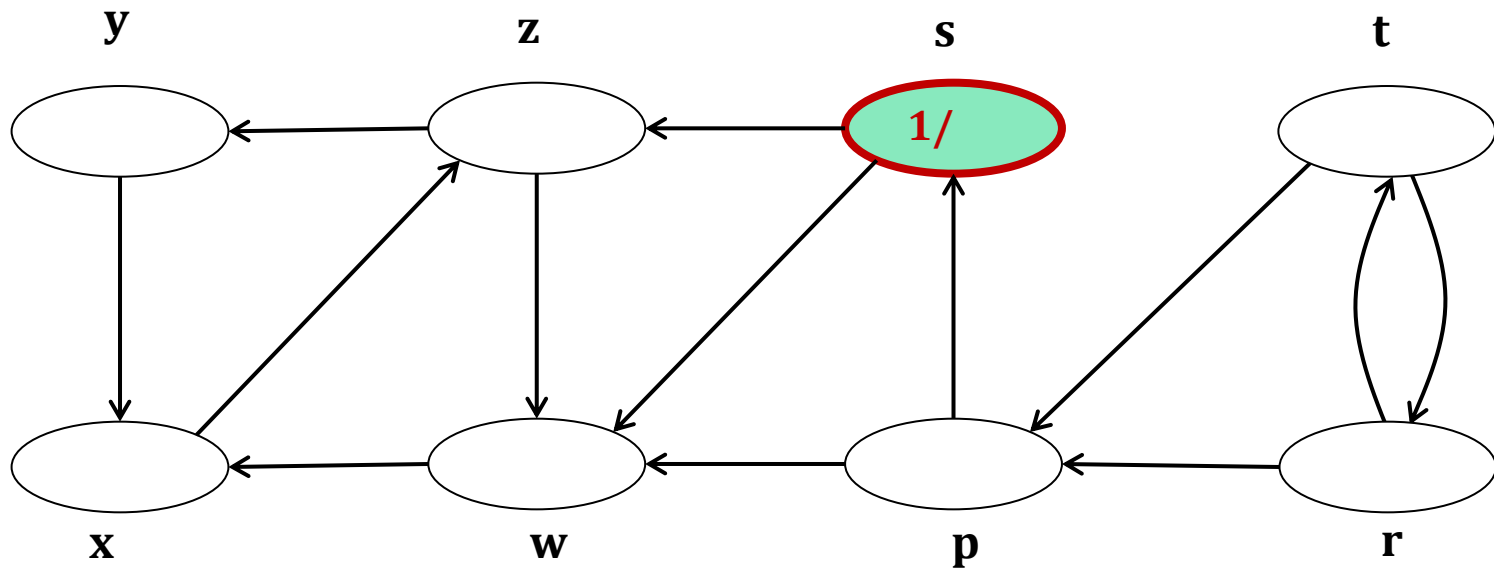
T : Δεντρικές-ακμές (tree-edges)

B : Οπισθο-ακμές ή ανιούσες (back-edges)

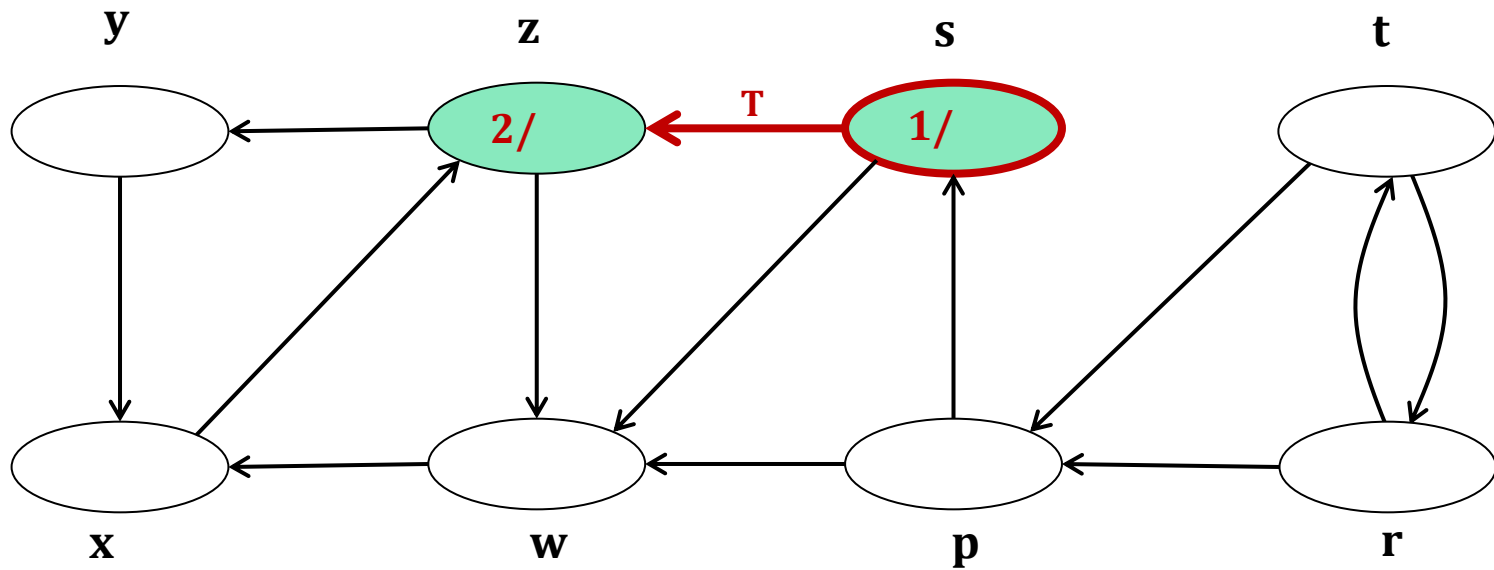
F : Εμπρόσθιες-ακμές ή κατιούσες (forward-edges)

C : Διασχίζουσες ή εγκάρσιες-ακμές (cross-edges)

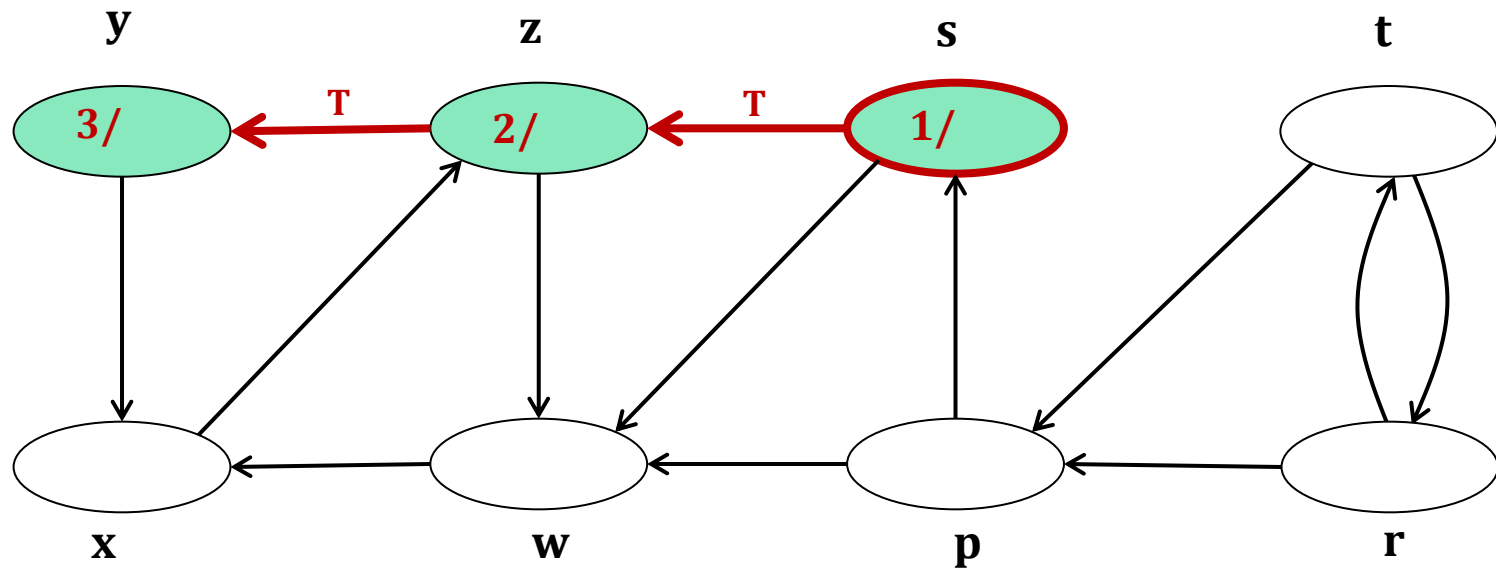
□ Παράδειγμα



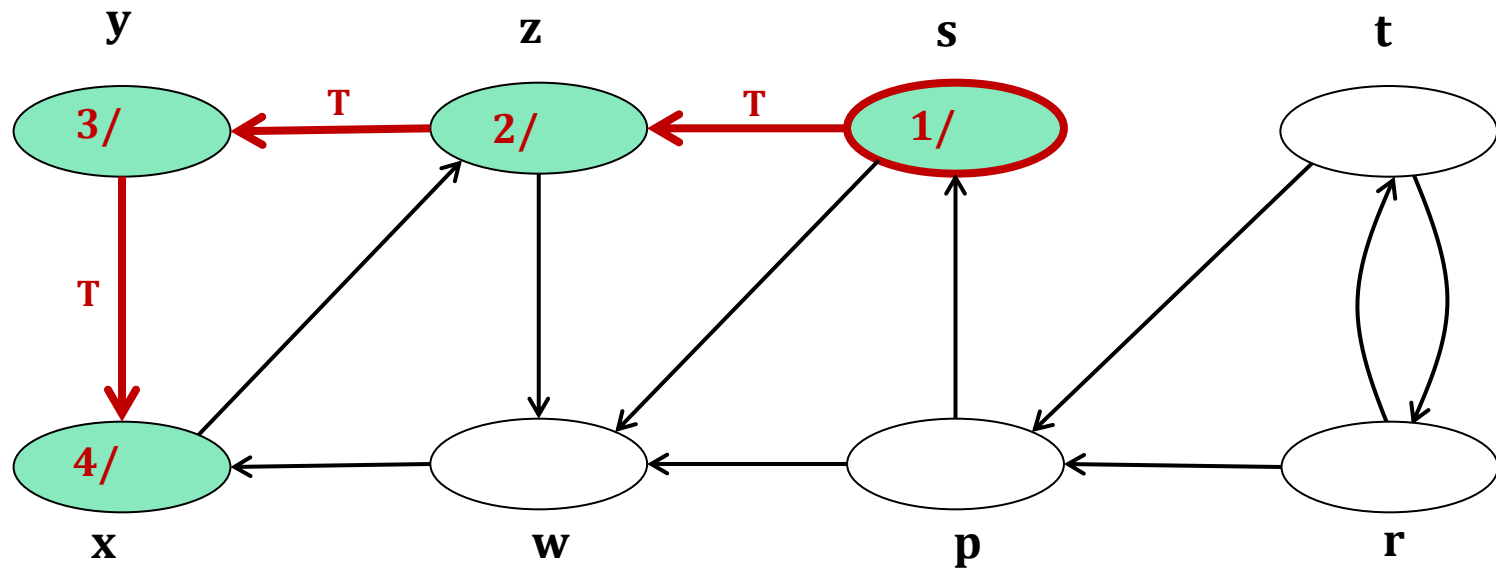
□ Παράδειγμα



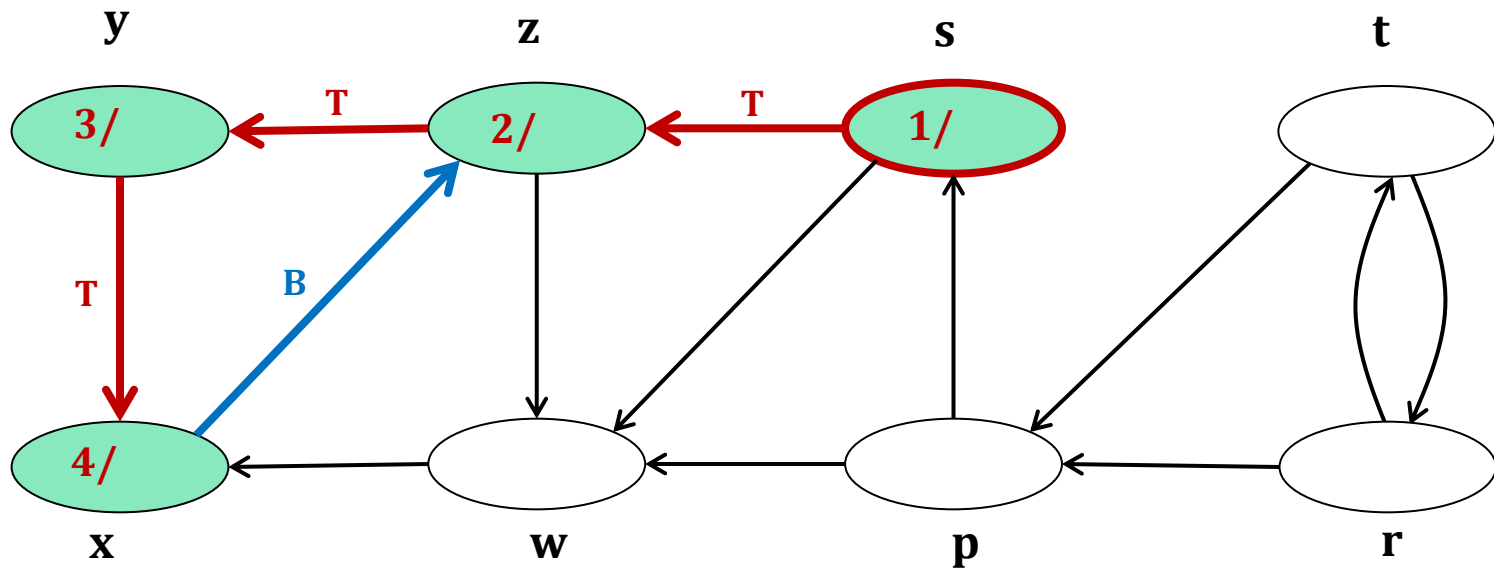
□ Παράδειγμα



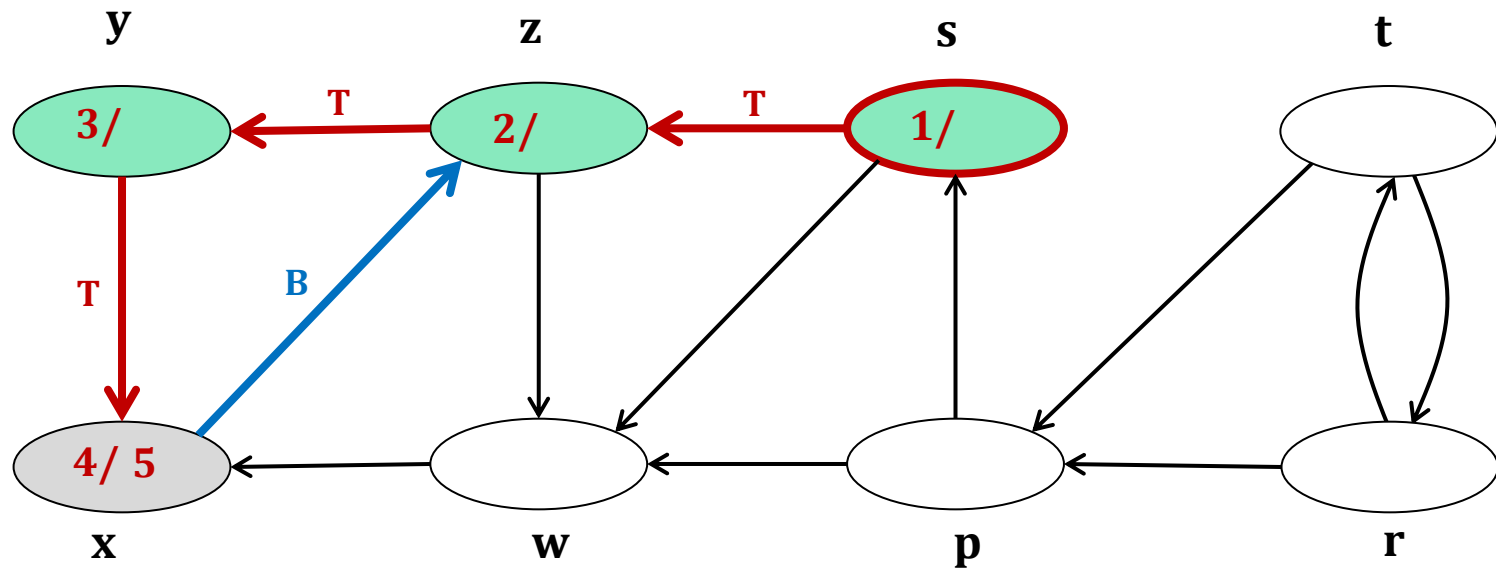
□ Παράδειγμα



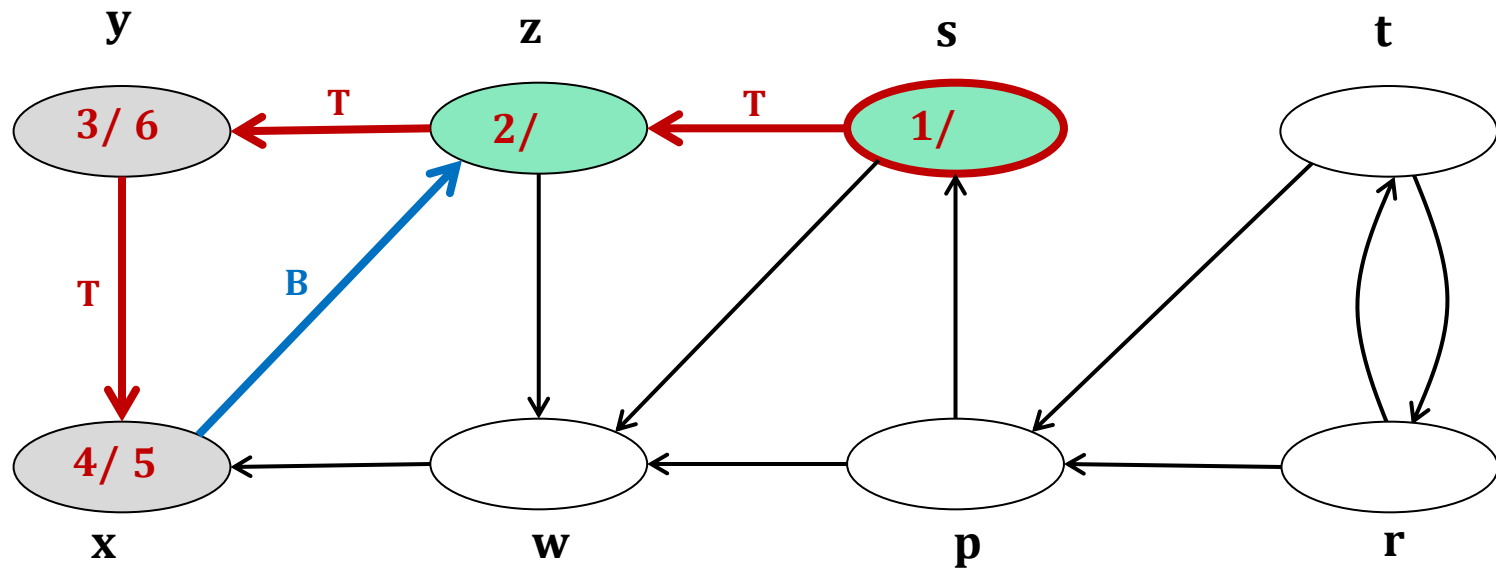
□ Παράδειγμα



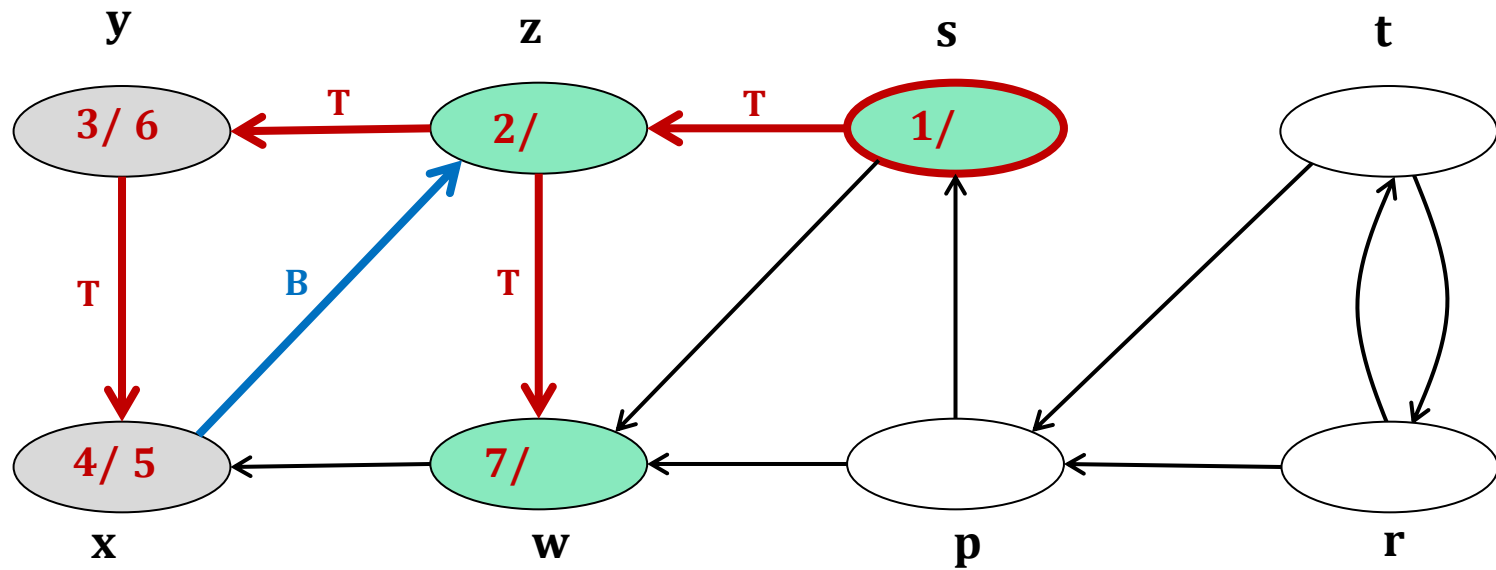
□ Παράδειγμα



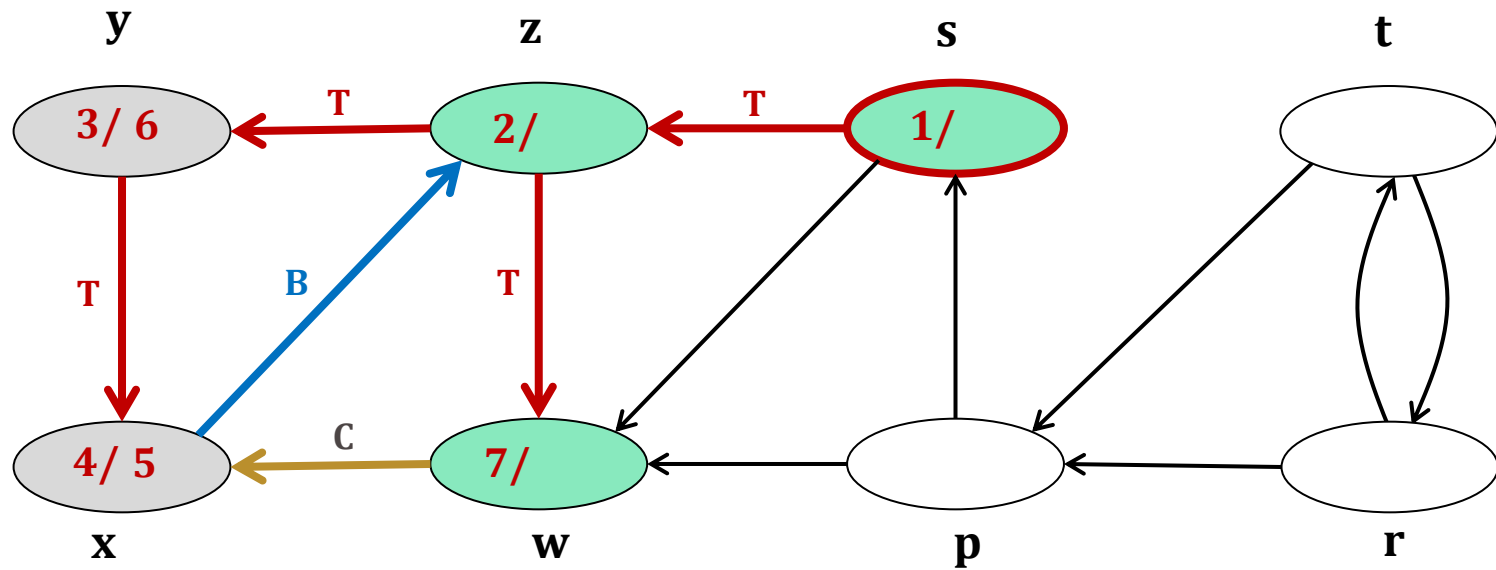
□ Παράδειγμα



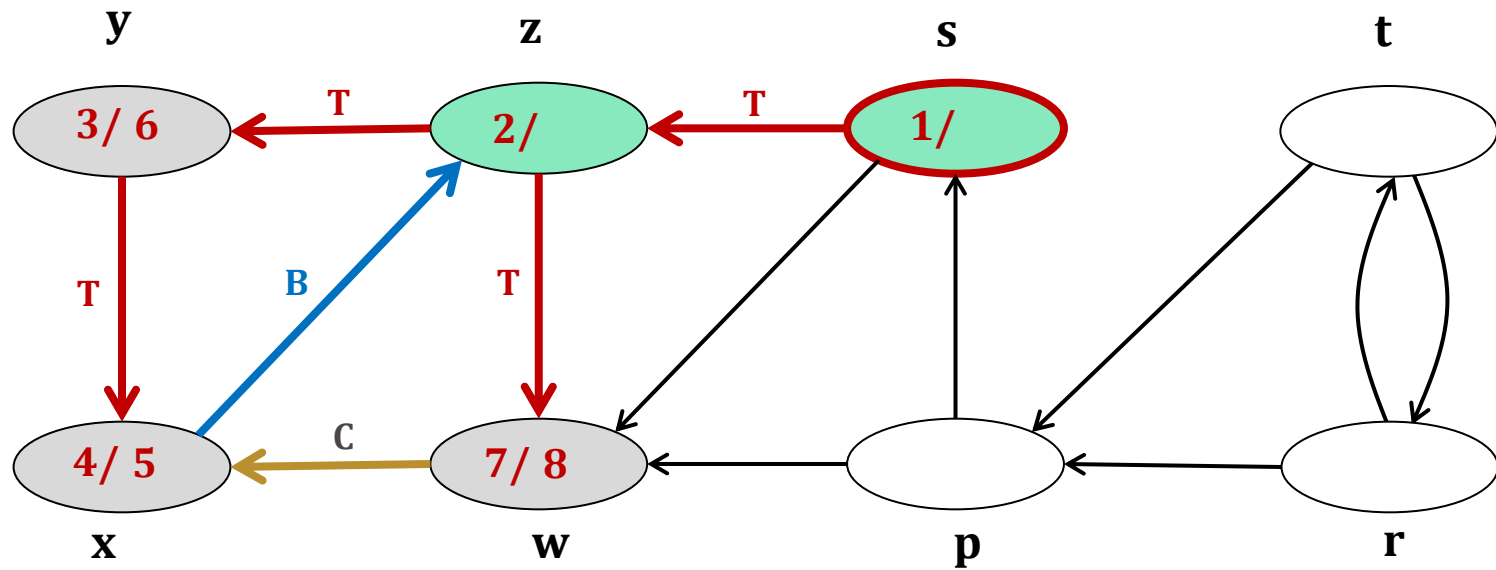
□ Παράδειγμα



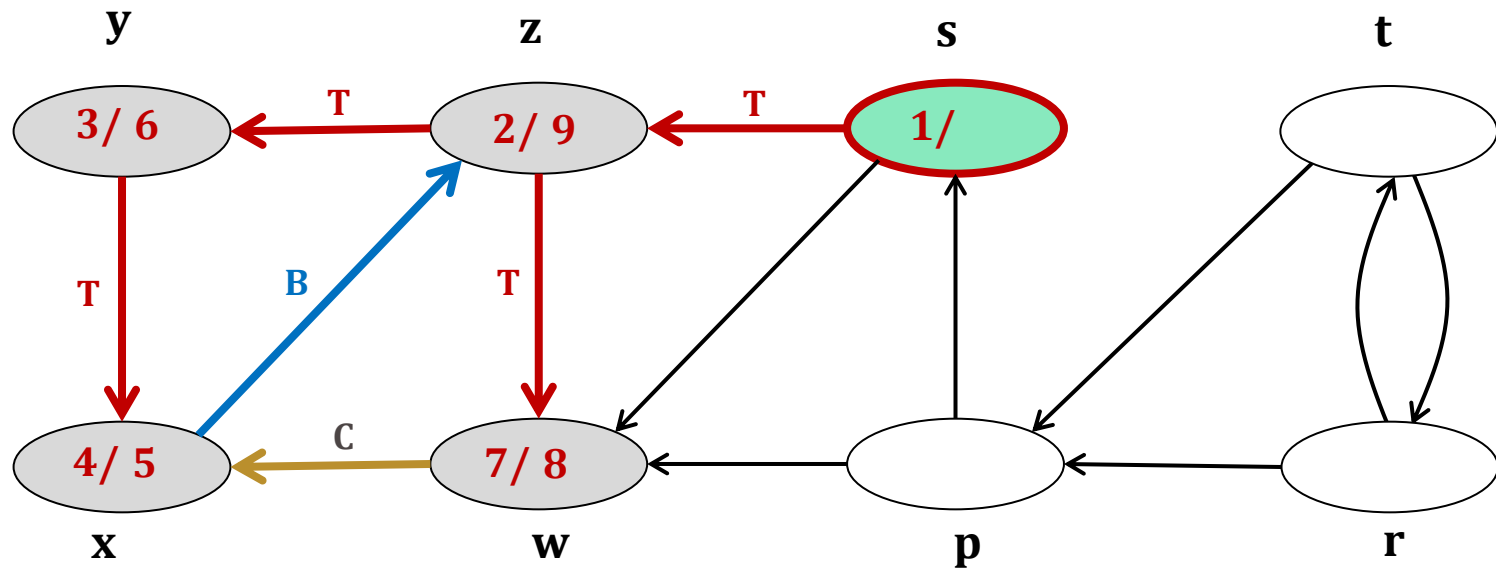
□ Παράδειγμα



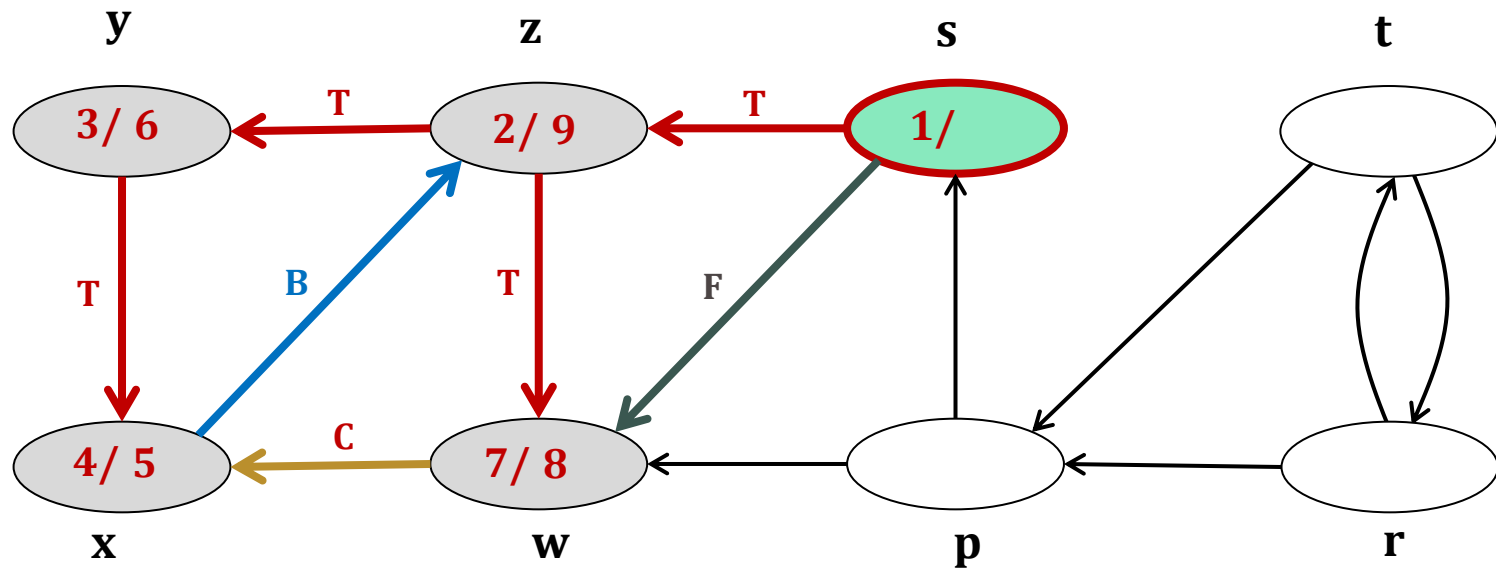
□ Παράδειγμα



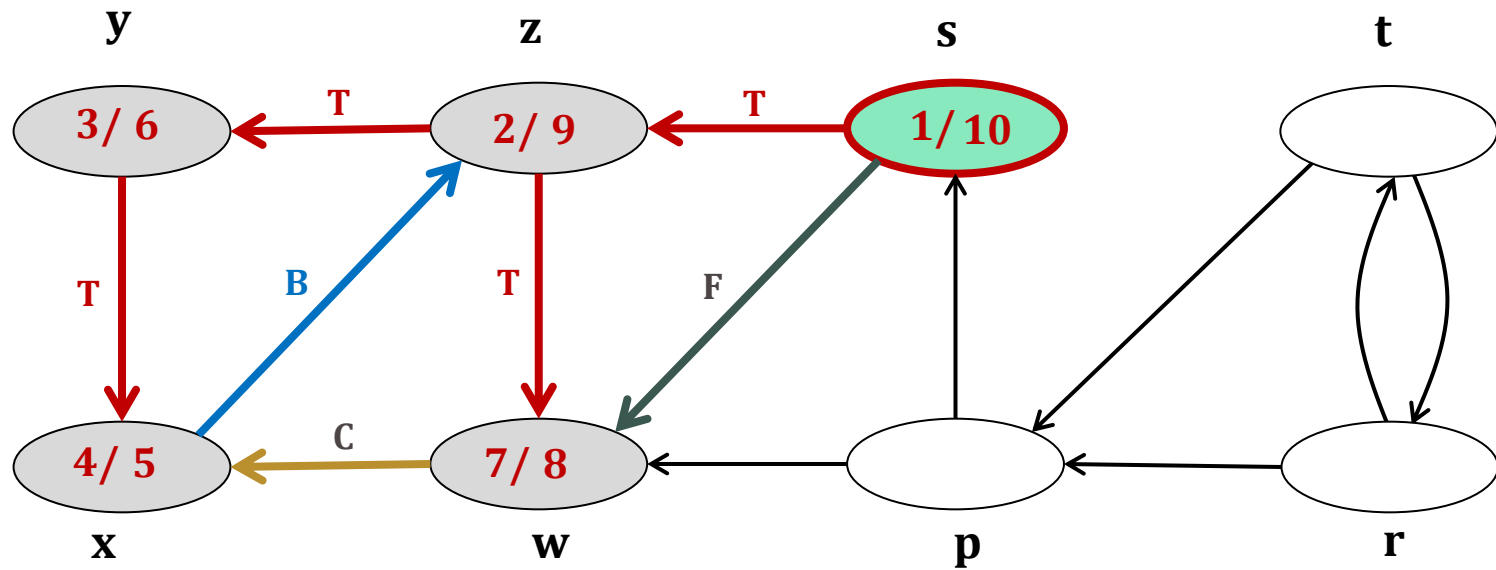
□ Παράδειγμα



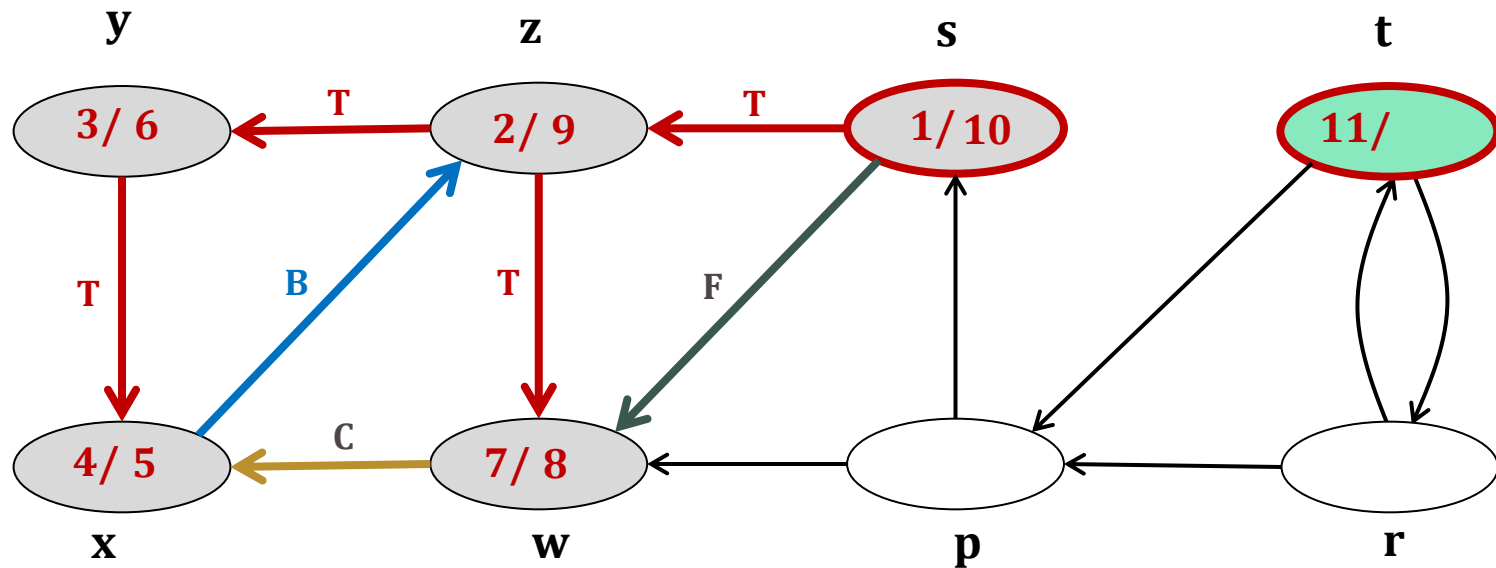
□ Παράδειγμα



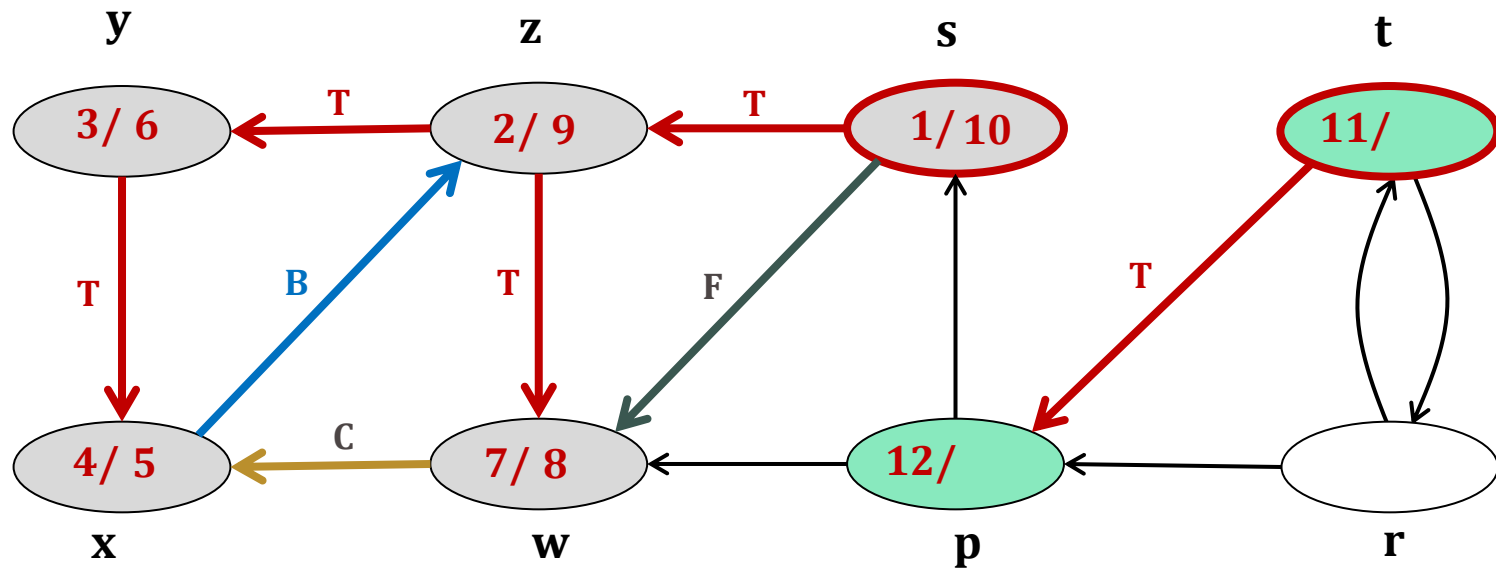
□ Παράδειγμα



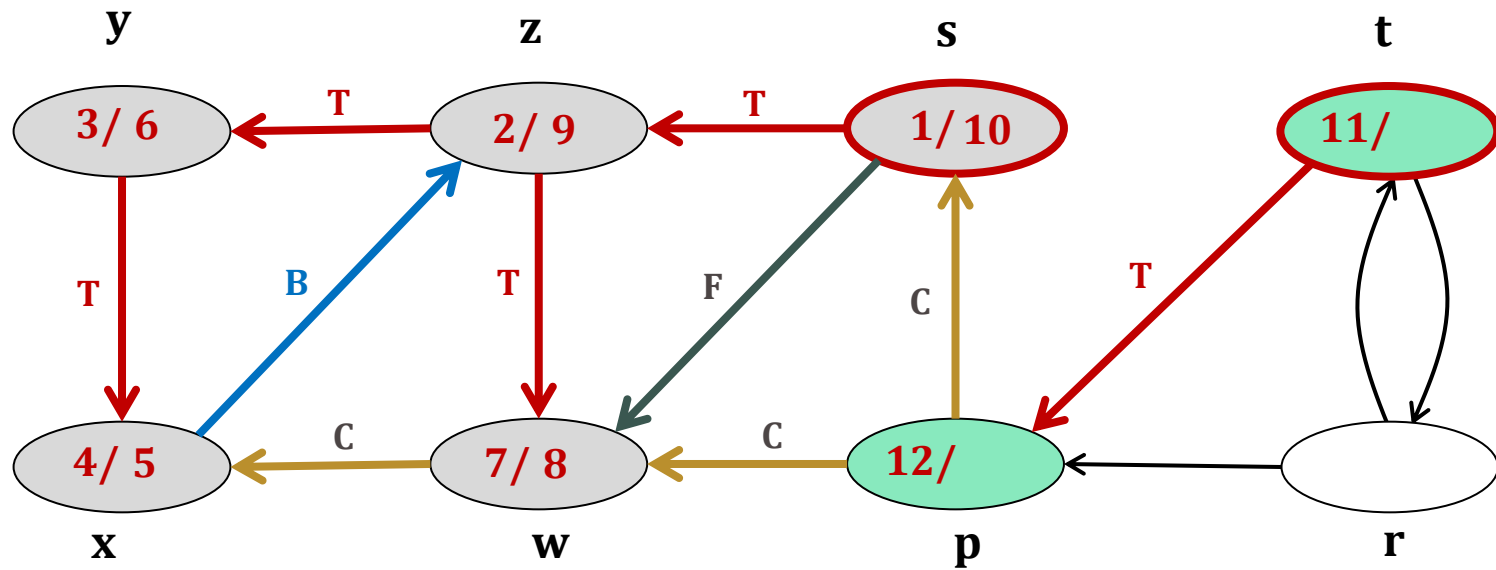
□ Παράδειγμα



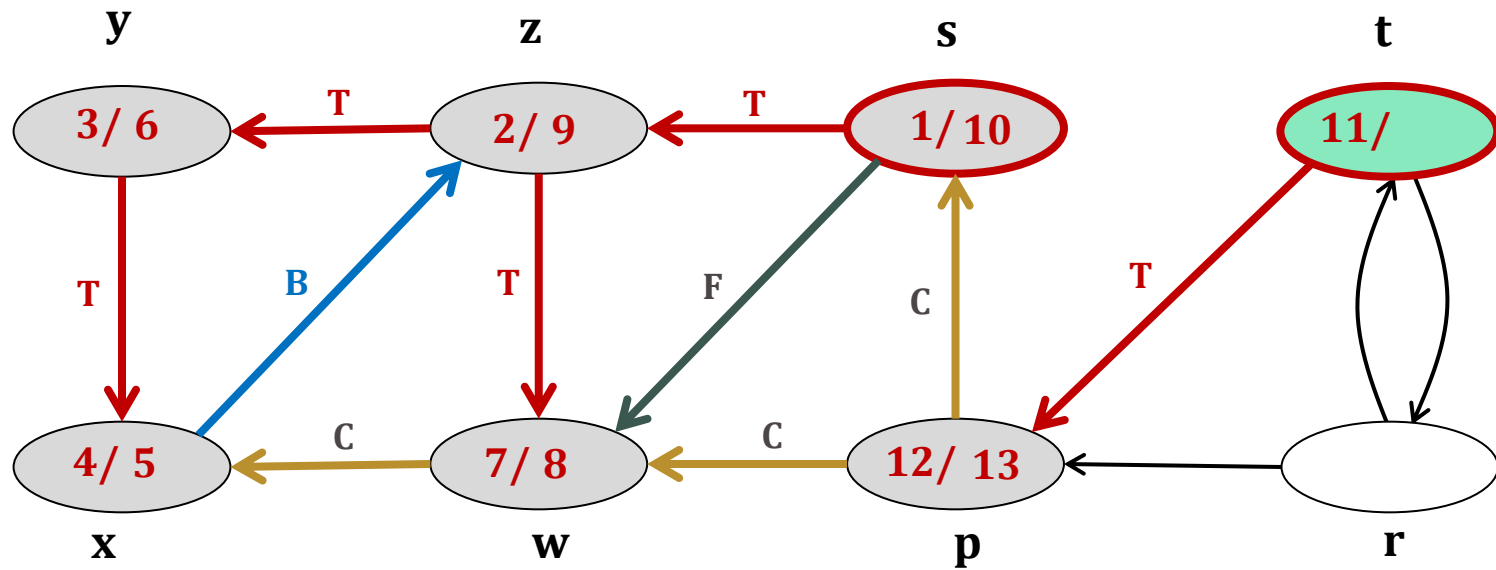
□ Παράδειγμα



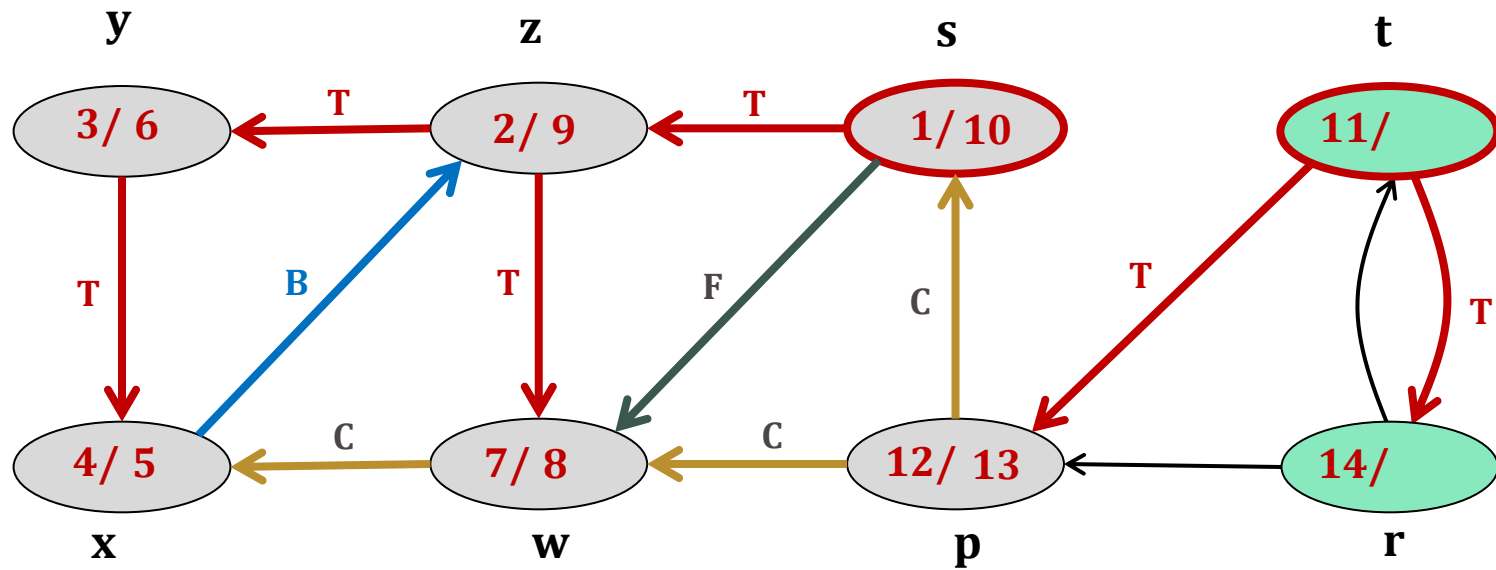
□ Παράδειγμα



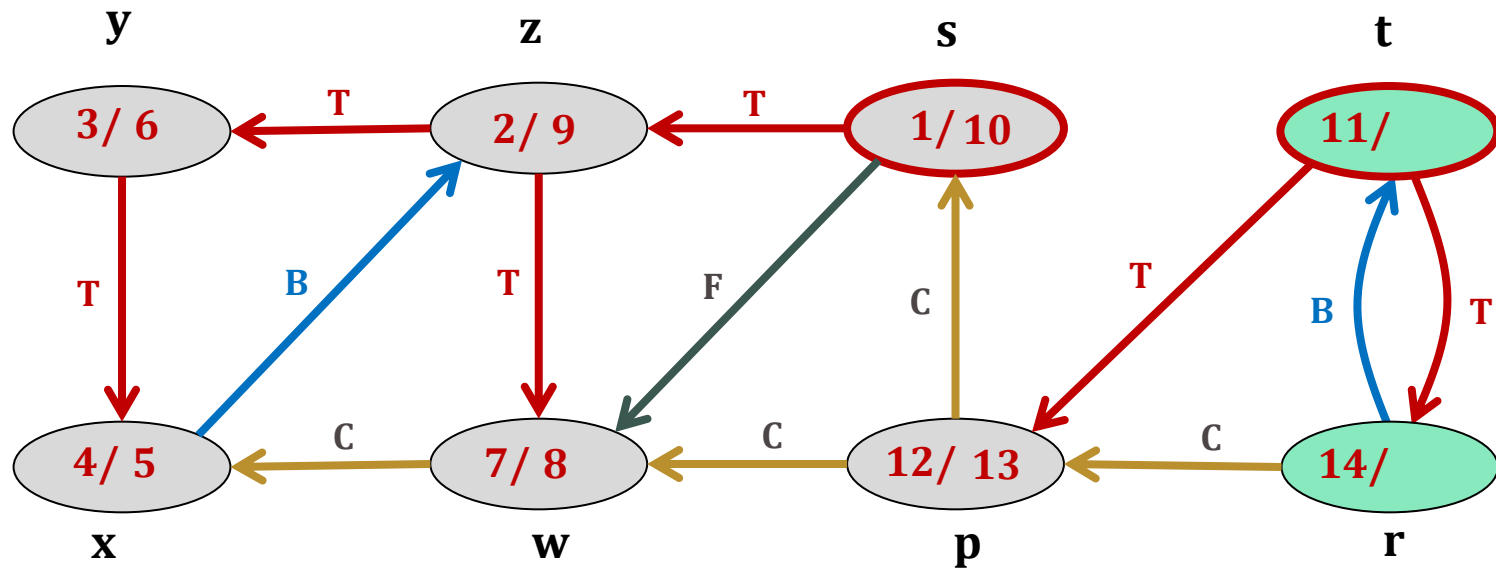
□ Παράδειγμα



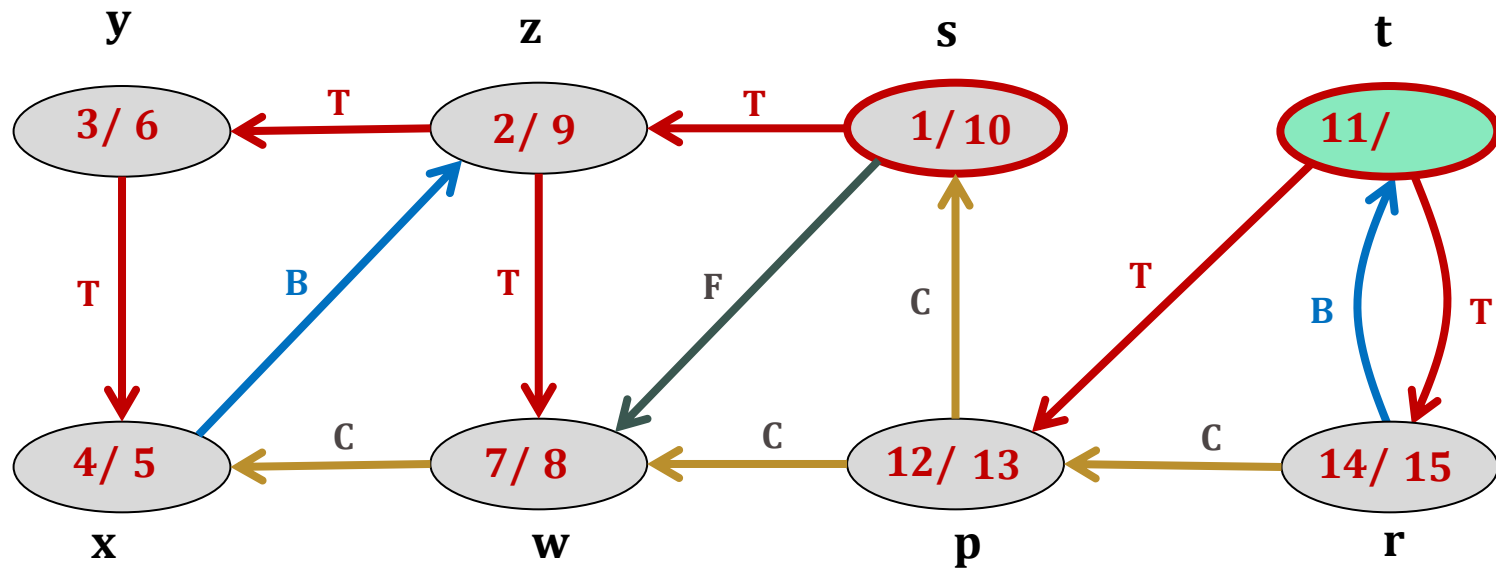
□ Παράδειγμα



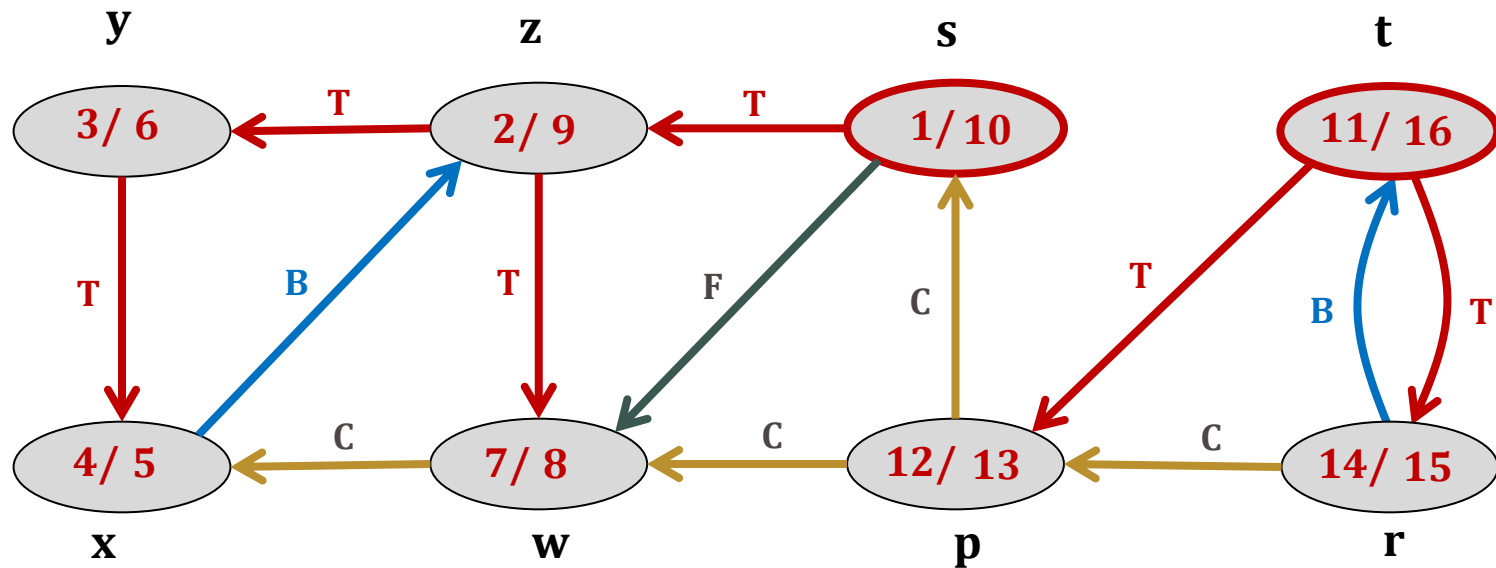
□ Παράδειγμα



□ Παράδειγμα



□ Παράδειγμα

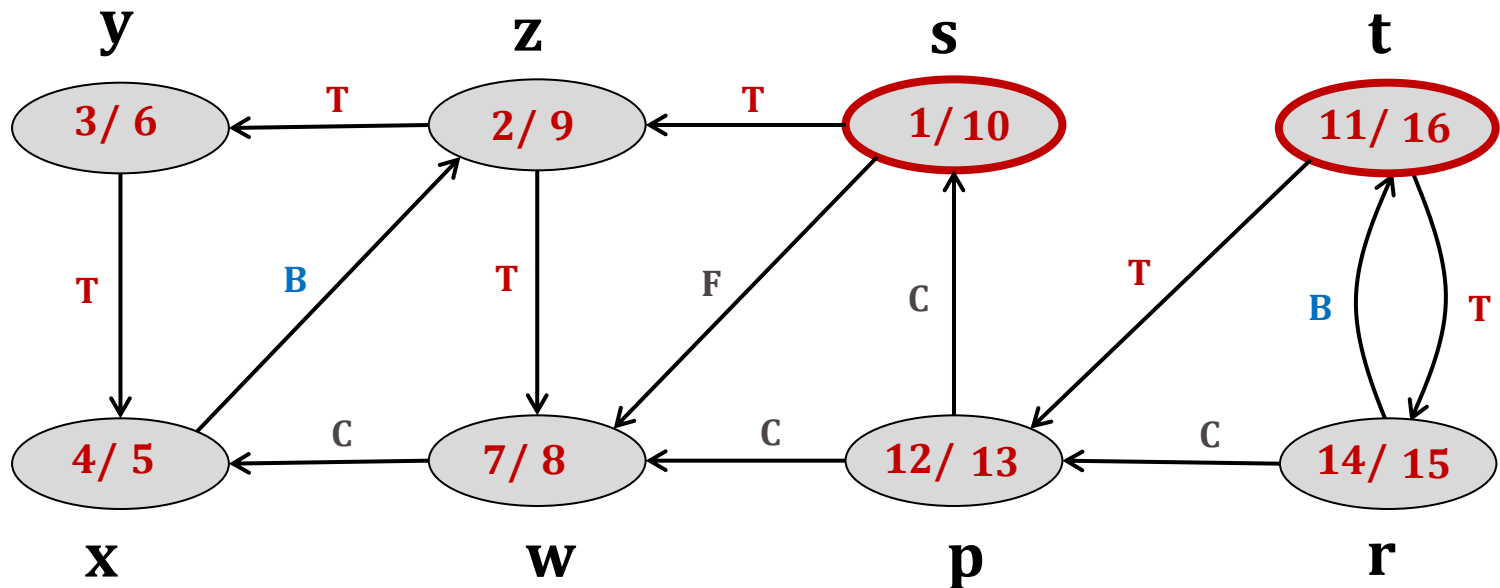
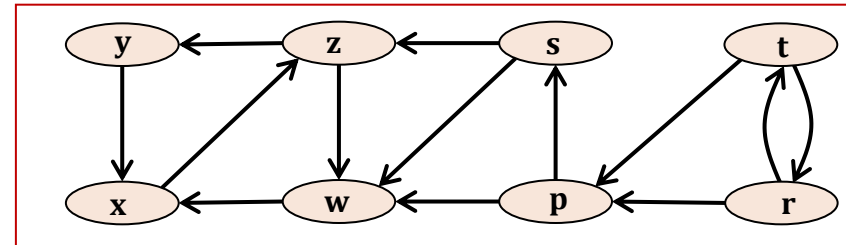


T & F: $a(x) < t(y) < a(y) < t(x)$

B: $a(y) < a(x) < t(y) < t(y)$

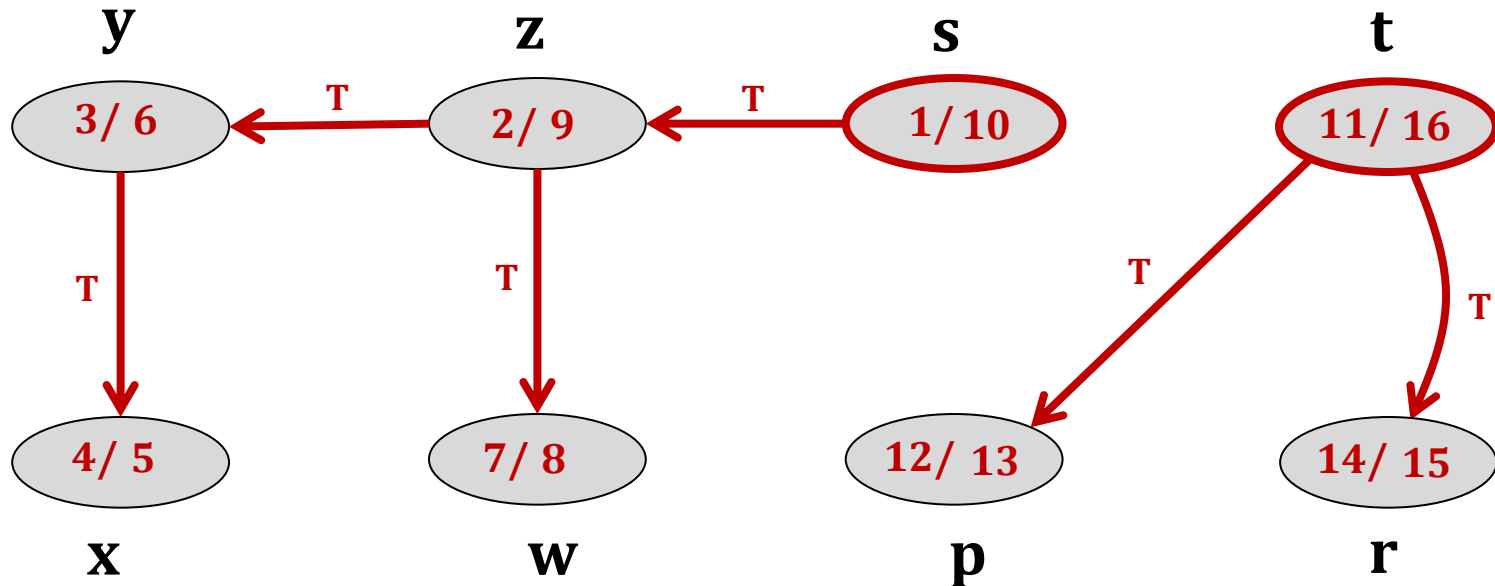
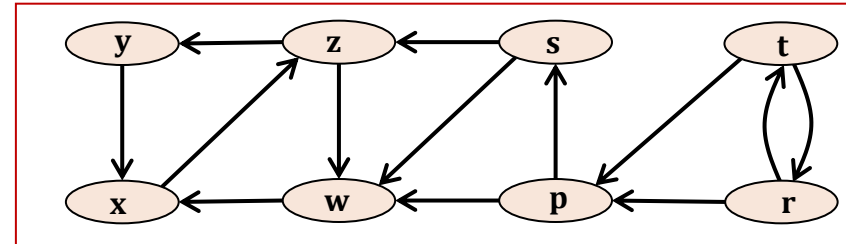
C: $a(y) < t(y) < a(x) < t(x)$

□ DFS-δένδρο (δάσος)

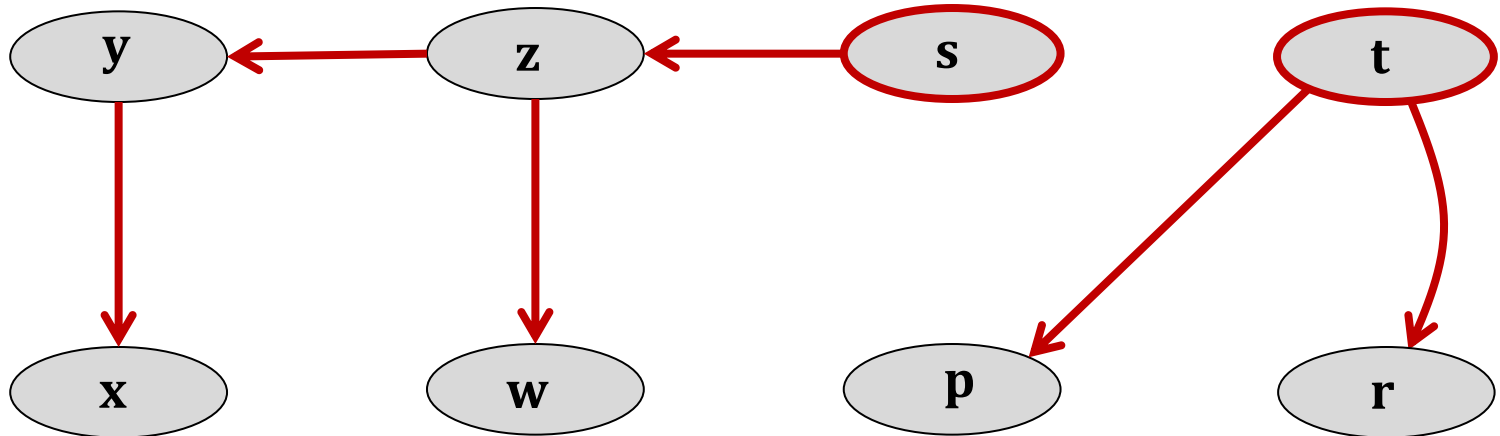
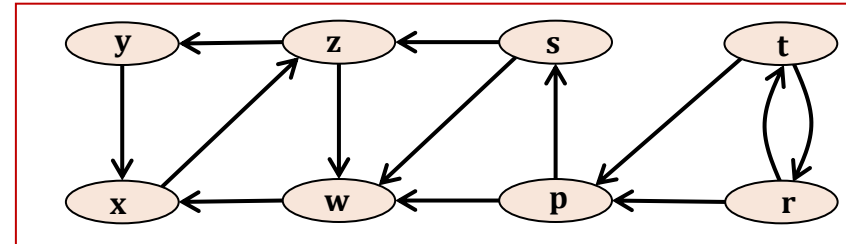


Διερεύνηση κατά Βάθος - DFS

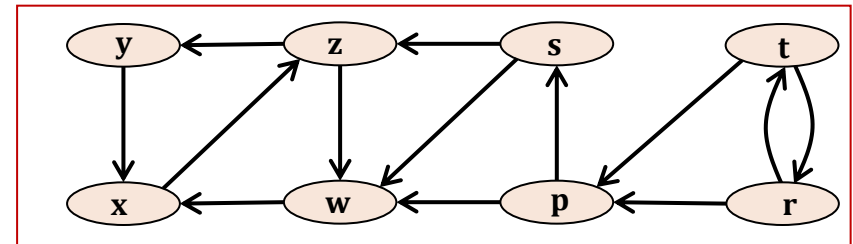
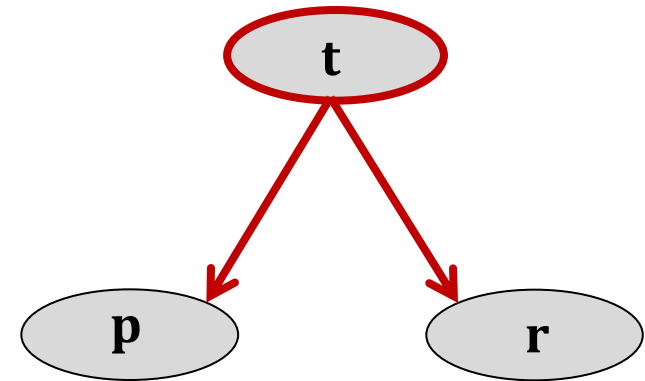
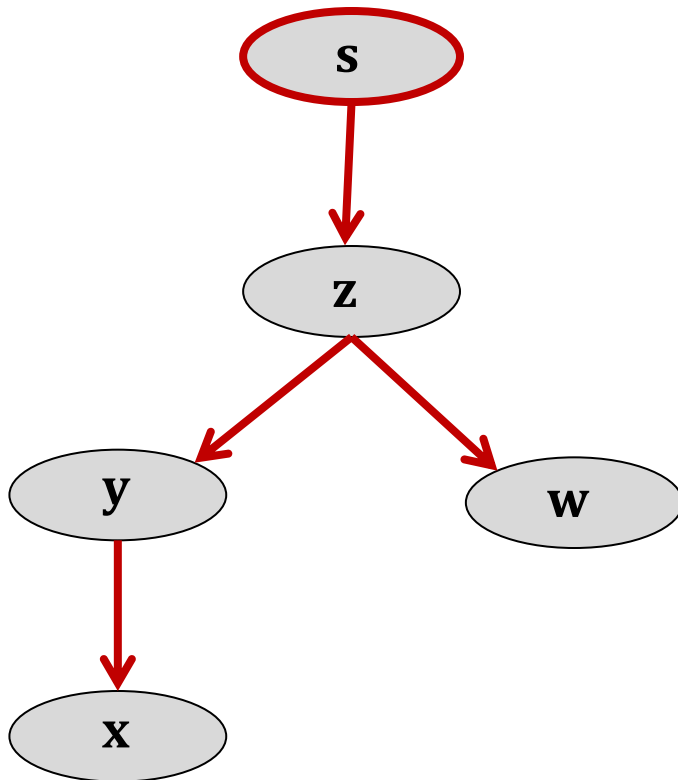
□ DFS-δένδρο (δάσος)



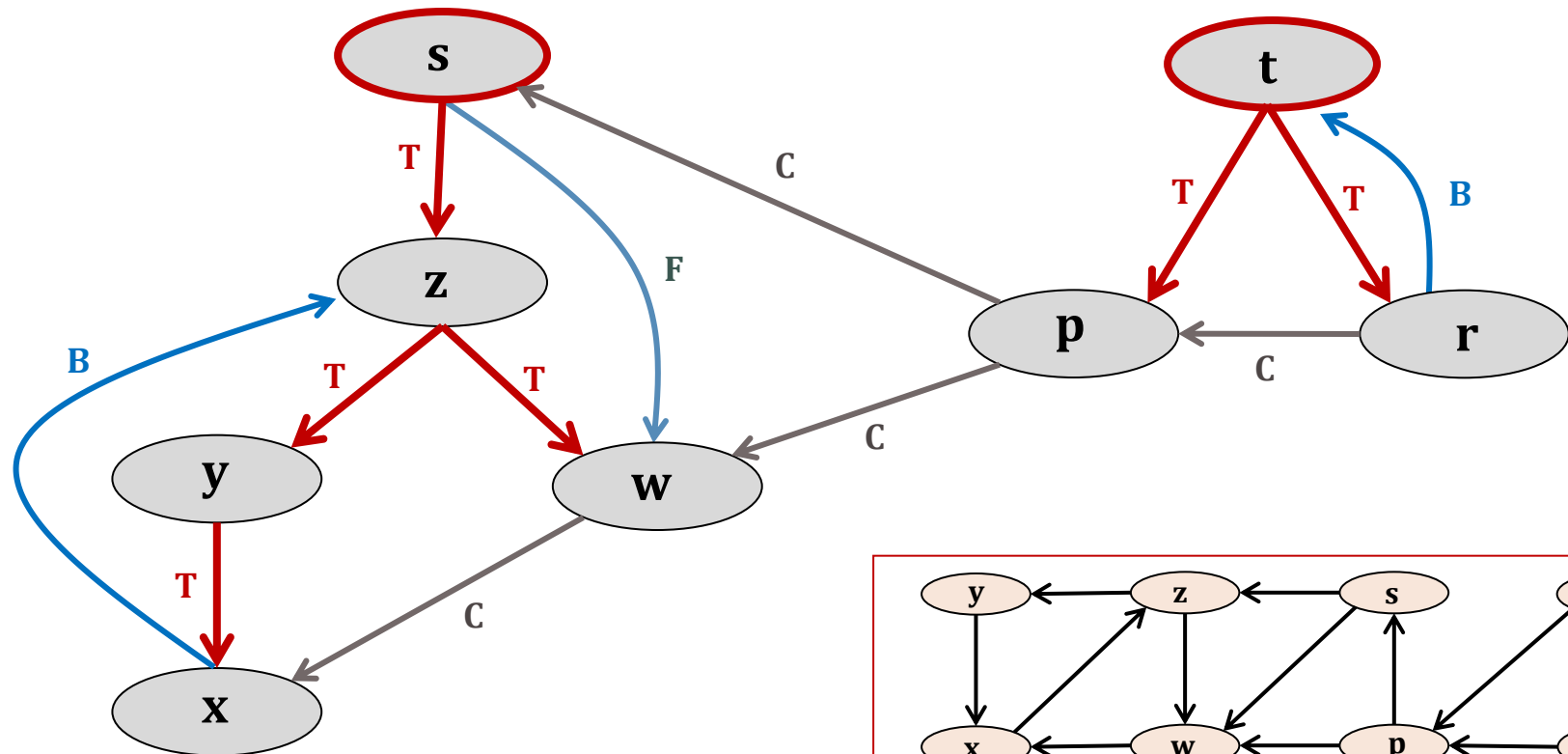
□ DFS-δένδρο (δάσος)



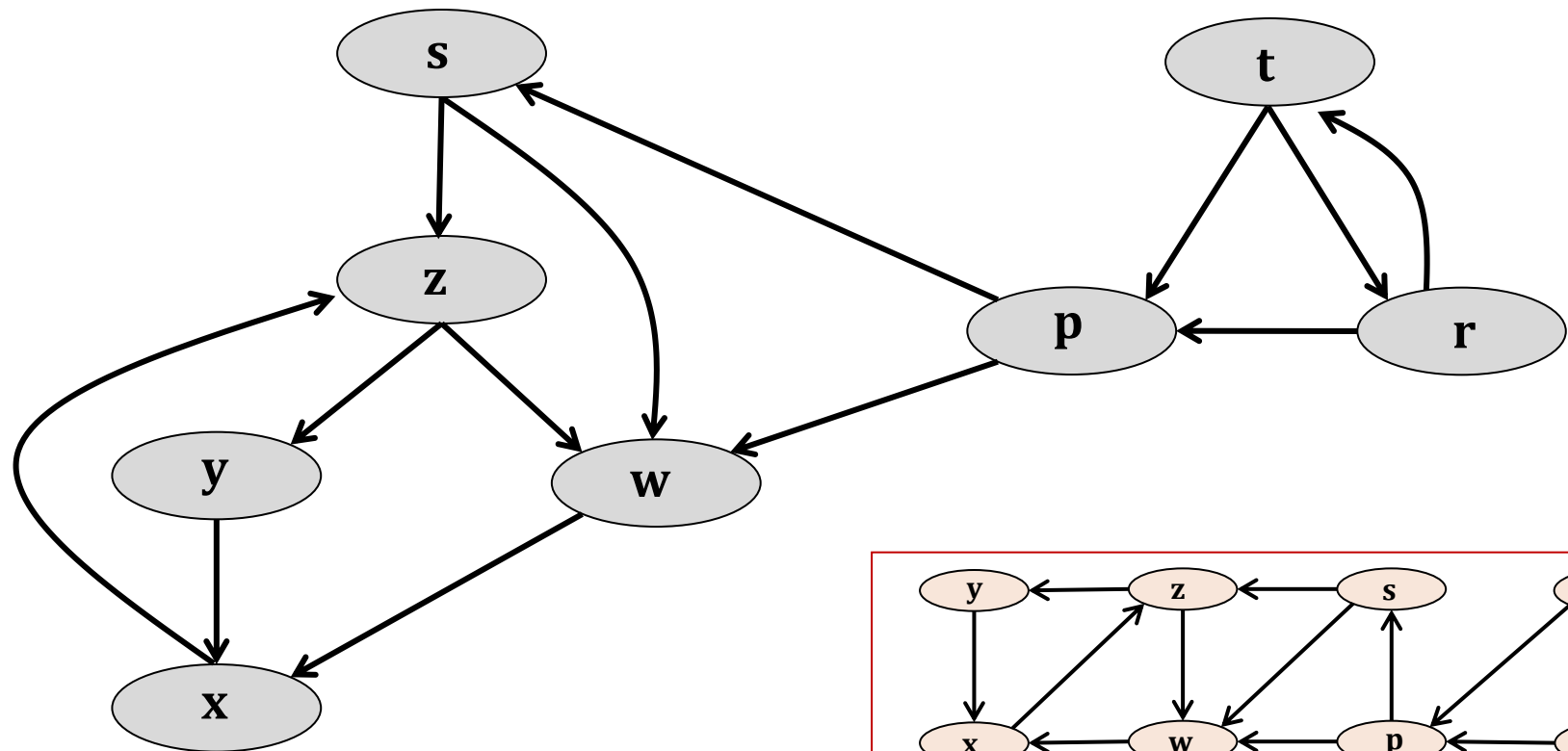
□ DFS-δένδρο (δάσος)



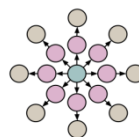
□ DFS-αναπαράσταση



□ DFS-αναπαράσταση

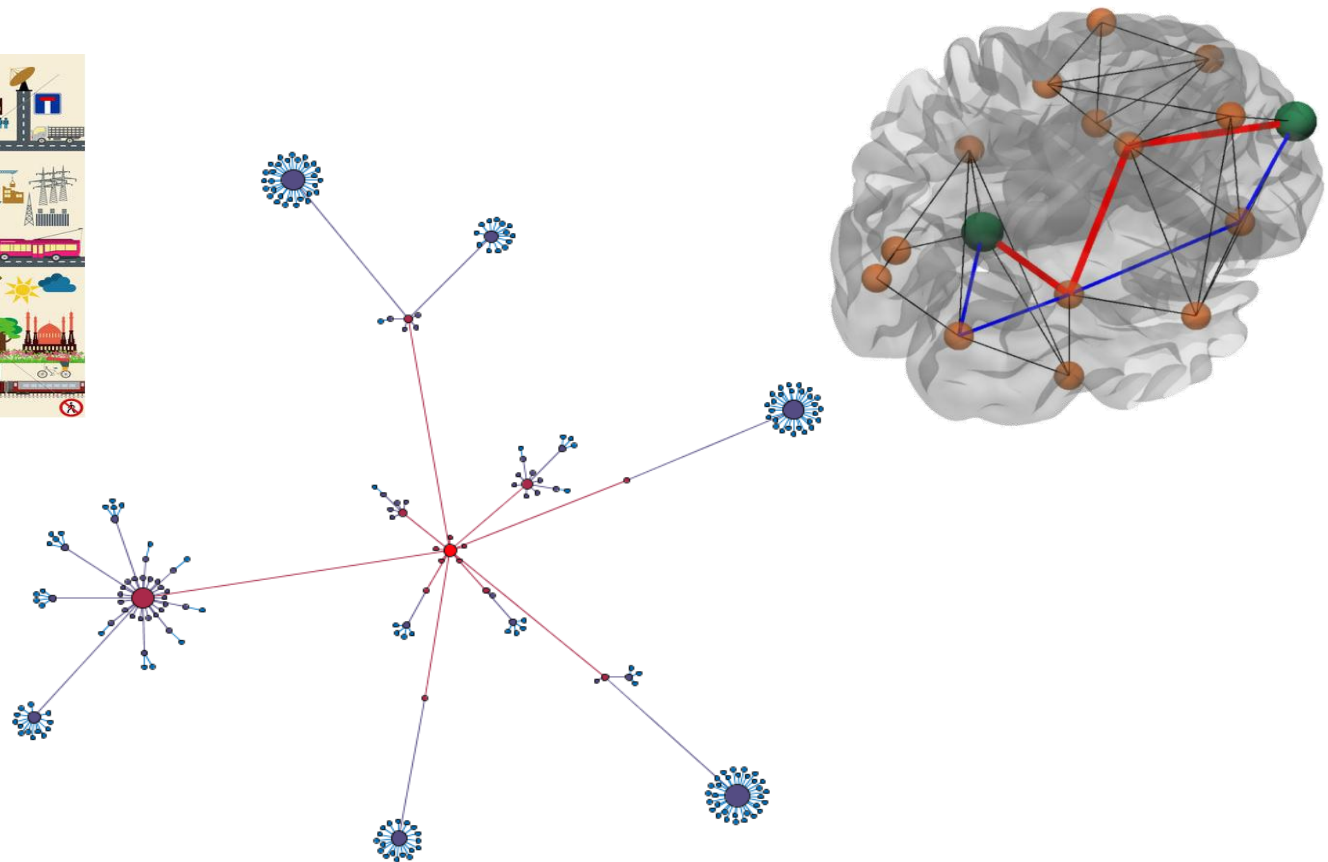


BFS vs DFS

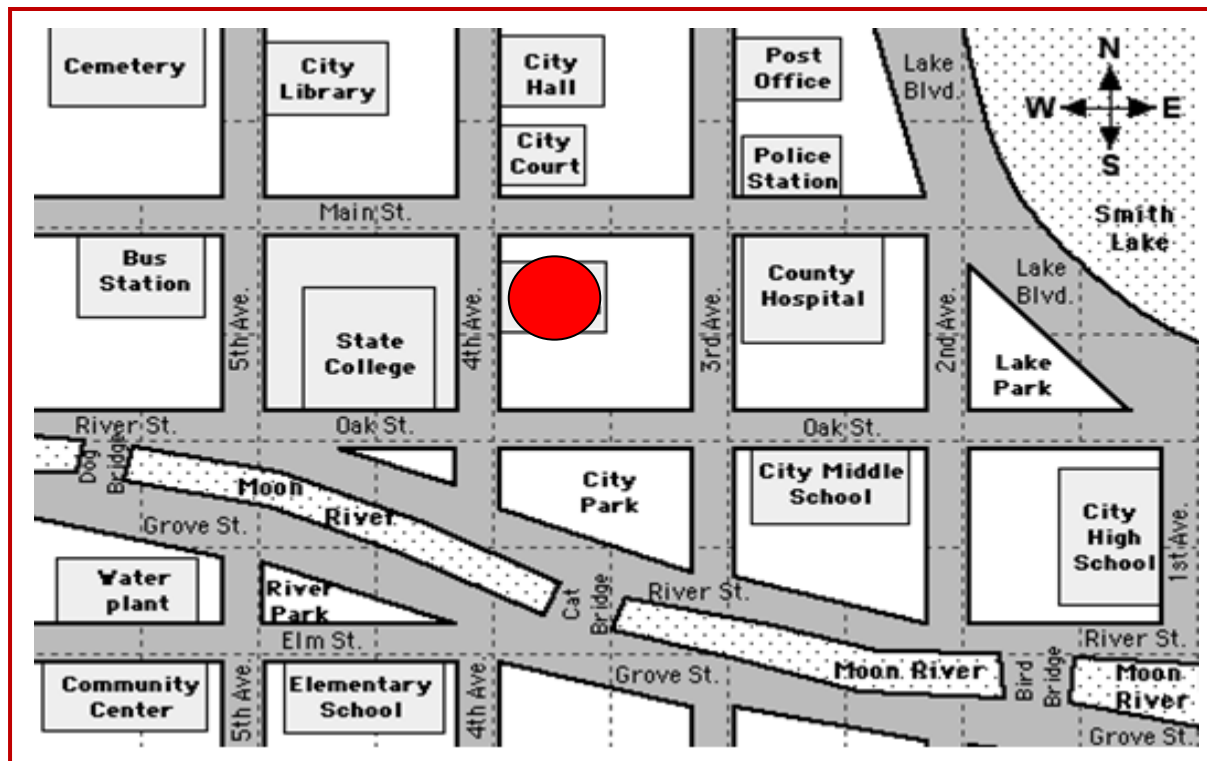


	BFS	DFS
Data structure	Queue	Stack
No. of vertex orderings	1 ordering	2 orderings
Edge types	Tree edge Cross edge	Tree edge Back edge
Applications	Connectivity Acyclicity Minimum edge paths	Connectivity Acyclicity Articulation points
Efficiency for adjacency matrix	$\Theta(n^2)$	$\Theta(n^2)$
Efficiency for adjacency lists	$\Theta(n + e)$	$\Theta(n + e)$

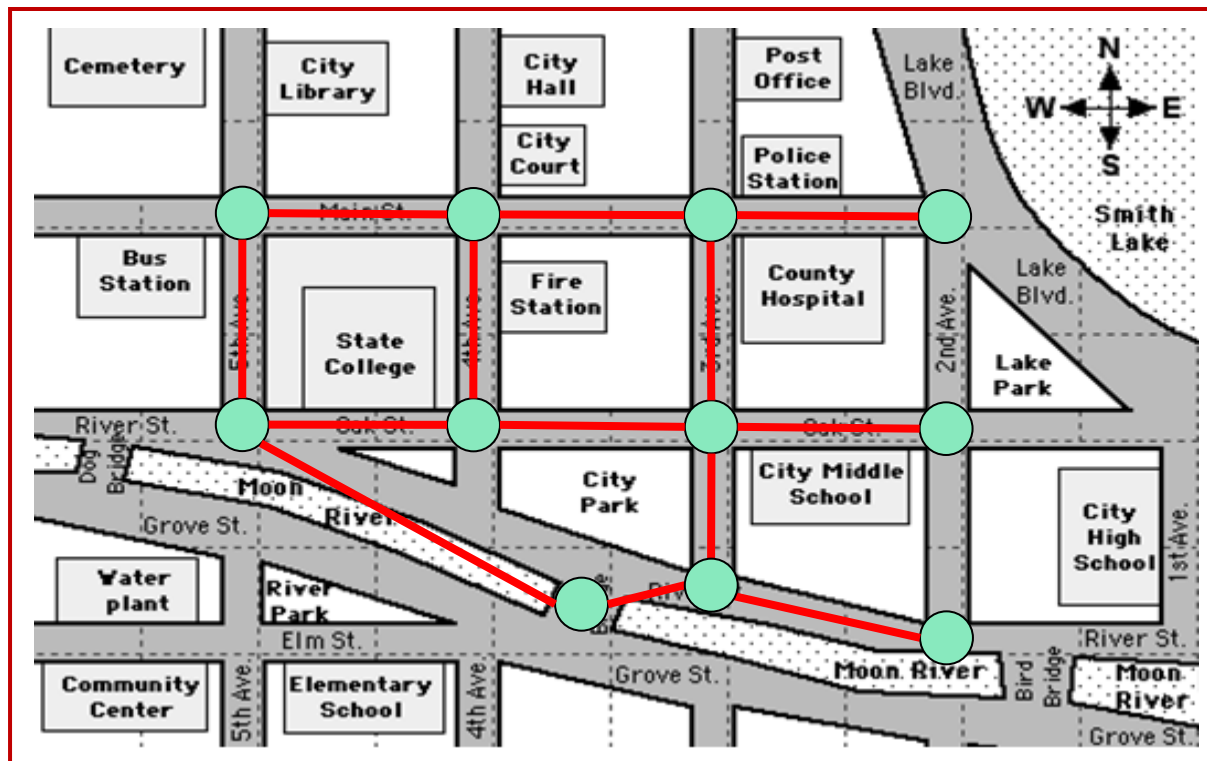
Ε Τοποθέτηση Υπηρεσιών και Λαβύρινθοι



Ε Τοποθέτηση Πυροσβεστικού Τμήματος

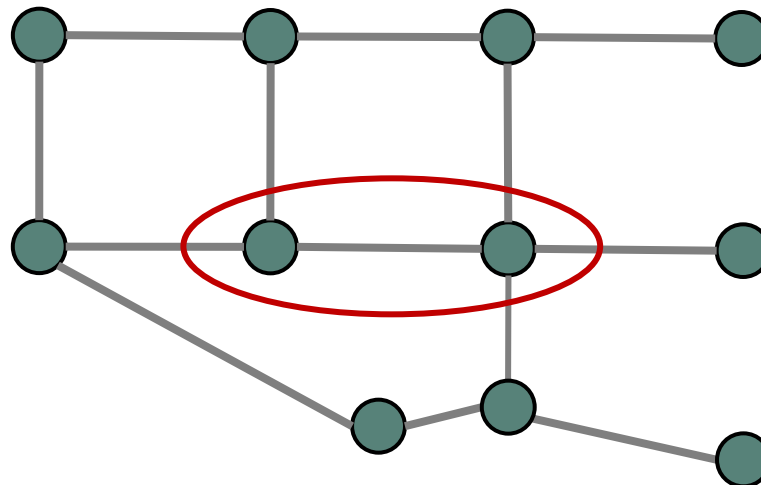


Ε Τοποθέτηση Πυροσβεστικού Τμήματος



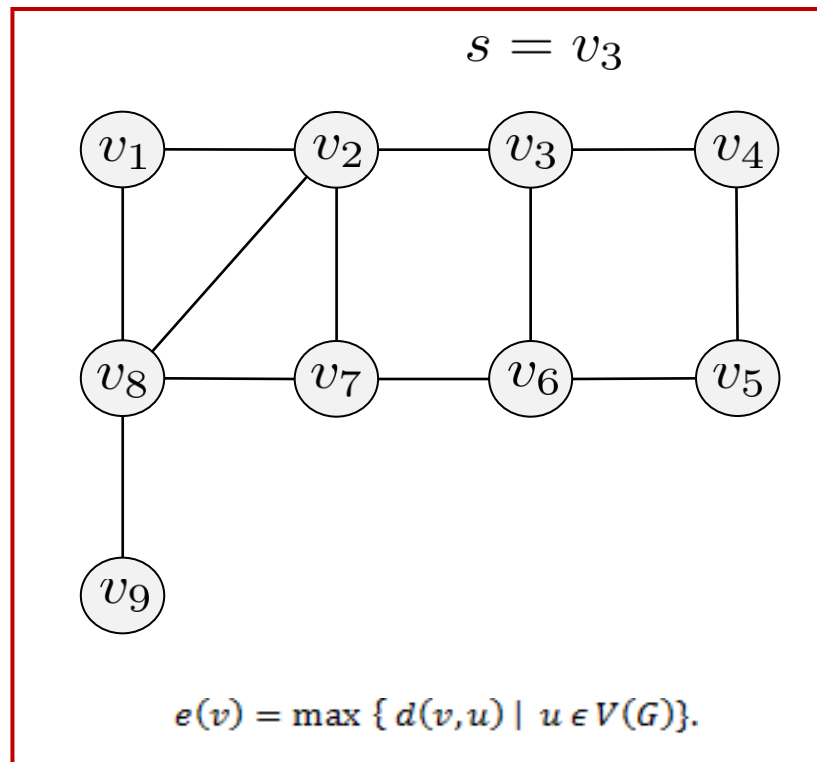
Ε Τοποθέτηση Πυροσβεστικού Τμήματος

Ελάχιστη Εκκεντρικότητα
(min eccentricity)

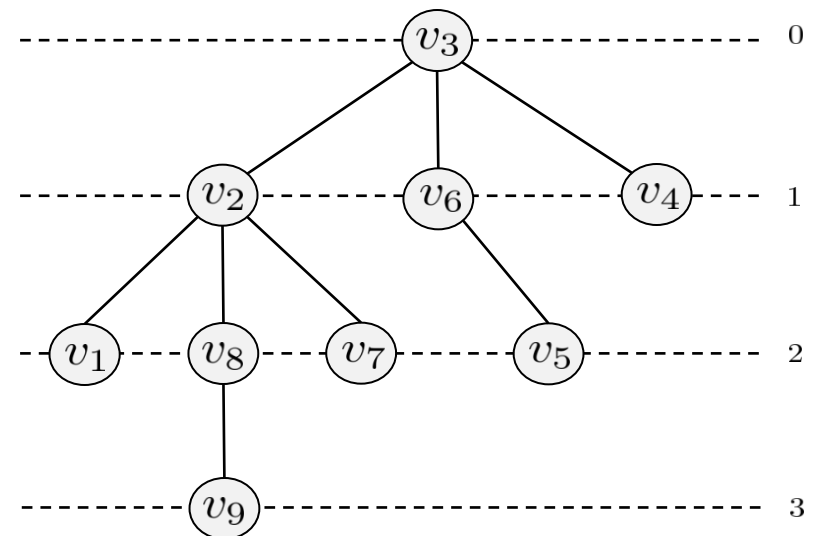


$$e(v) = \max \{ d(v, u) \mid u \in V(G) \}.$$

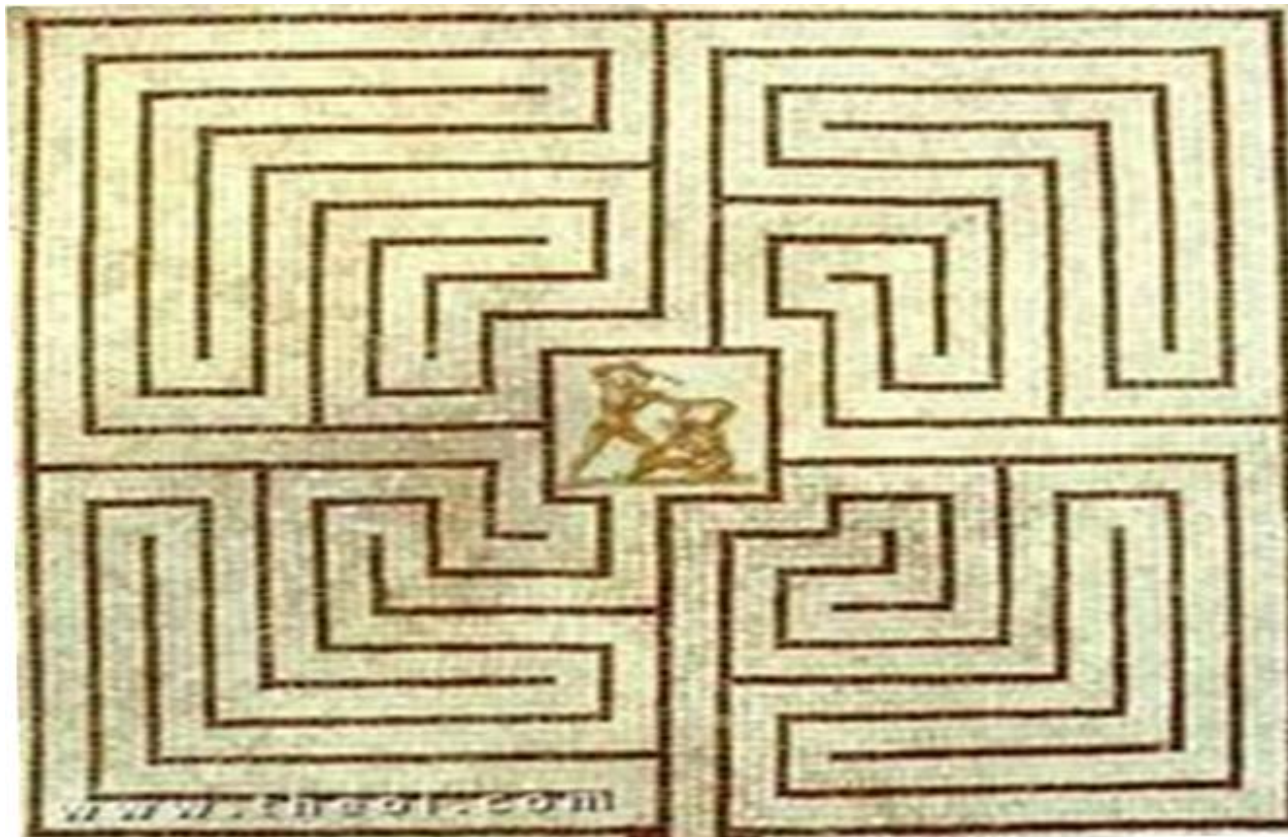
Ε Τοποθέτηση Πυροσβεστικού Τμήματος



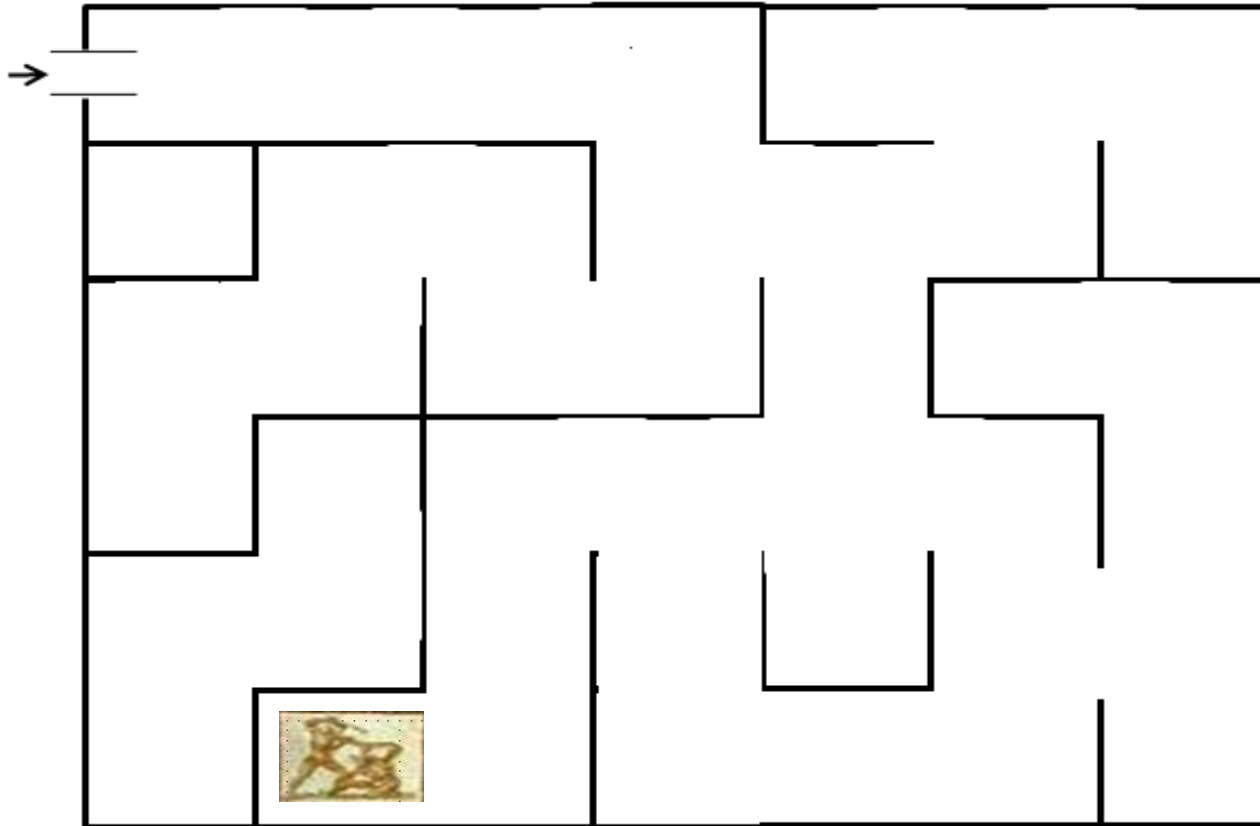
BFS-δένδρο



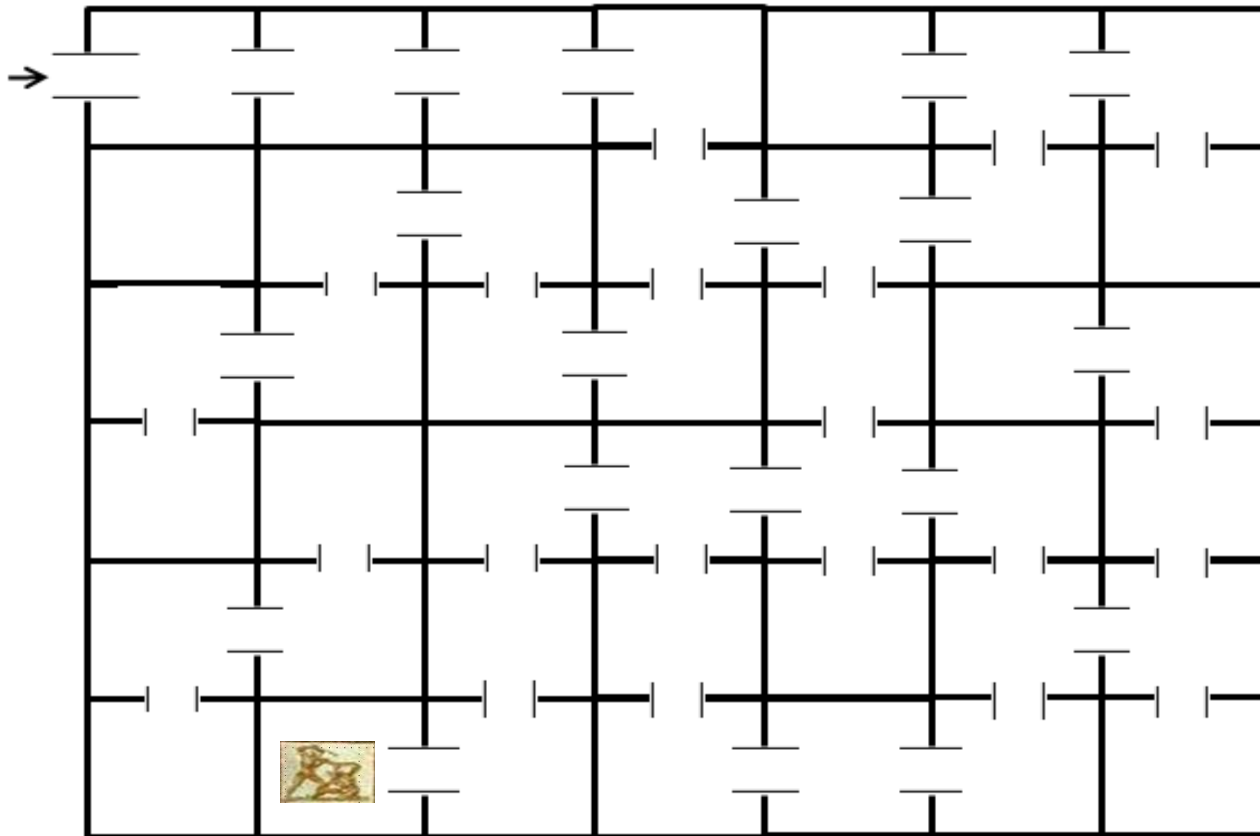
Ε Ξετυλίγοντας τον Μίτο της Αριάδνης !



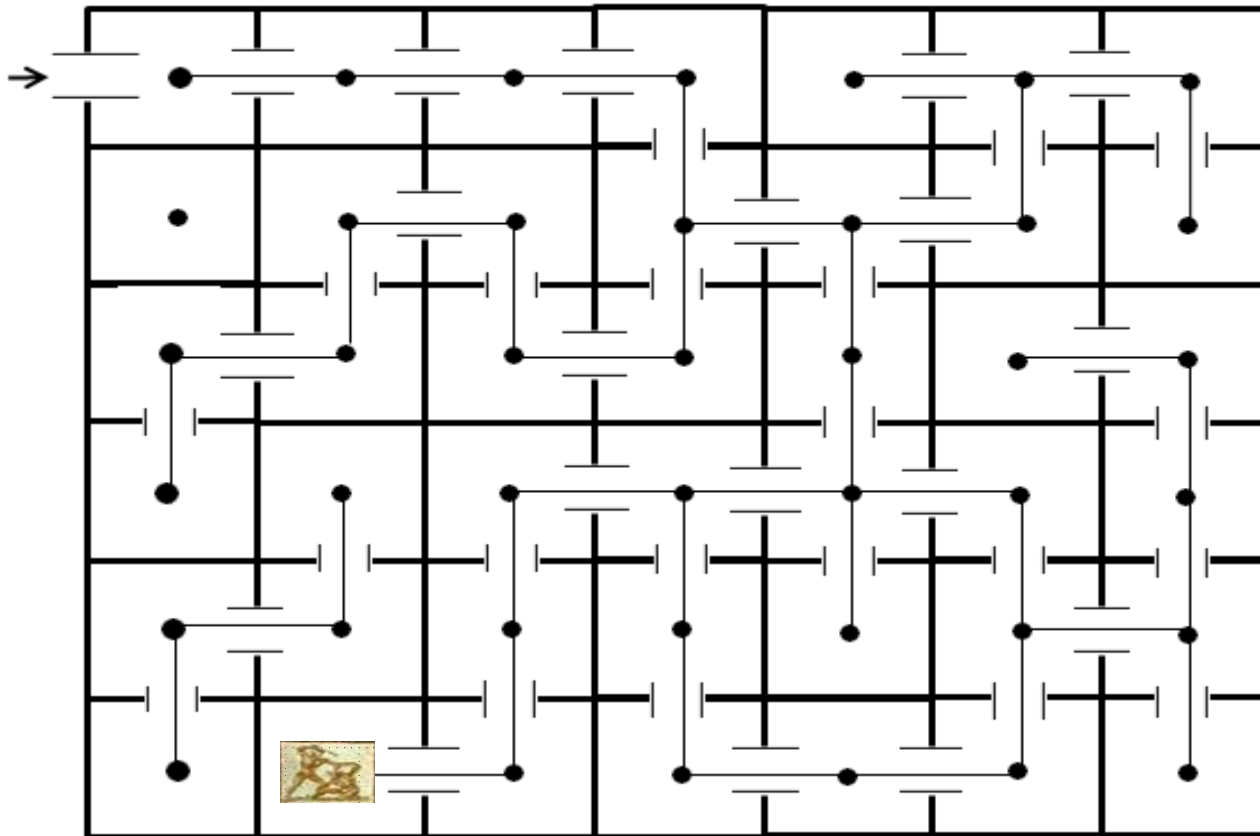
Ε Ξετυλίγοντας τον Μίτο της Αριάδνης !



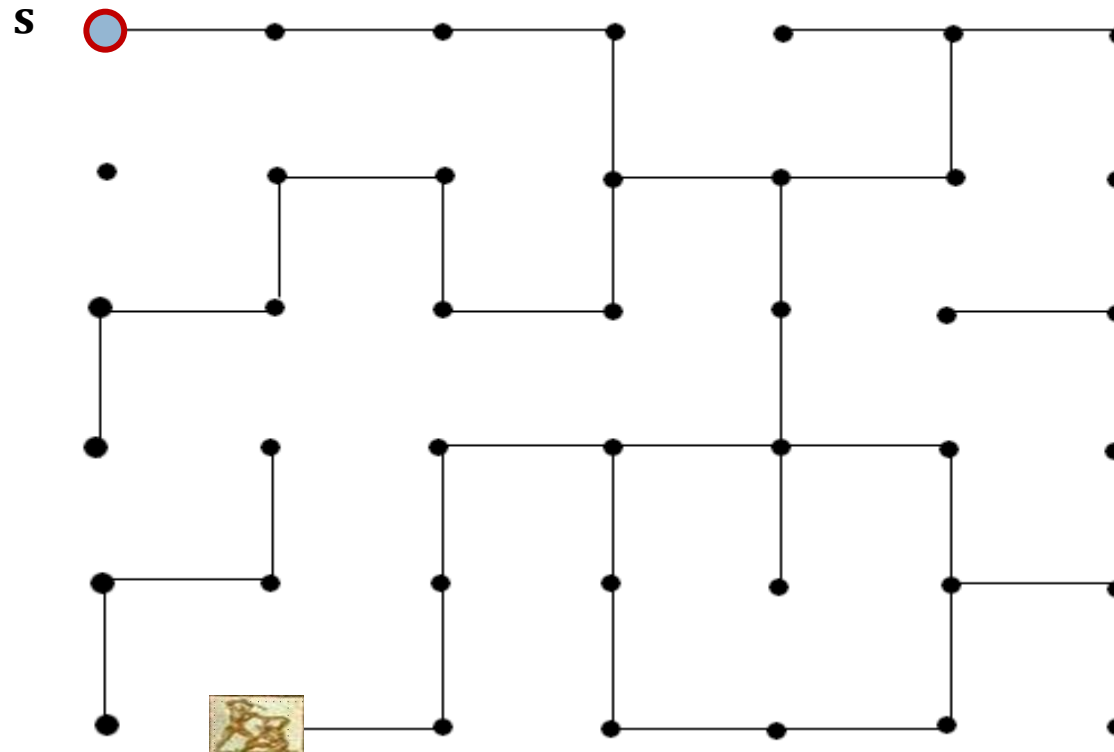
Ε Ξετυλίγοντας τον Μίτο της Αριάδνης !



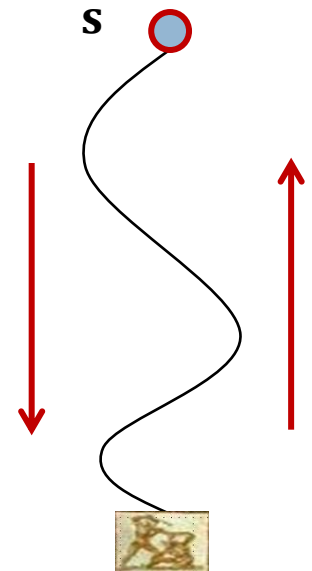
Ε Ξετυλίζοντας τον Μίτο της Αριάδνης !



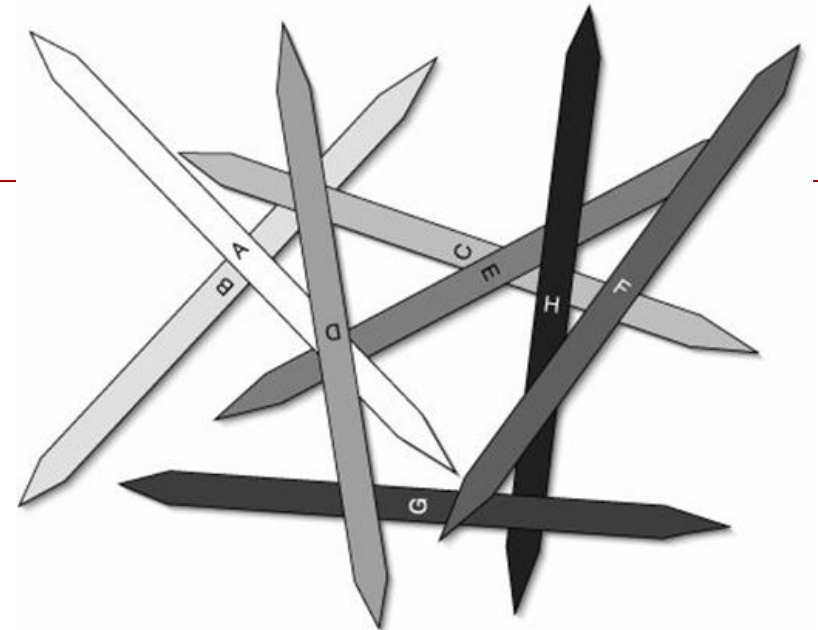
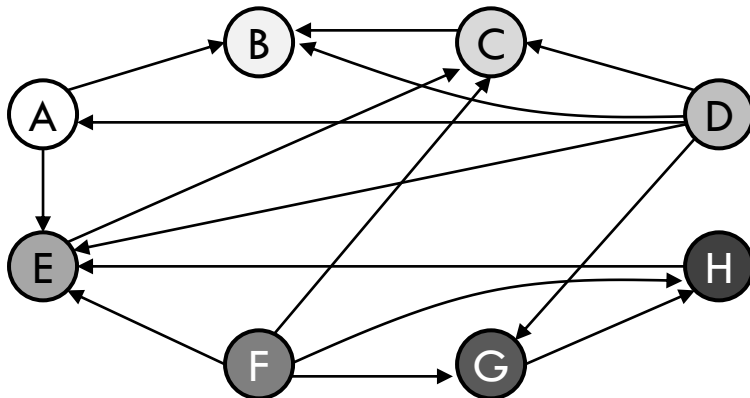
Ξετυλίγοντας τον Μίτο της Αριάδνης !



DFS-διερεύνηση

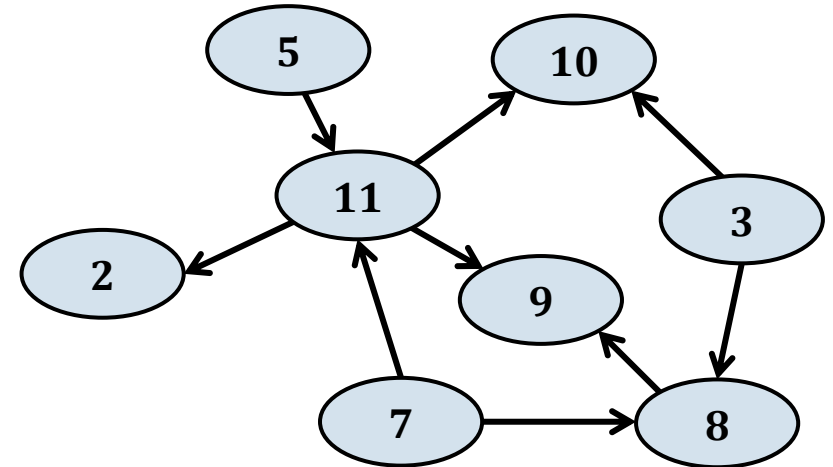


Το Πρόβλημα της Τοπολογικής Ταξινόμησης



Κατευθυνόμενο Άκυκλο Γράφημα (DAG) :

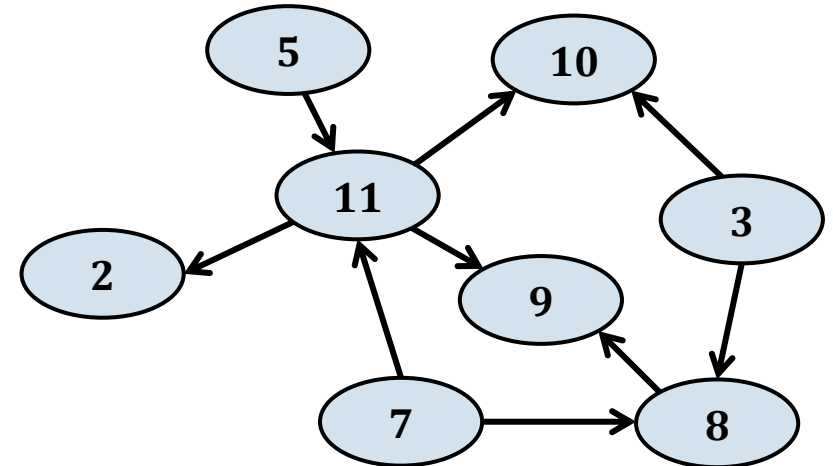
Ένα κατευθυνόμενο γράφημα $G = (V, E)$ χωρίς κατευθυνόμενους κύκλους.



Κατευθυνόμενο Άκυκλο Γράφημα (DAG) :

Ένα κατευθυνόμενο γράφημα $G = (V, E)$ χωρίς κατευθυνόμενους κύκλους.

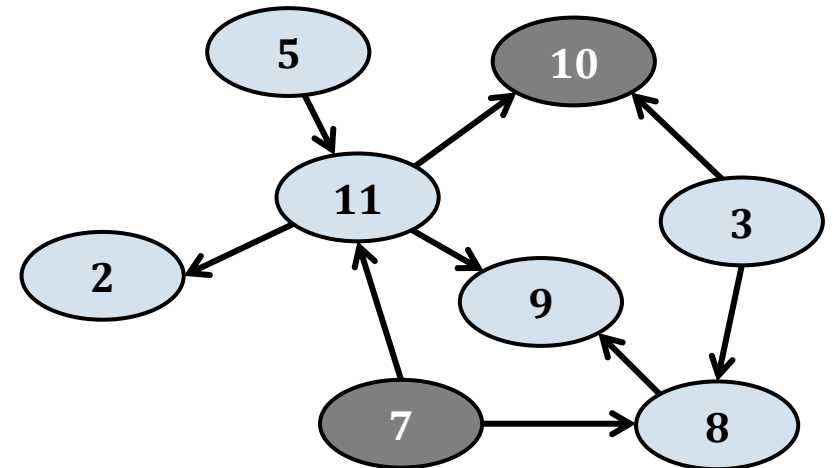
Το γράφημα DAG έχει ένα τουλάχιστον κόμβο με **in-degree = 0** και ένα τουλάχιστον με **out-degree = 0** !!!



Κατευθυνόμενο Άκυκλο Γράφημα (DAG) :

Ένα κατευθυνόμενο γράφημα $G = (V, E)$ χωρίς κατευθυνόμενους κύκλους.

Το γράφημα DAG έχει ένα τουλάχιστον κόμβο με **in-degree = 0** και ένα τουλάχιστον με **out-degree = 0** !!!

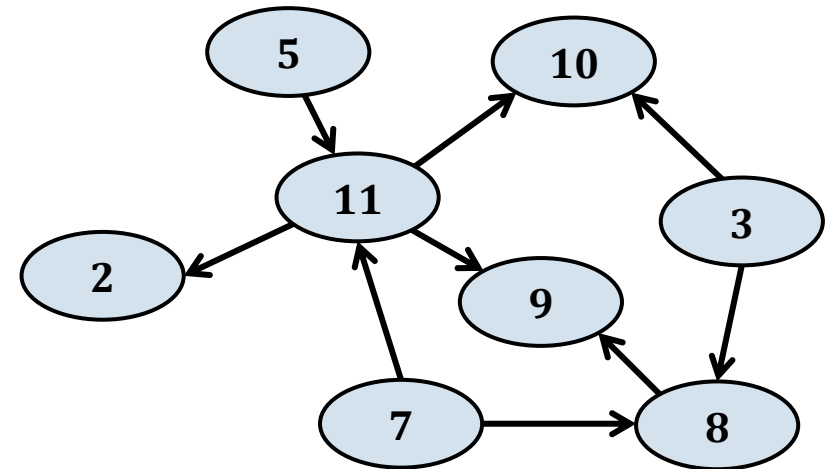


Κατευθυνόμενο Άκυκλο Γράφημα (DAG) :

Ένα κατευθυνόμενο γράφημα $G = (V, E)$ χωρίς κατευθυνόμενους κύκλους.

Το γράφημα DAG έχει ένα τουλάχιστον κόμβο με **in-degree = 0** και ένα τουλάχιστον με **out-degree = 0** !!!

Κάθε υπογράφημα $G' = (V', E')$ ενός DAG είναι κατευθυνόμενο άκυκλο γράφημα !!!



Οι κόμβοι ενός κατευθυνόμενου άκυκλου γραφήματος G μπορούν να «τοποθετηθούν» σε μια σειρά, η οποία ονομάζεται **Τοπολογική Ταξινόμηση** των κόμβων του G , και η οποία έχει μια χαρακτηριστική και χρήσιμη ιδιότητα !!!

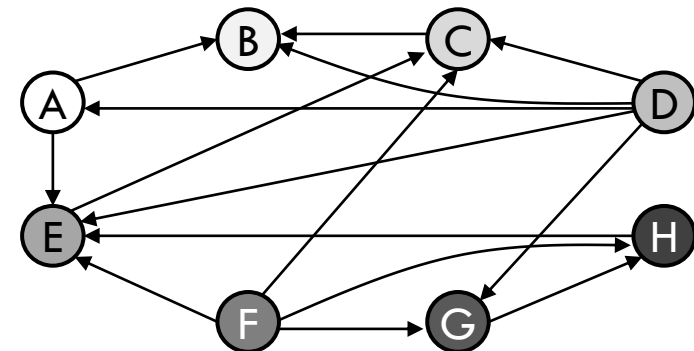
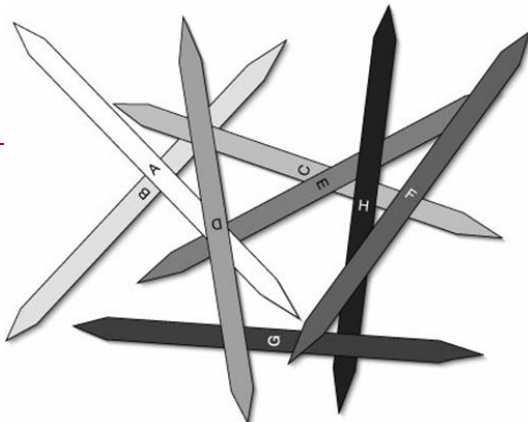


Τοπολογική Ταξινόμηση

■ Τοπολογική Ταξινόμηση ενός DAG :

Μια διάταξη $TopS = (u_1, u_2, \dots, u_n)$ των κόμβων ενός DAG G τέτοια ώστε :

$\forall \langle u_i, u_j \rangle \in E(G) \Rightarrow$ ο κόμβος u_i προηγείται του κόμβου u_j στην $TopS$.

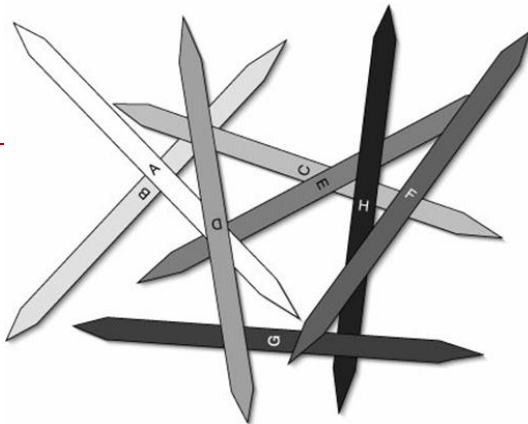


Τοπολογική Ταξινόμηση

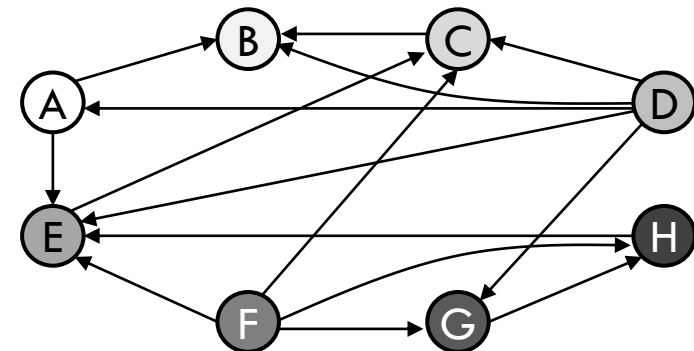
■ Τοπολογική Ταξινόμηση ενός DAG :

Μια διάταξη $TopS = (u_1, u_2, \dots, u_n)$ των κόμβων ενός DAG G τέτοια ώστε :

$\forall \langle u_i, u_j \rangle \in E(G) \Rightarrow$ ο κόμβος u_i προηγείται του κόμβου u_j στην $TopS$.



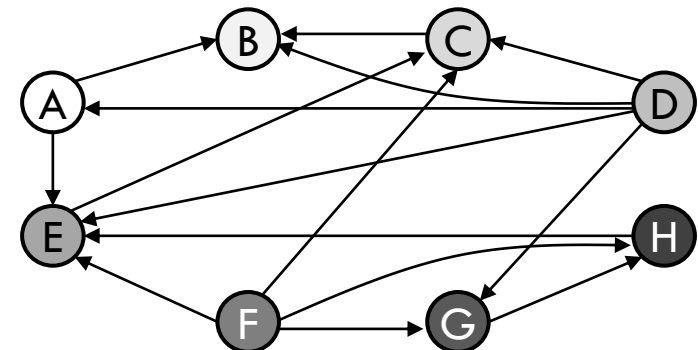
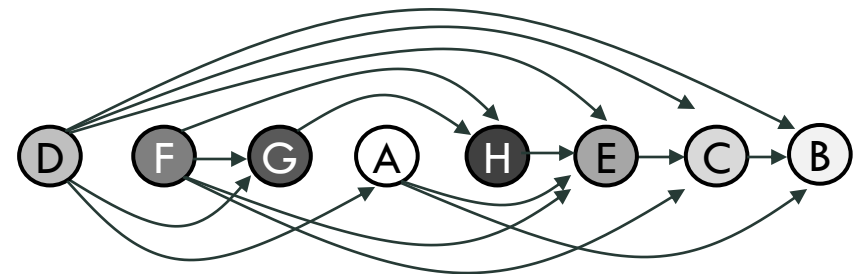
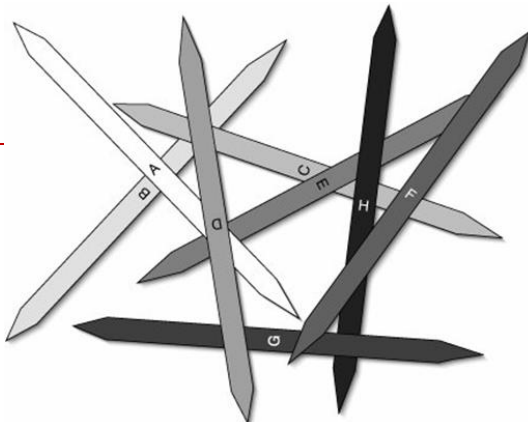
(D) (F) (G) (A) (H) (E) (C) (B)



Τοπολογική Ταξινόμηση

■ Τοπολογική Ταξινόμηση ενός DAG :

Μια διάταξη $TopS = (u_1, u_2, \dots, u_n)$ των κόμβων ενός DAG G τέτοια ώστε :
 $\forall \langle u_i, u_j \rangle \in E(G) \Rightarrow$ ο κόμβος u_i προηγείται του κόμβου u_j στην $TopS$.



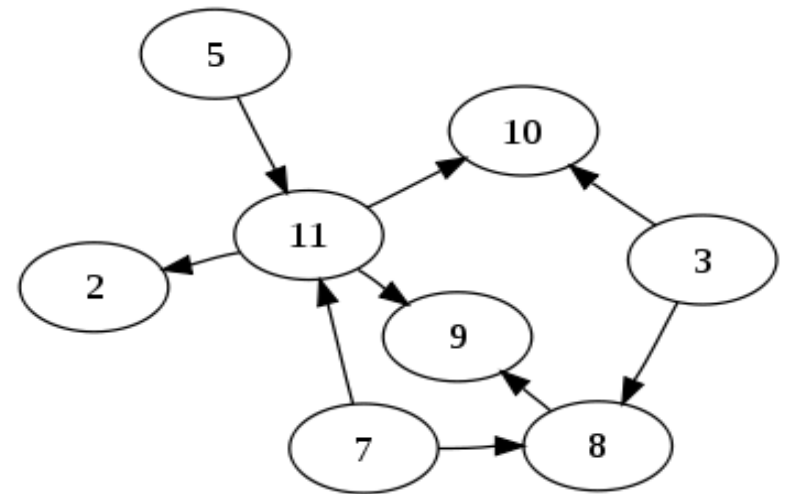
Τοπολογική Ταξινόμηση

Τοπολογική Ταξινόμηση ενός DAG :

Μια διάταξη $TopS = (u_1, u_2, \dots, u_n)$ των κόμβων ενός DAG G τέτοια ώστε :

$\forall \langle u_i, u_j \rangle \in E(G) \Rightarrow$ ο κόμβος u_i προηγείται του κόμβου u_j στην $TopS$.

Η TS ενός DAG είναι μοναδική ?

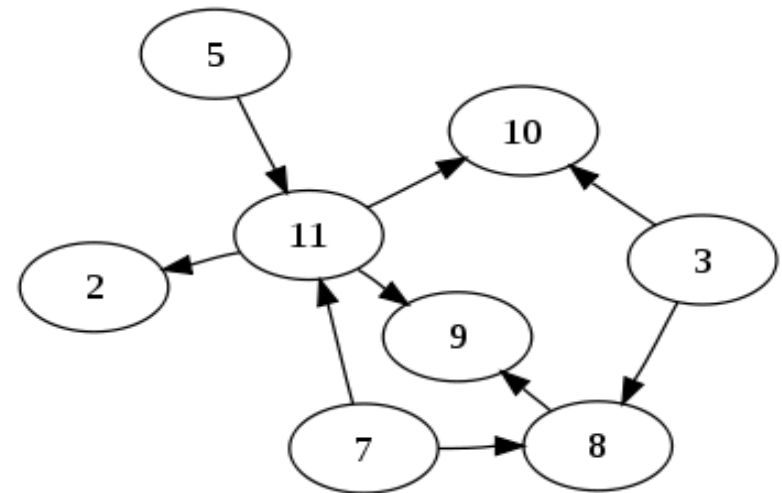
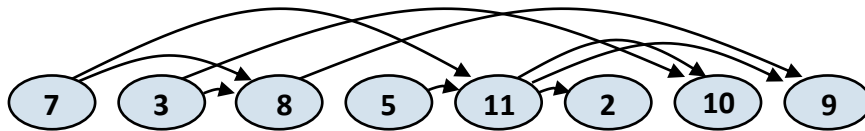


Τοπολογική Ταξινόμηση

Τοπολογική Ταξινόμηση ενός DAG :

Μια διάταξη $TopS = (u_1, u_2, \dots, u_n)$ των κόμβων ενός DAG G τέτοια ώστε :
 $\forall \langle u_i, u_j \rangle \in E(G) \Rightarrow$ ο κόμβος u_i προηγείται του κόμβου u_j στην $TopS$.

Η TS ενός DAG είναι μοναδική ?



Γενικά, η Τοπολογική Ταξινόμηση δεν είναι μοναδική :

$TS_1 = (3, 7, 8, 9, 5, 11, 2, 10)$, $TS_2 = (7, 5, 3, 8, 11, 9, 10, 2)$, ..., $TS_k = (5, 3, 7, 11, 8, 10, 2, 9)$

Τοπολογική Ταξινόμηση

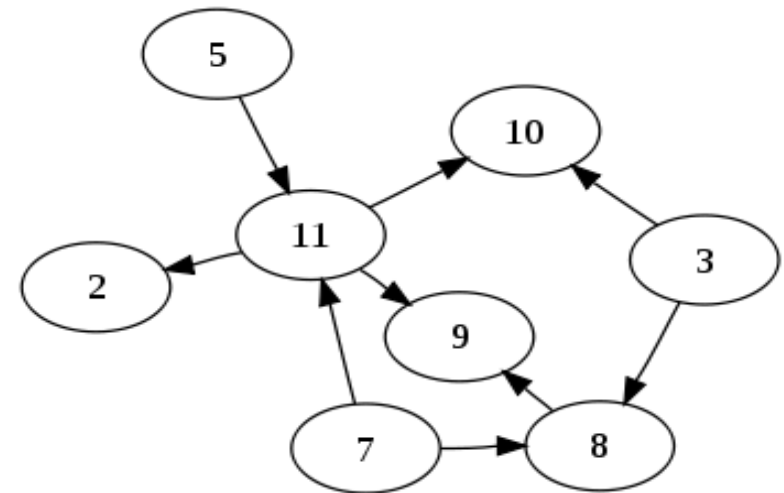
■ Τοπολογική Ταξινόμηση ενός DAG :

Μια διάταξη $TopS = (u_1, u_2, \dots, u_n)$ των κόμβων ενός DAG G τέτοια ώστε :

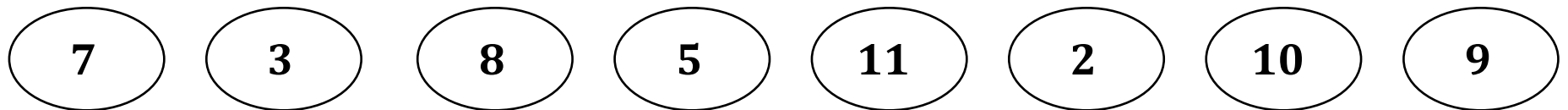
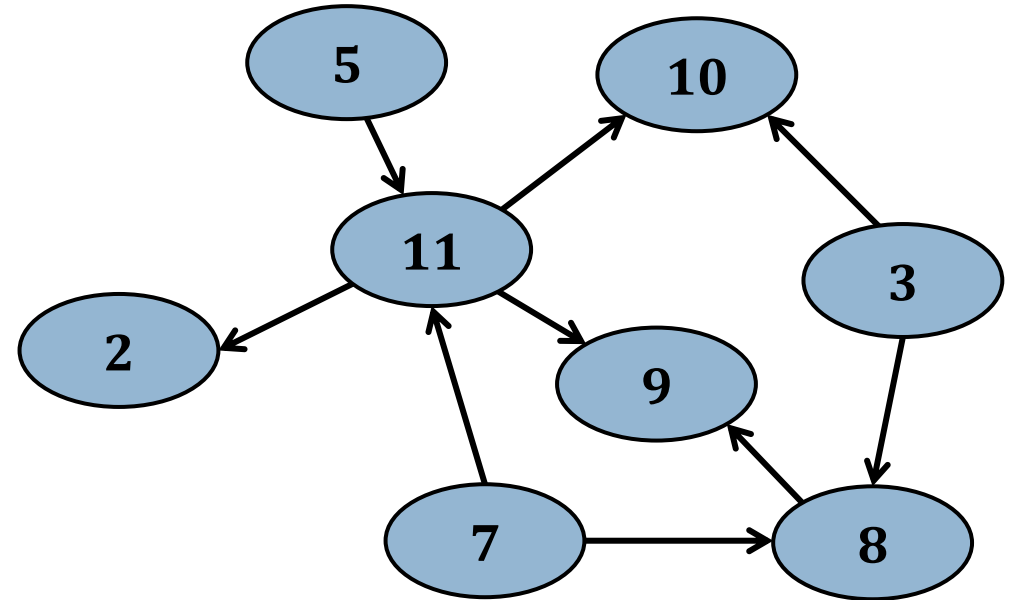
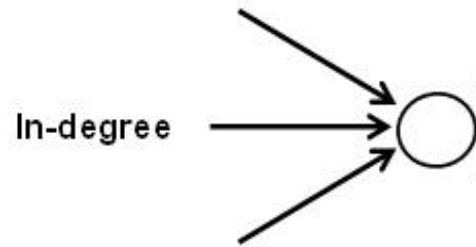
$\forall \langle u_i, u_j \rangle \in E(G) \Rightarrow$ ο κόμβος u_i προηγείται του κόμβου u_j στην $TopS$.

Υπολογισμός διάταξης TS

- ✓ Μέθοδος In-degree
- ✓ Μέθοδος Out-degree
- ✓ Μέθοδος DFS

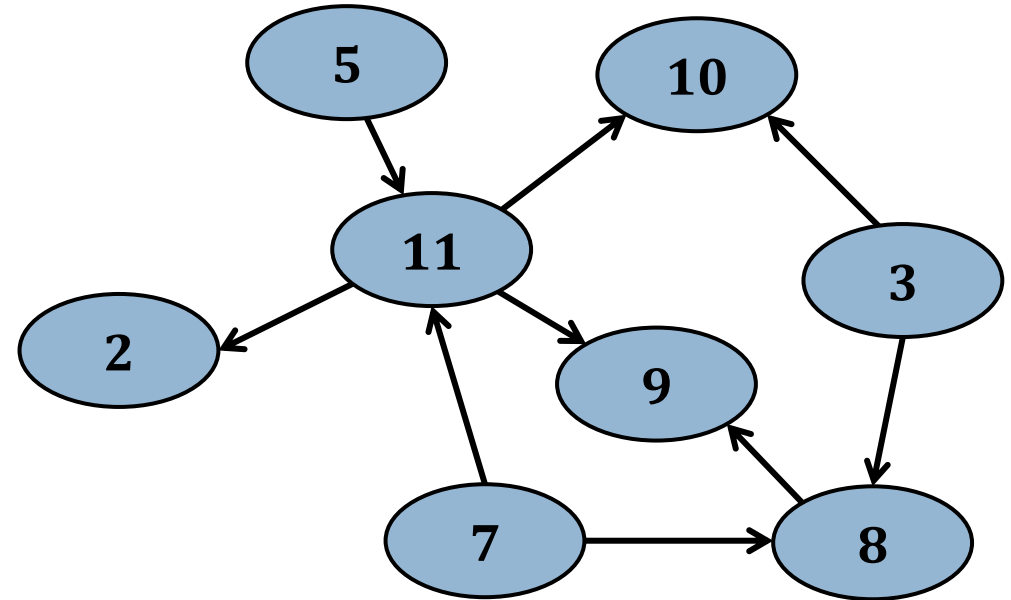


Μέθοδος In-degree



Αλγόριθμος **TS-Indeg**(G)

Υπολογισμός μιας TS του $G=(V,E)$

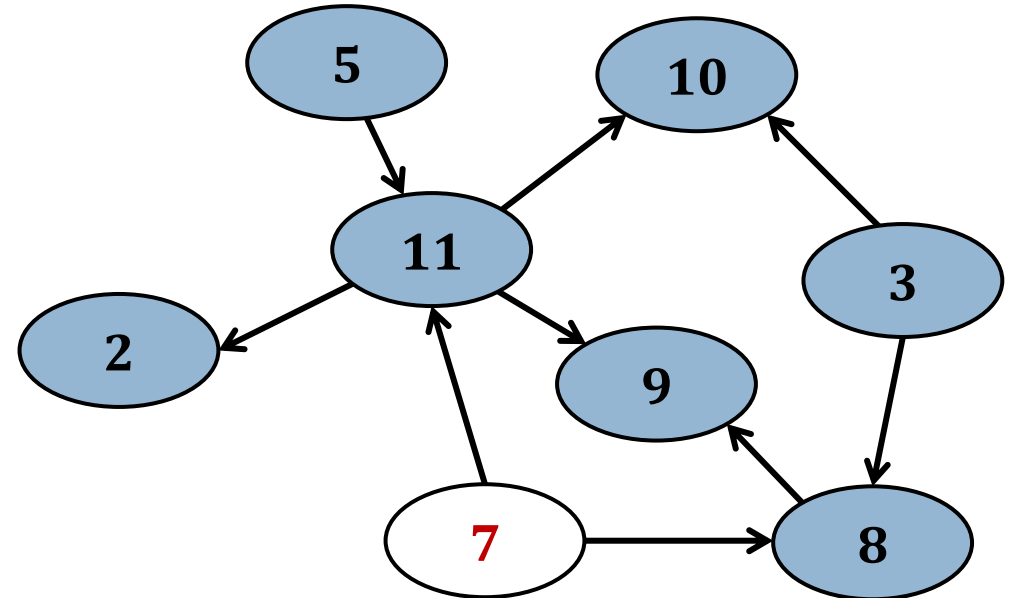


Αλγόριθμος **TS-Indeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{In-degree} = 0$

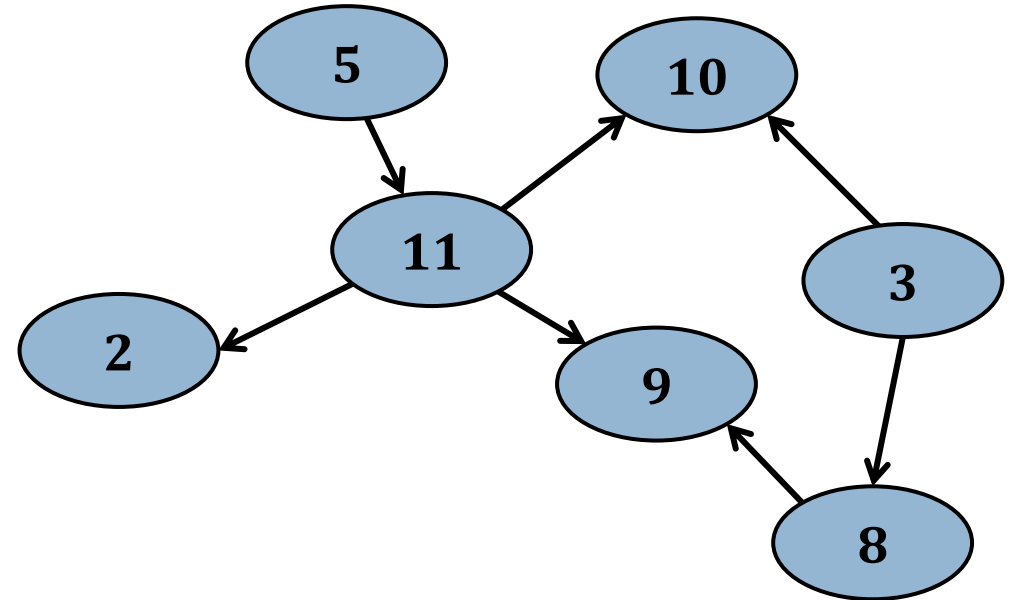
Κόμβος **7**



Αλγόριθμος **TS-Indeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου **7**
από το γράφημα, και
Εισαγωγή του στο τέλος
μια λίστας



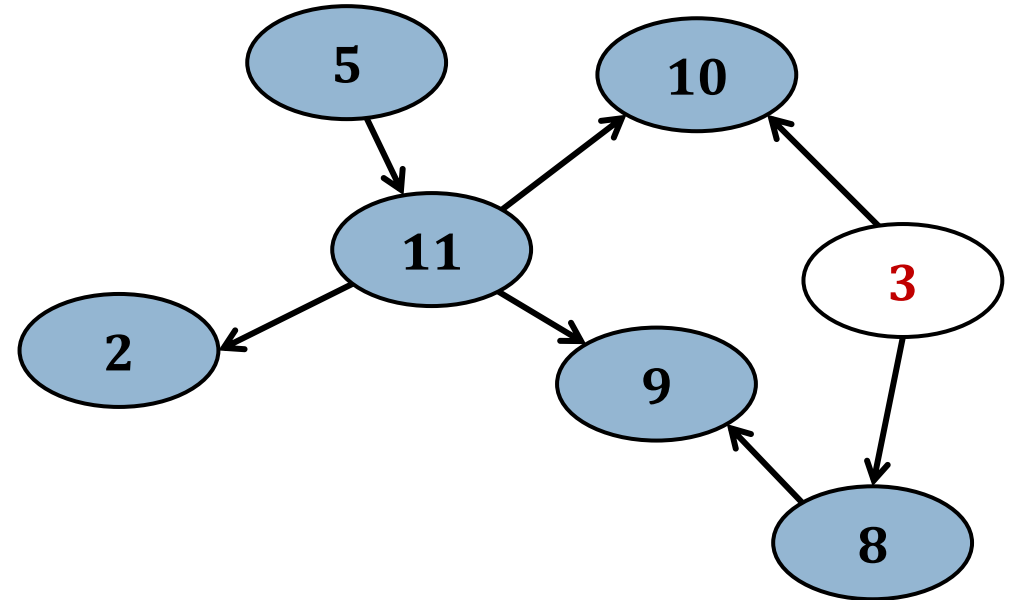
7

Αλγόριθμος **TS-Indeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{In-degree} = 0$

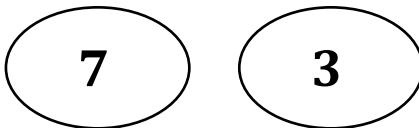
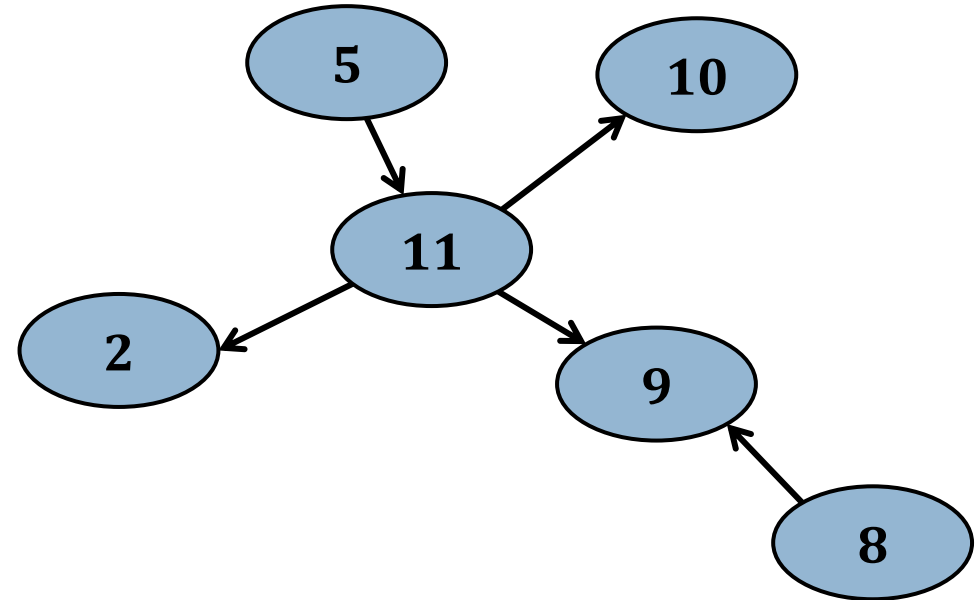
Κόμβος **3**



Αλγόριθμος **TS-Indeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου **3**
από το γράφημα, και
Εισαγωγή του στο τέλος
μια λίστας

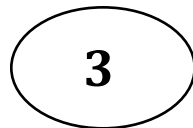
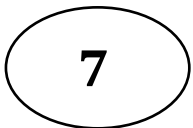
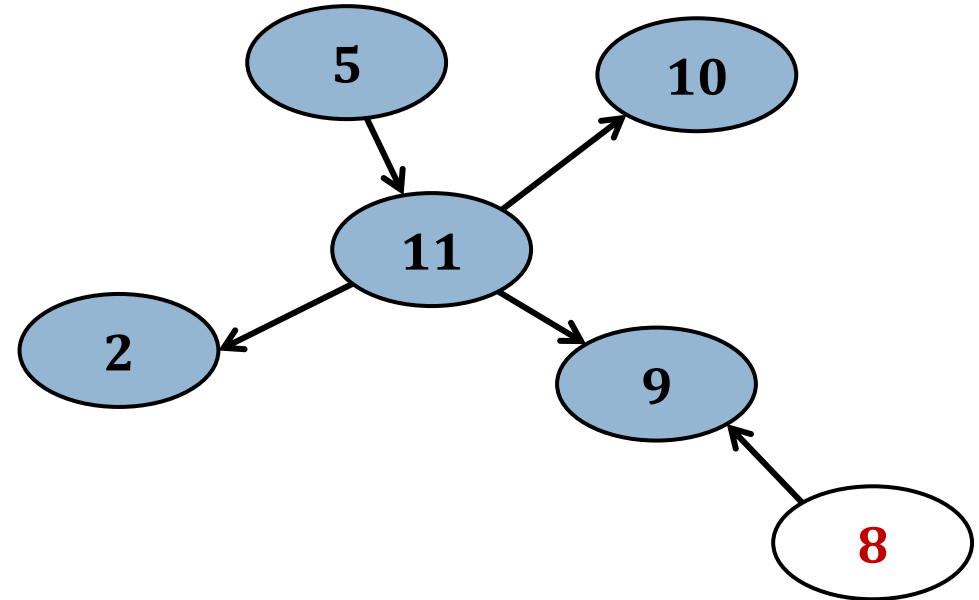


Αλγόριθμος **TS-Indeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{In-degree} = 0$

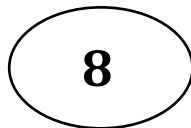
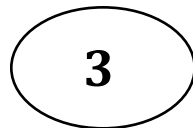
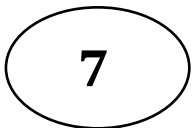
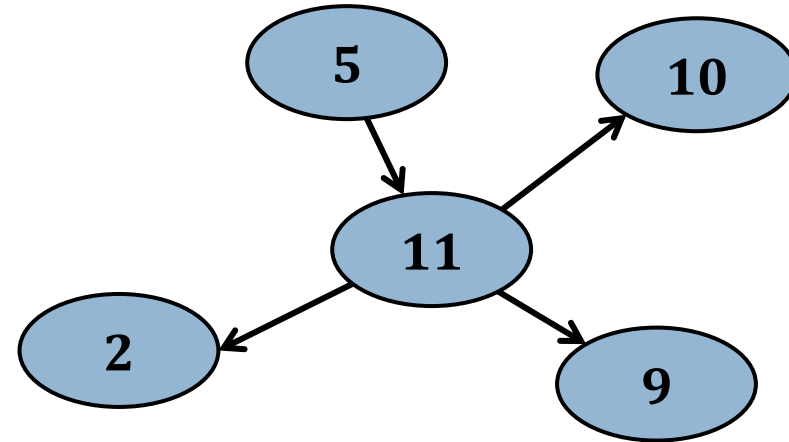
Κόμβος **8**



Αλγόριθμος **TS-Indeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου **8**
από το γράφημα, και
Εισαγωγή του στο τέλος
μια λίστας

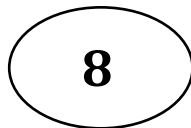
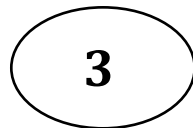
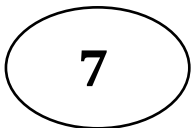
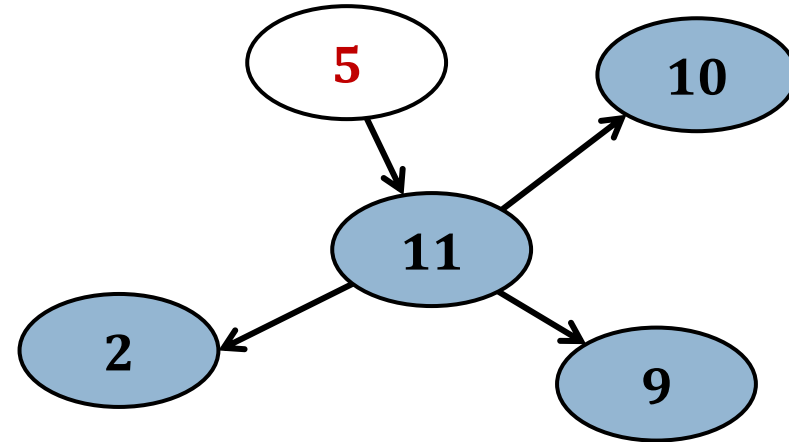


Αλγόριθμος **TS-Indeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{In-degree} = 0$

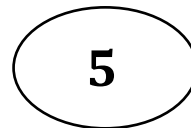
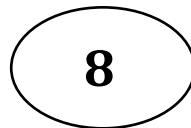
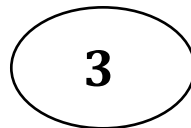
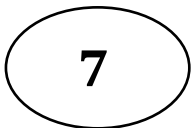
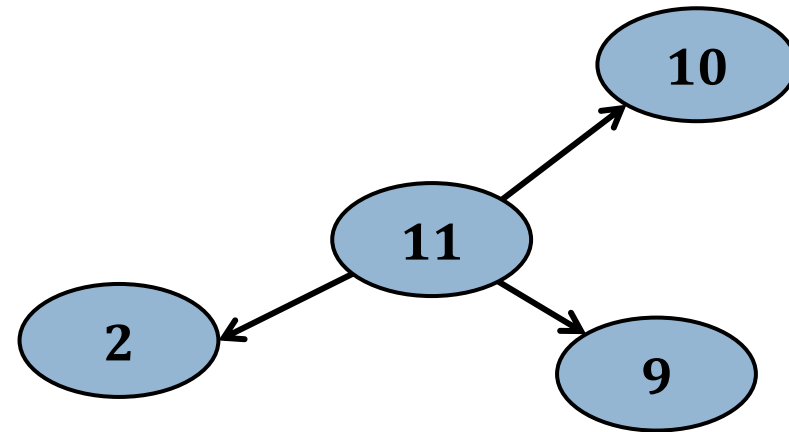
Κόμβος **5**



Αλγόριθμος **TS-Indeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου **5**
από το γράφημα, και
Εισαγωγή του στο τέλος
μια λίστας

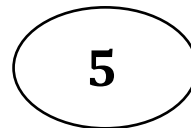
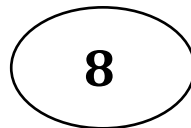
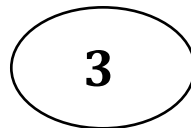
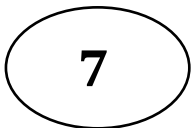
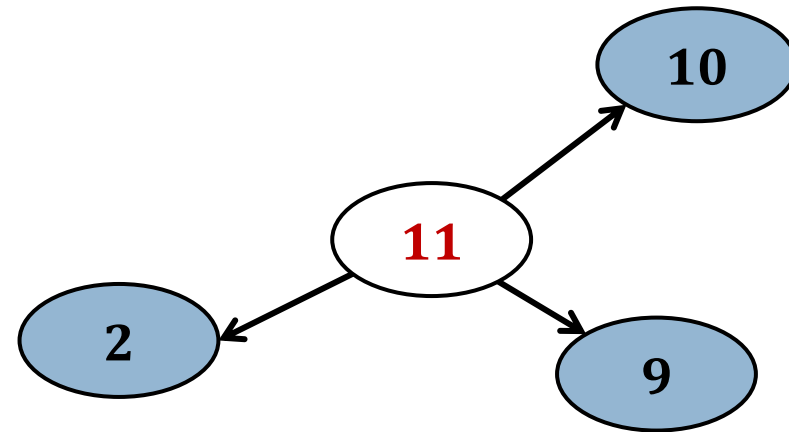


Αλγόριθμος **TS-Indeg**(G)

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{In-degree} = 0$

Κόμβος **11**



Αλγόριθμος **TS-Indeg**(G)

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου **11**
από το γράφημα, και
Εισαγωγή του στο τέλος
μια λίστας

10

2

9

7

3

8

5

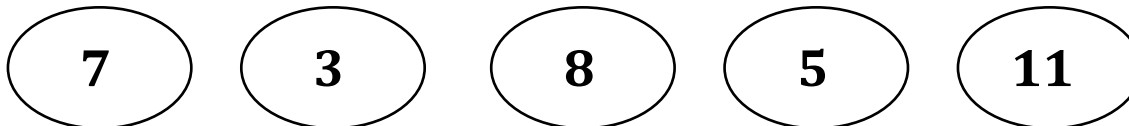
11

Αλγόριθμος **TS-Indeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{In-degree} = 0$

Κόμβος **2**



Αλγόριθμος **TS-Indeg**(G)

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου 2
από το γράφημα, και
Εισαγωγή του στο τέλος
μια λίστας

10

9

7

3

8

5

11

2

Αλγόριθμος **TS-Indeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{In-degree} = 0$

Κόμβος **10**

10

9

7

3

8

5

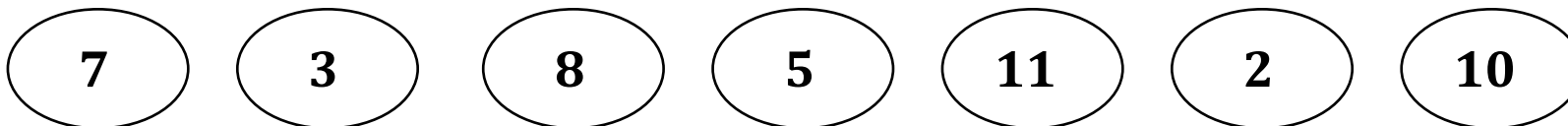
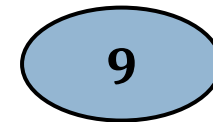
11

2

Αλγόριθμος **TS-Indeg**(G)

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου **10**
από το γράφημα, και
Εισαγωγή του στο τέλος
μια λίστας

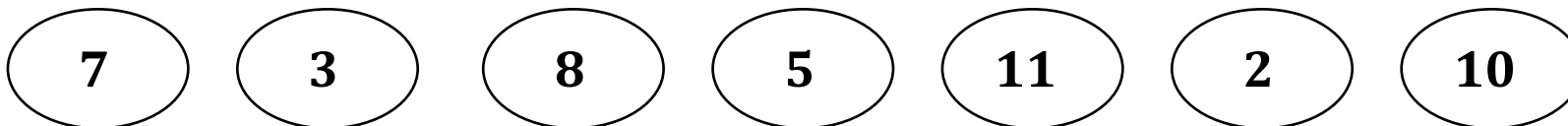
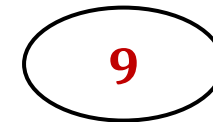


Αλγόριθμος **TS-Indeg**(G)

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{In-degree} = 0$

Κόμβος **9**



Αλγόριθμος **TS-Indeg**(G)

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου **9**
από το γράφημα, και

Εισαγωγή του στο τέλος
μια λίστας

7

3

8

5

11

2

10

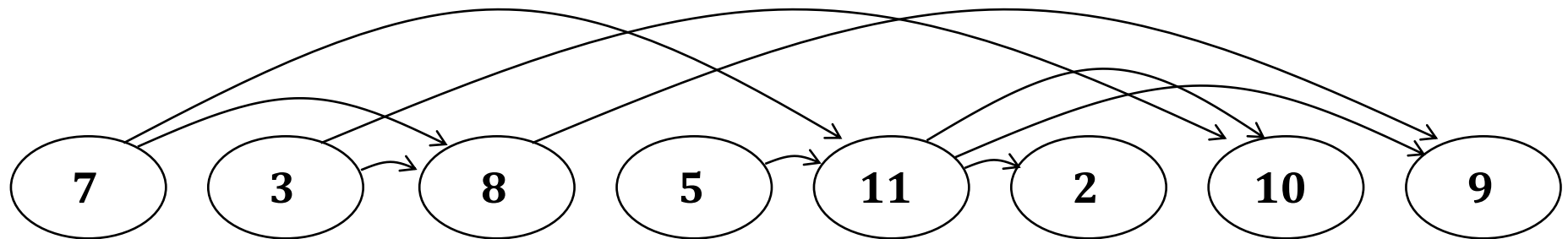
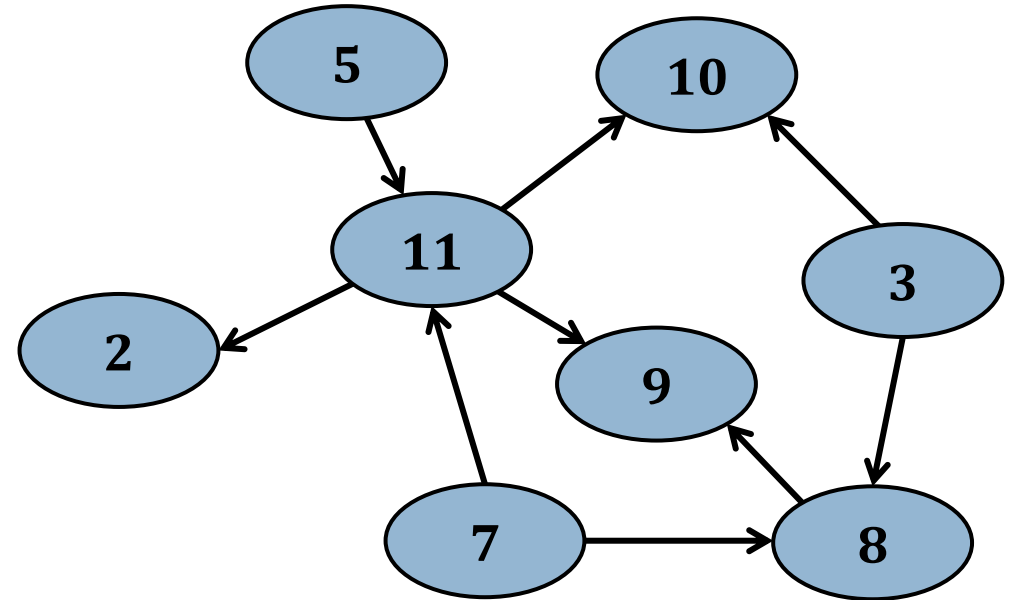
9

Αλγόριθμος **TS-Indeg(G)**

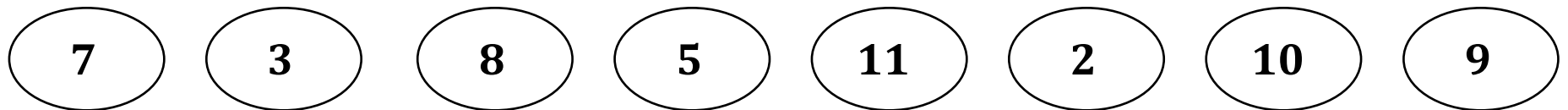
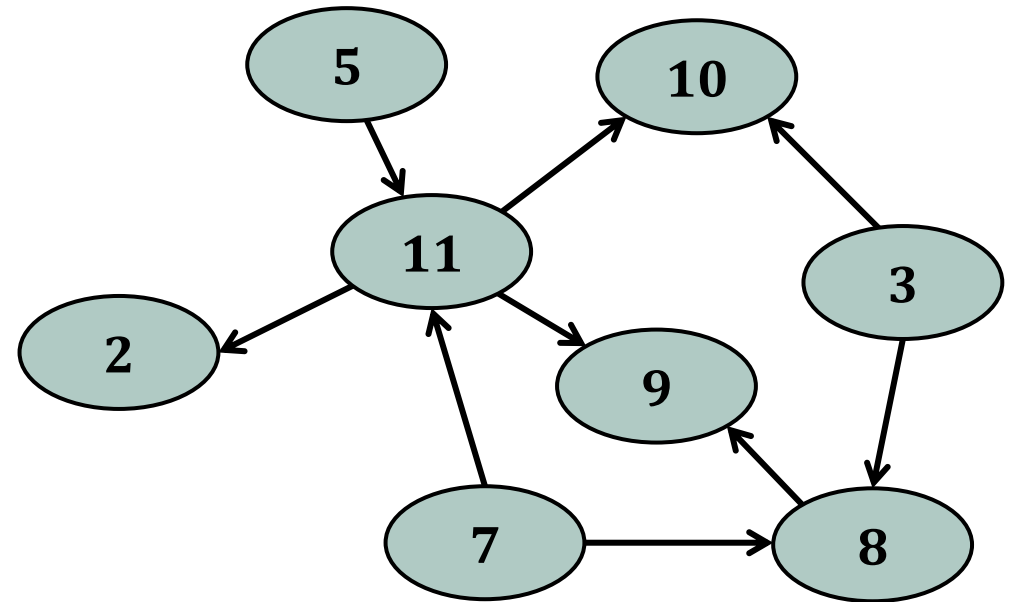
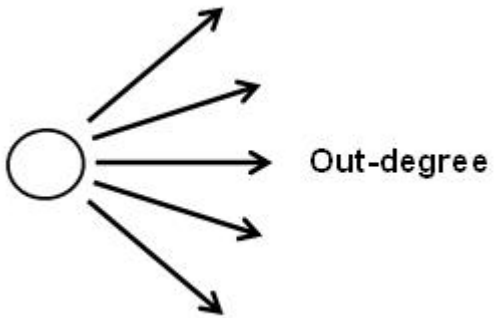
Υπολογισμός μιας TS του $G=(V,E)$

Τερματισμός Αλγόριθμου

Ορθότητα
Πολυπλοκότητα

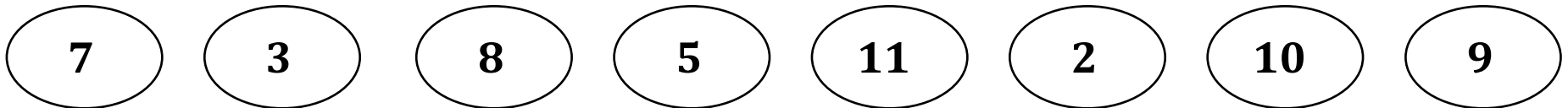
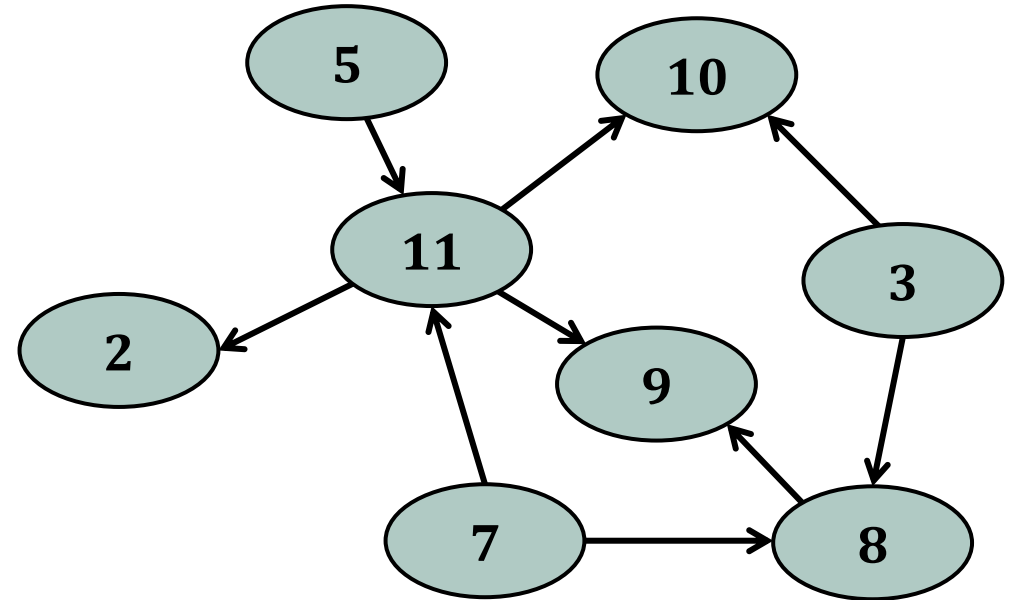


Μέθοδος Out-degree



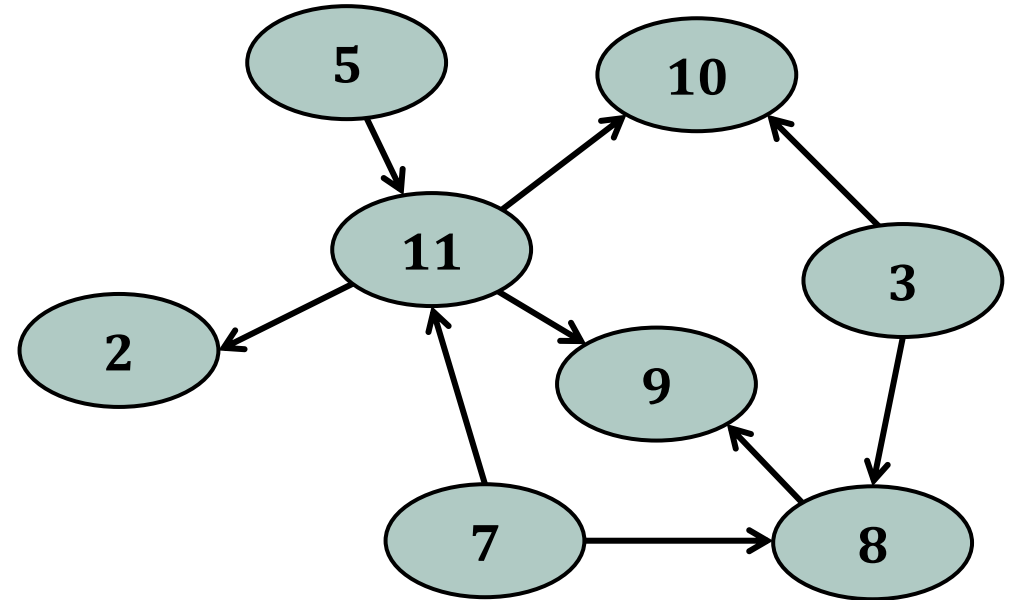
Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$



Αλγόριθμος **TS-Outdeg**(G)

Υπολογισμός μιας TS του $G=(V,E)$

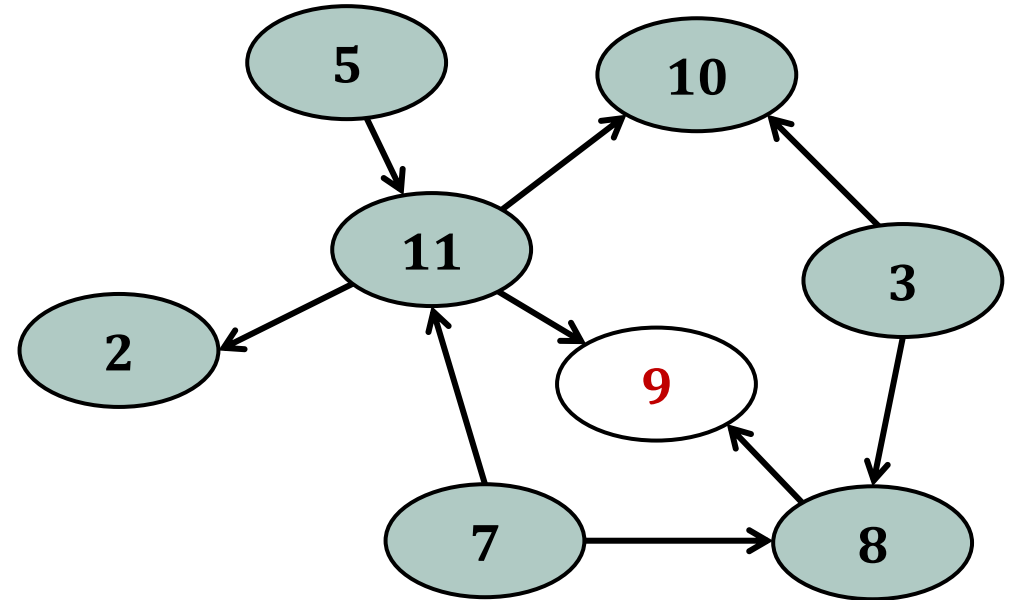


Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{Out-degree} = 0$

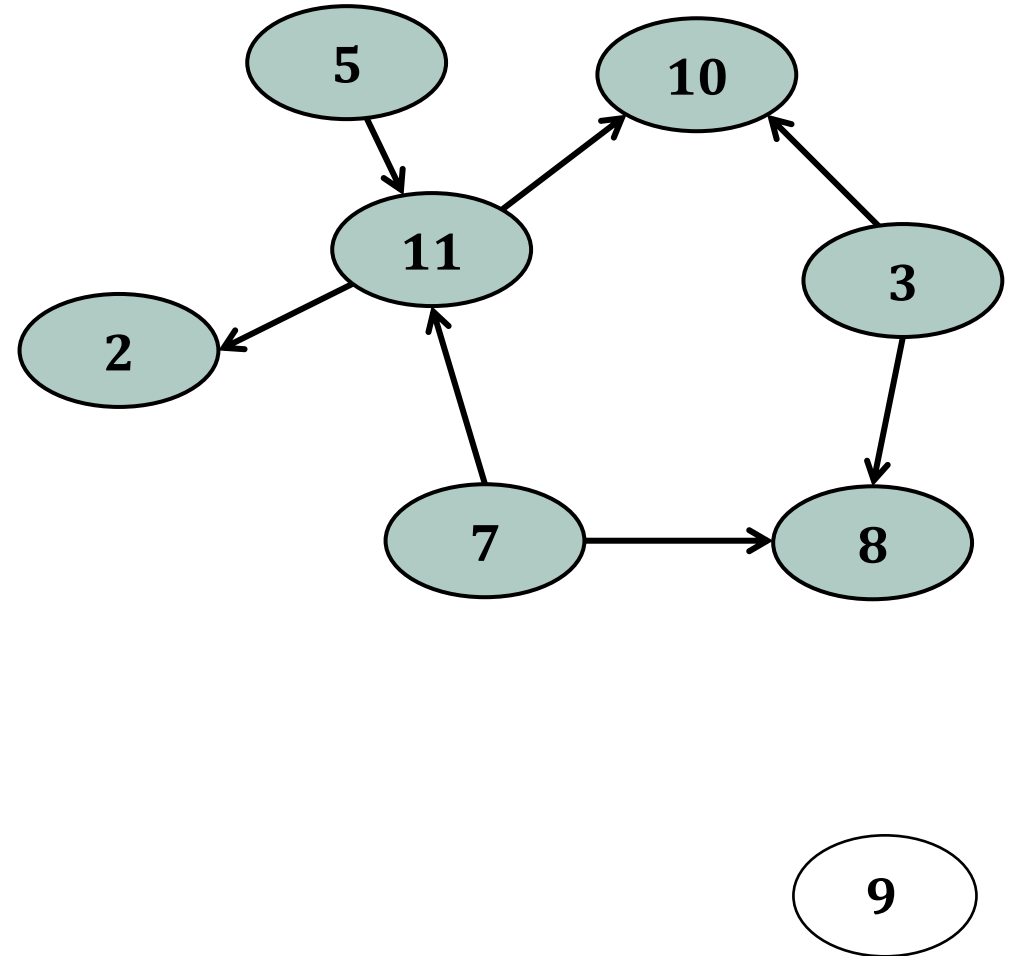
Κόμβος **9**



Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου **9**
από το γράφημα, και
Εισαγωγή του στην αρχή
μια λίστας

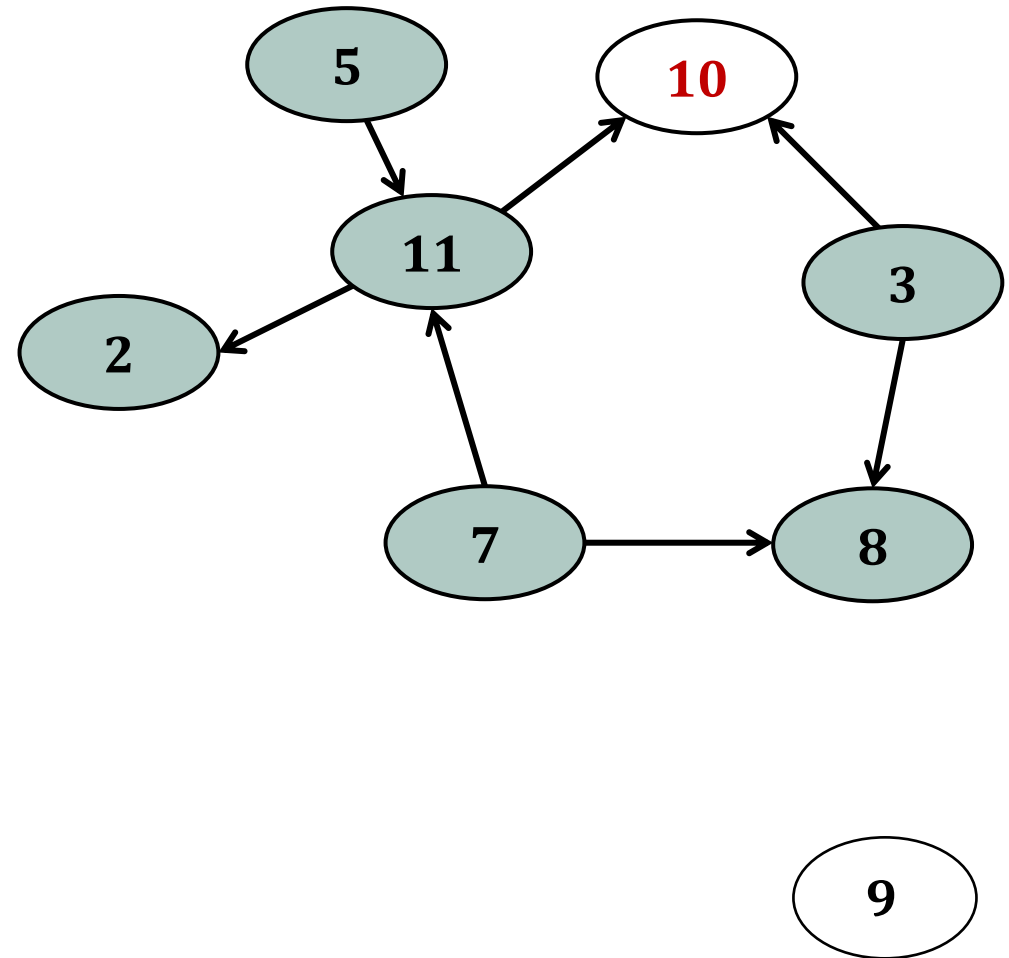


Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{Out-degree} = 0$

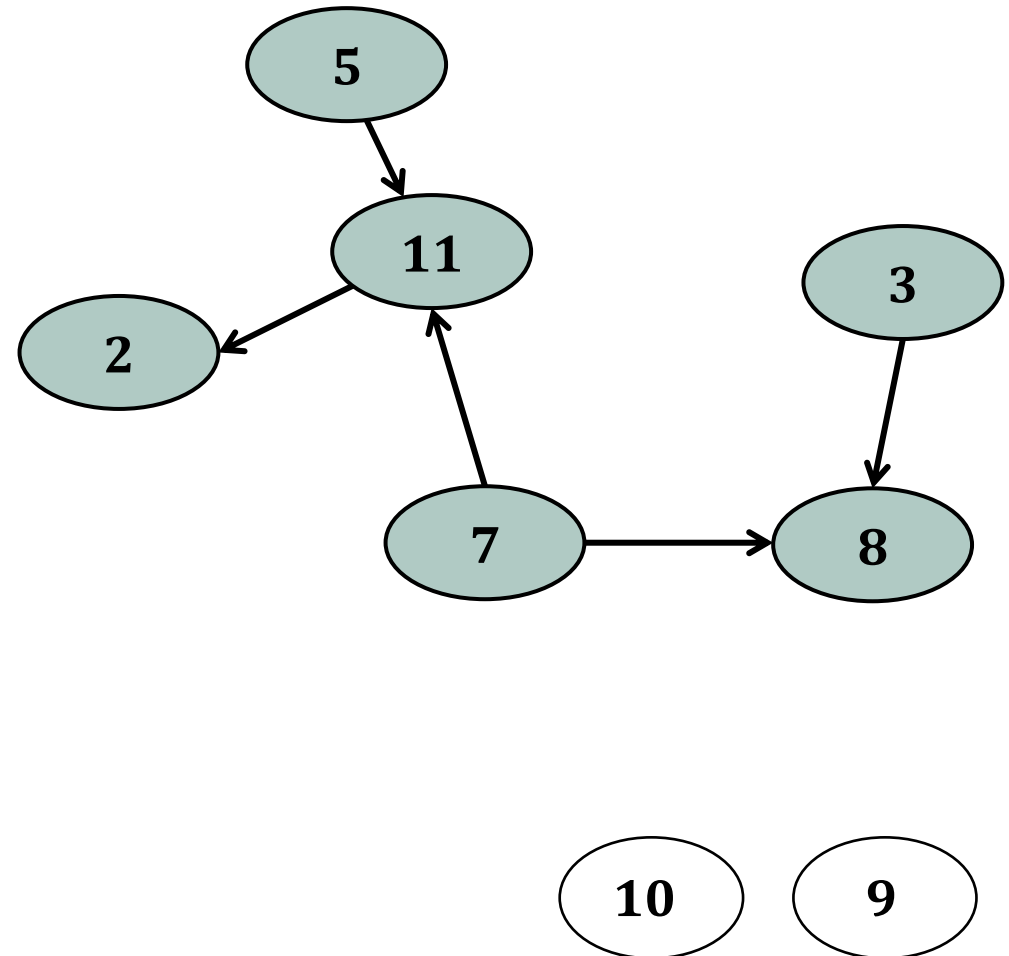
Κόμβος **10**



Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου 10
από το γράφημα, και
Εισαγωγή του στην αρχή
μια λίστας

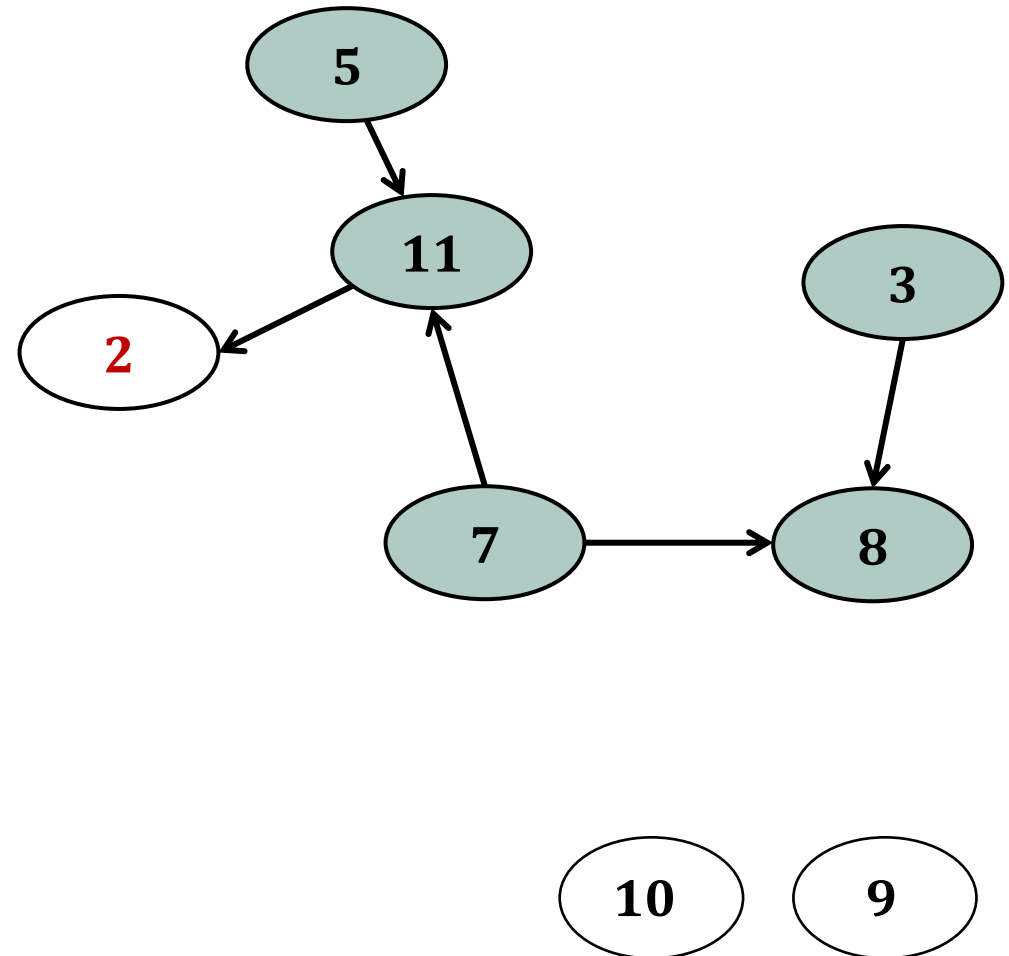


Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{Out-degree} = 0$

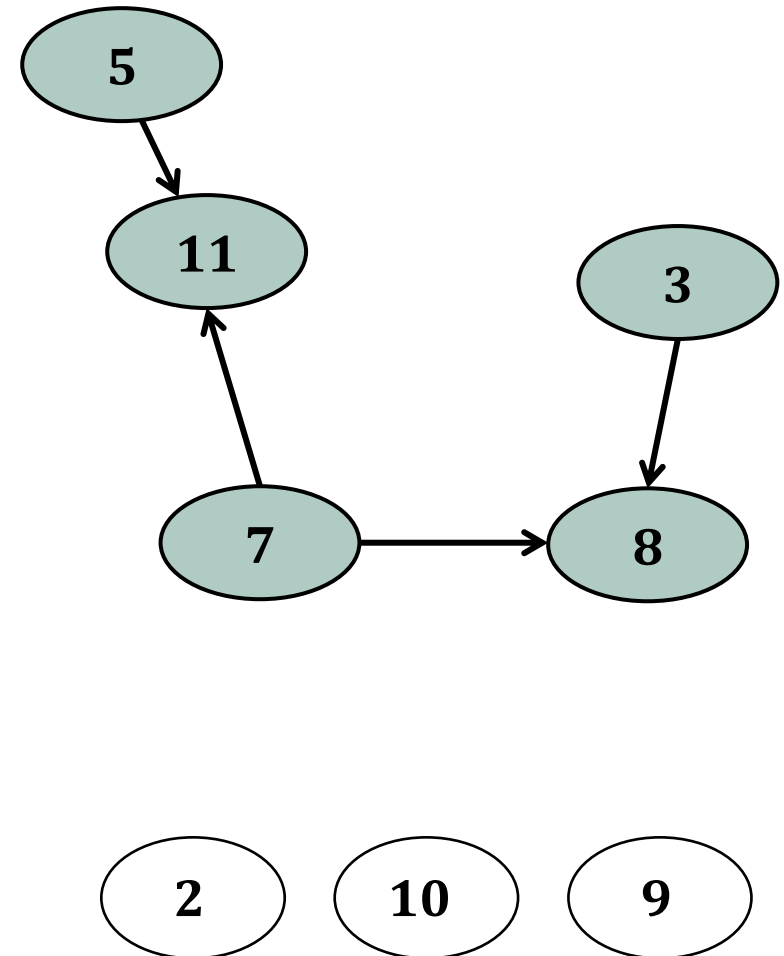
Κόμβος **2**



Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου 2
από το γράφημα, και
Εισαγωγή του στην αρχή
μια λίστας

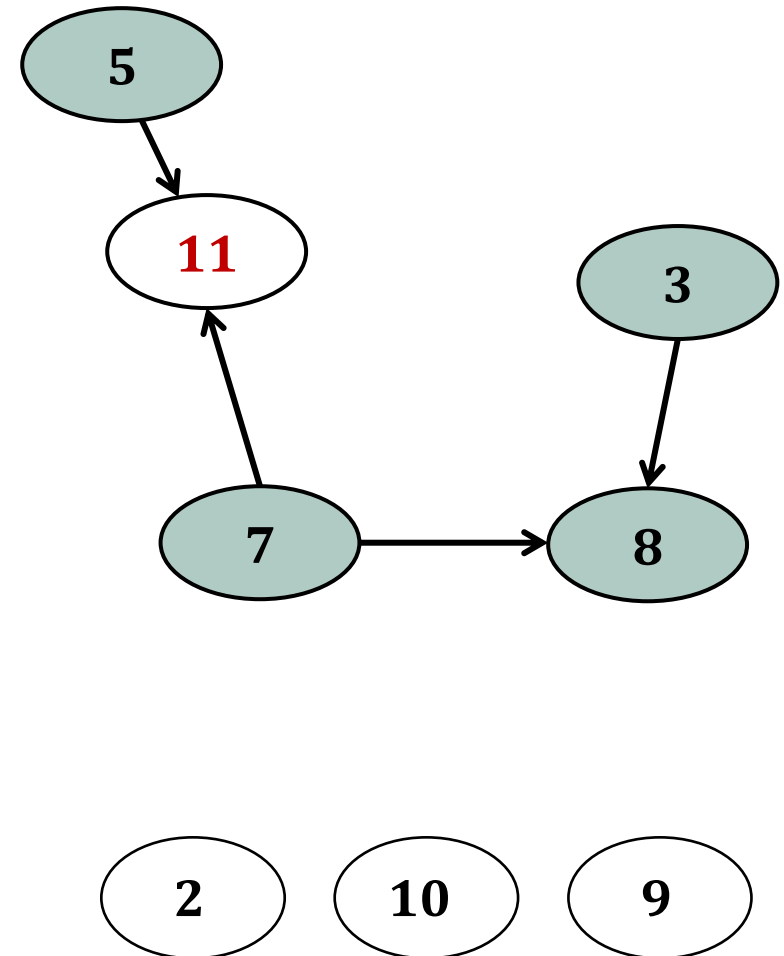


Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{Out-degree} = 0$

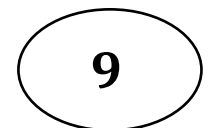
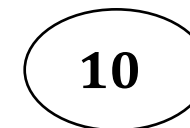
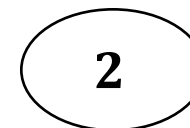
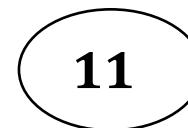
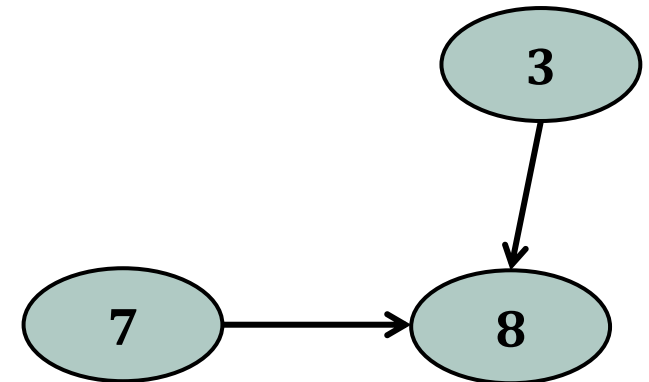
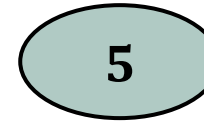
Κόμβος **11**



Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου 11
από το γράφημα, και
Εισαγωγή του στην αρχή
μια λίστας

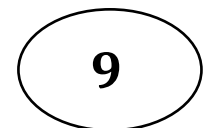
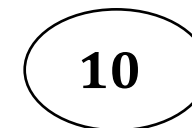
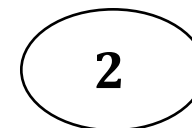
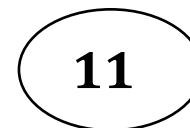
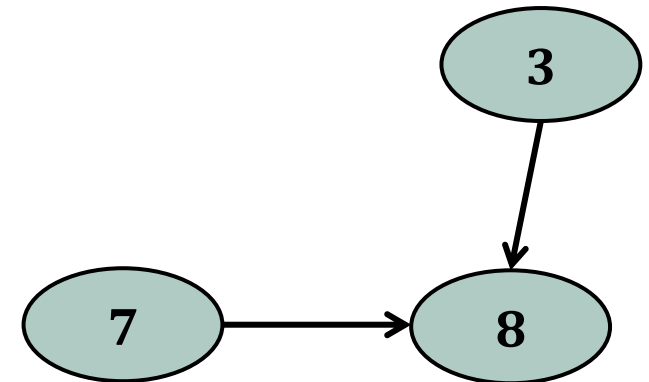


Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{Out-degree} = 0$

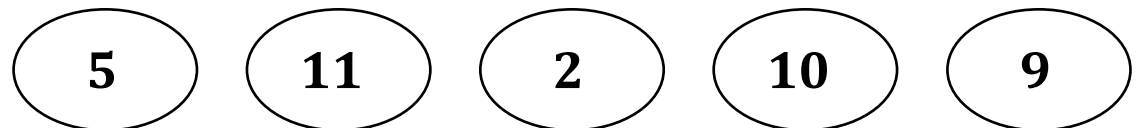
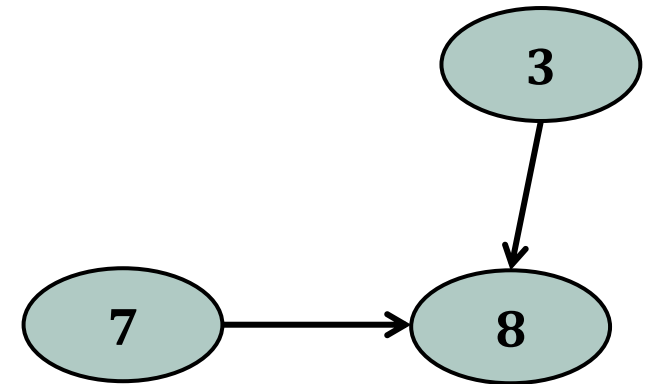
Κόμβος **5**



Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου **5**
από το γράφημα, και
Εισαγωγή του στην αρχή
μια λίστας

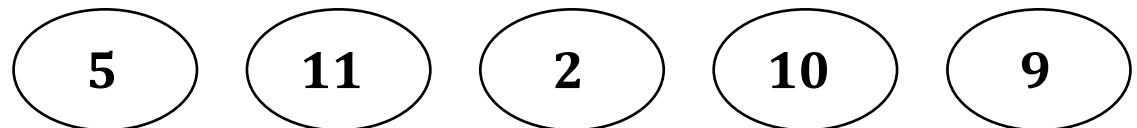
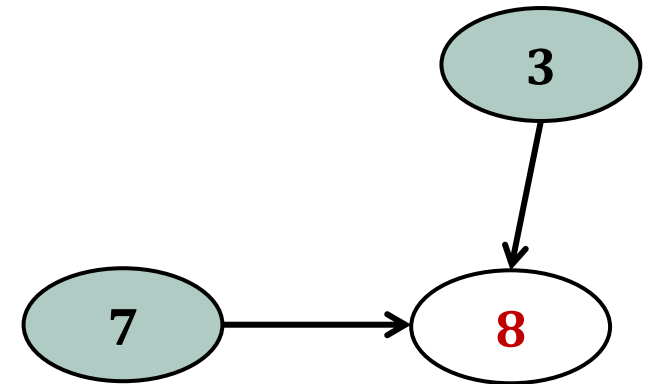


Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{Out-degree} = 0$

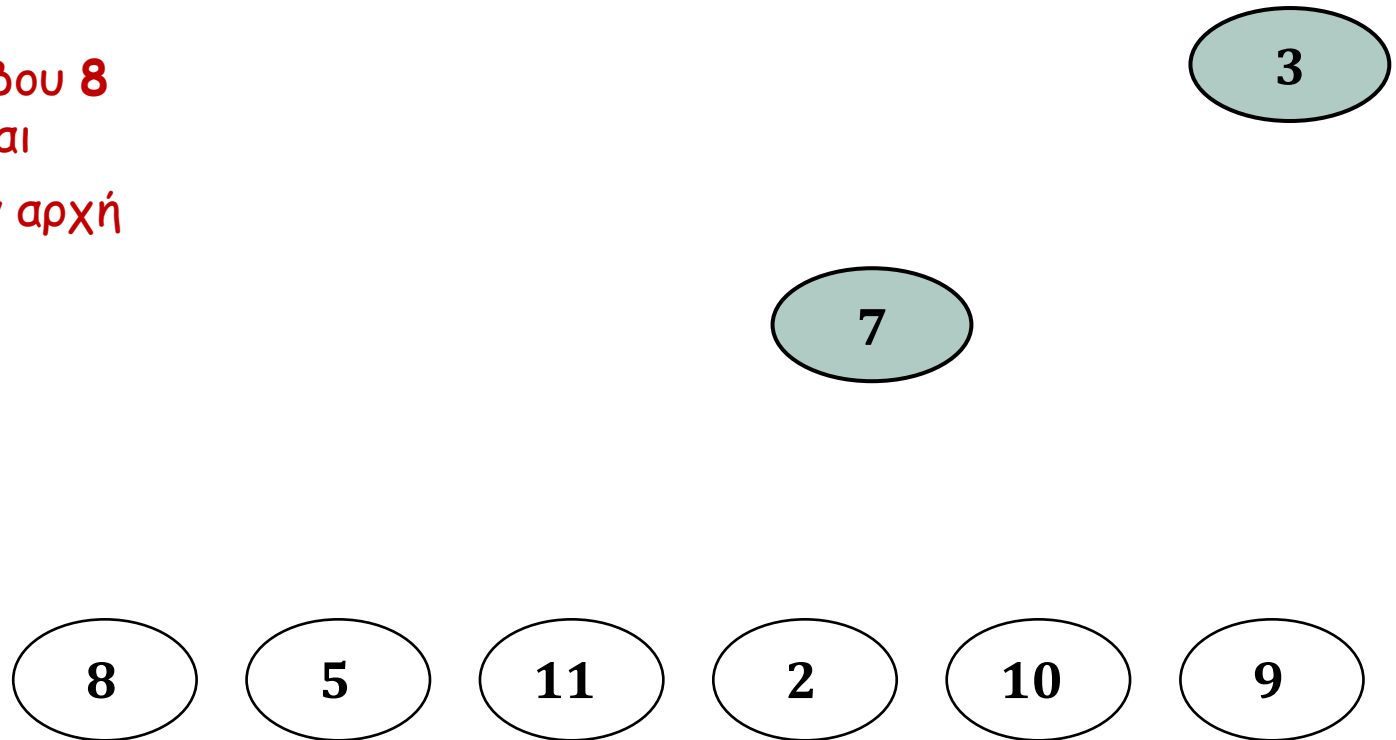
Κόμβος **8**



Αλγόριθμος **TS-Outdeg**(G)

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου 8
από το γράφημα, και
Εισαγωγή του στην αρχή
μια λίστας

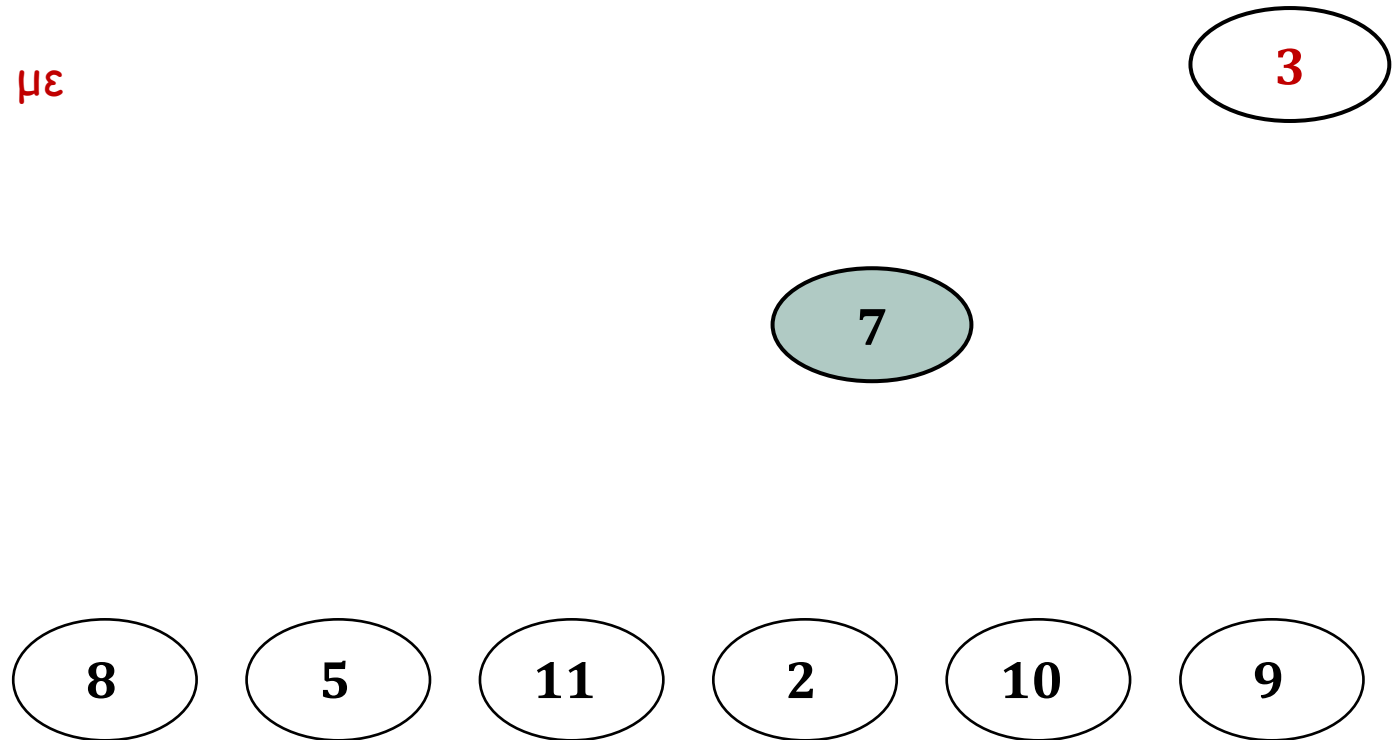


Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{Out-degree} = 0$

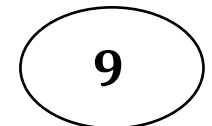
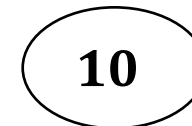
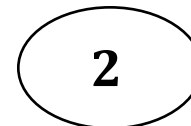
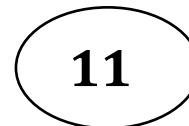
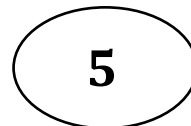
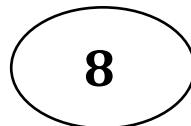
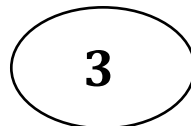
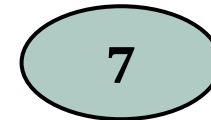
Κόμβος **3**



Αλγόριθμος **TS-Outdeg**(G)

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου 3
από το γράφημα, και
Εισαγωγή του στην αρχή
μια λίστας

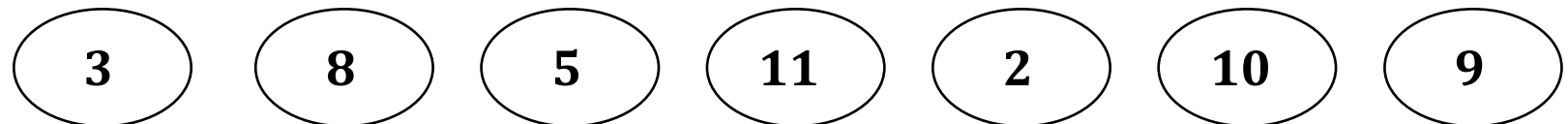
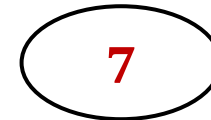


Αλγόριθμος **TS-Outdeg**(G)

Υπολογισμός μιας TS του $G=(V,E)$

Επίλεξε ένα κόμβο με
 $\text{Out-degree} = 0$

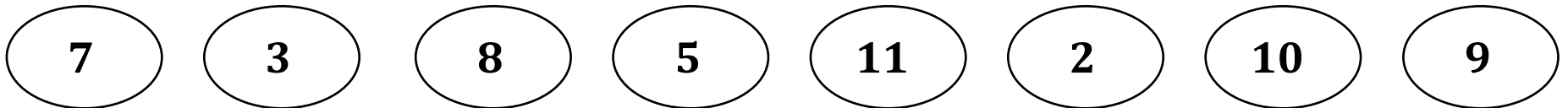
Κόμβος **7**



Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

Διαγραφή του κόμβου 7
από το γράφημα, και
Εισαγωγή του στην αρχή
μια λίστας

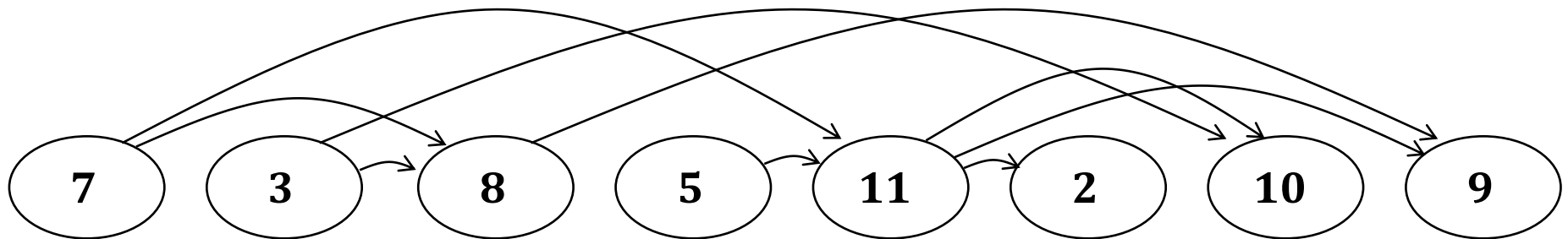
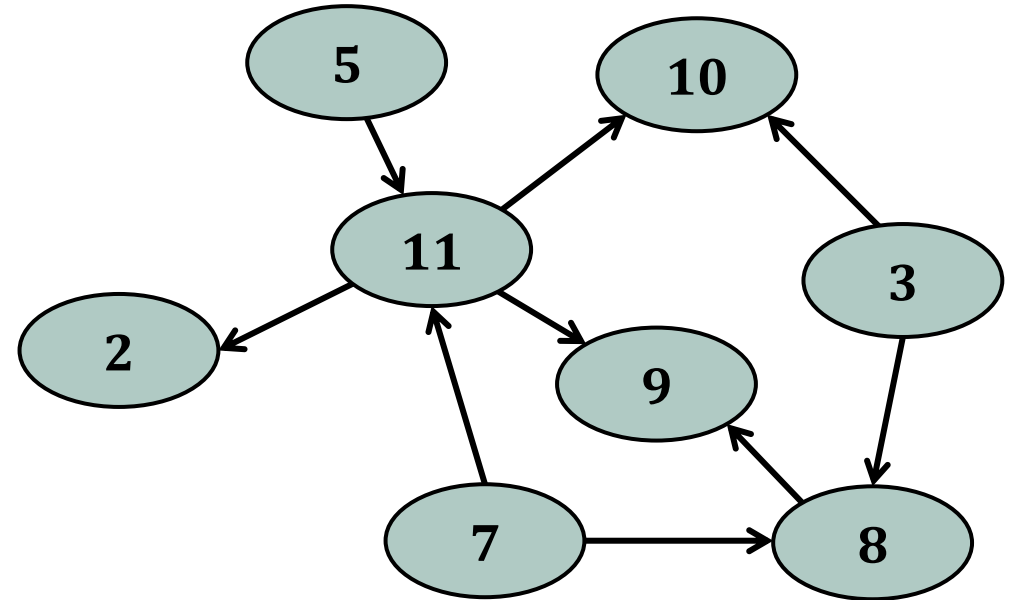


Αλγόριθμος **TS-Outdeg(G)**

Υπολογισμός μιας TS του $G=(V,E)$

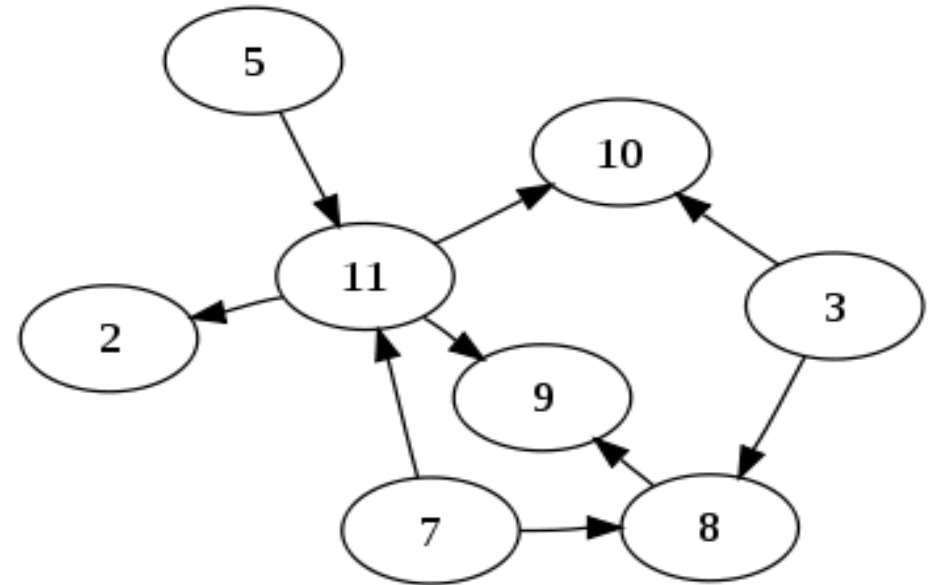
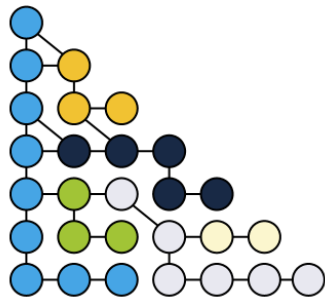
Τερματισμός Αλγόριθμου

Ορθότητα
Πολυπλοκότητα



Μέθοδος DFS

Depth-First Search



7

3

8

5

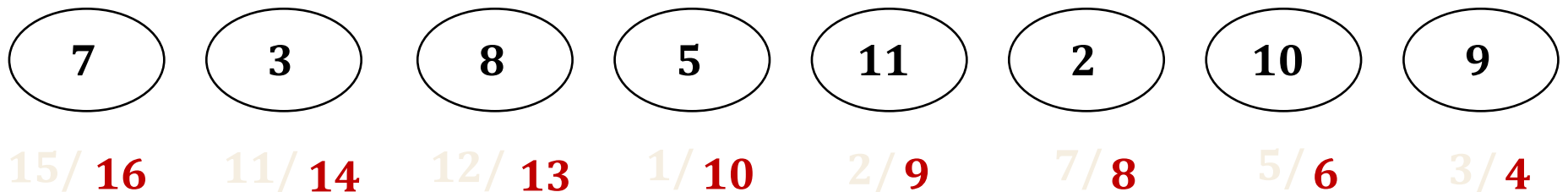
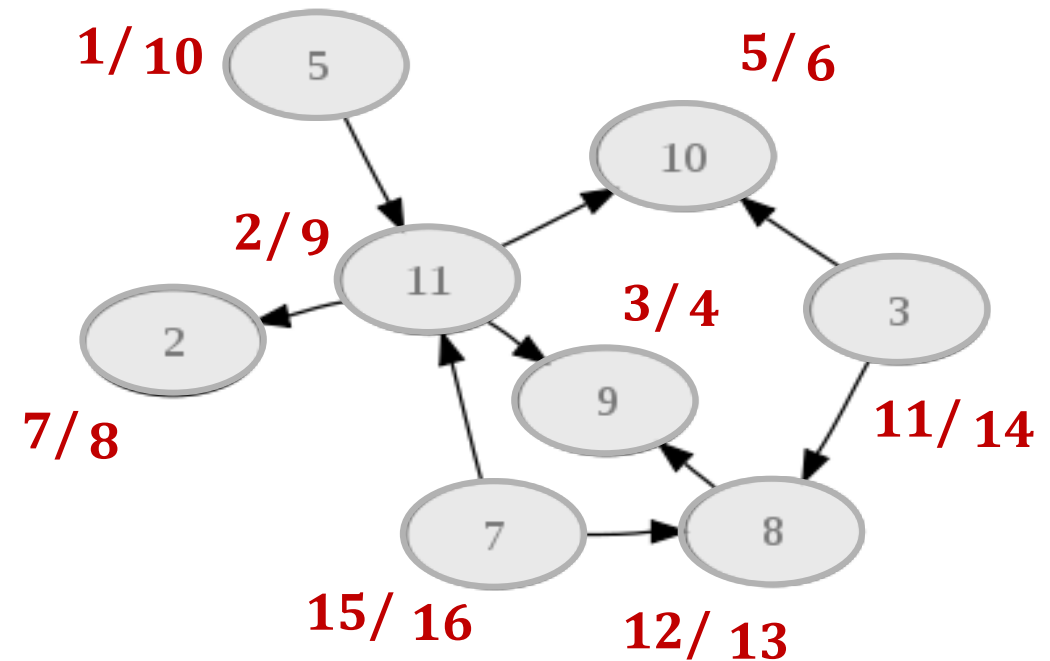
11

2

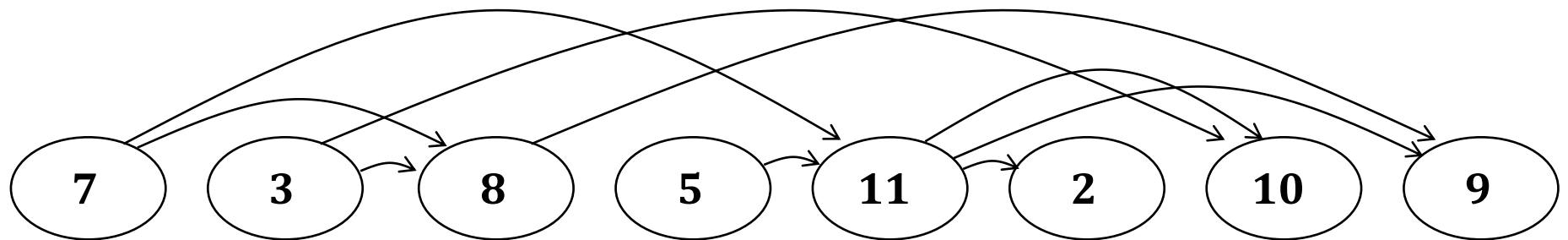
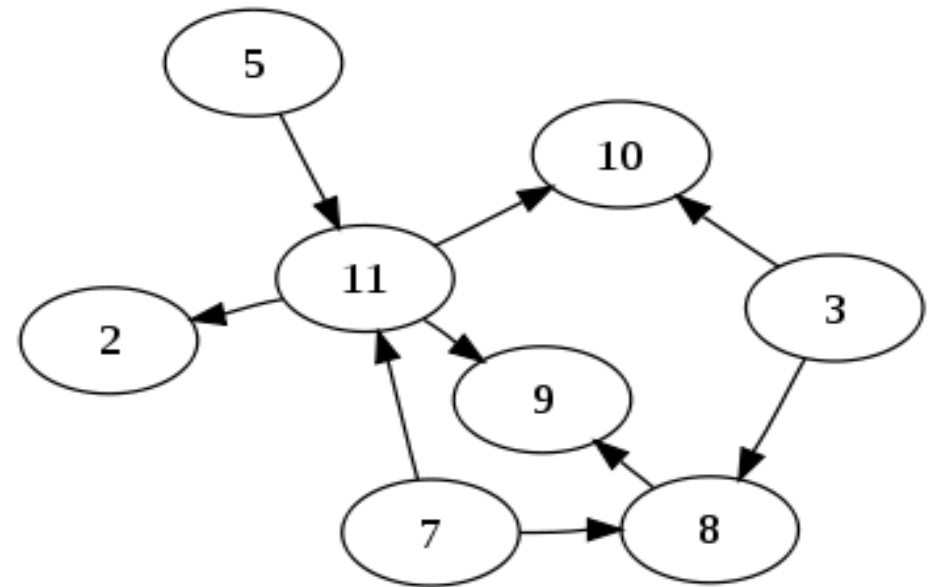
10

9

Αλγόριθμος **TS-DFS**(G)



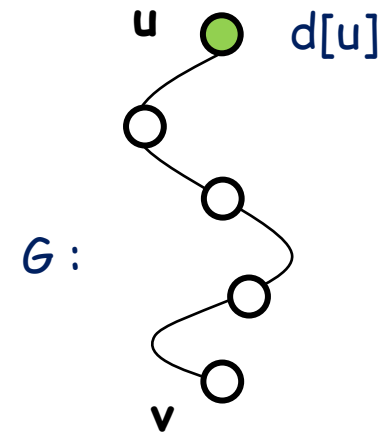
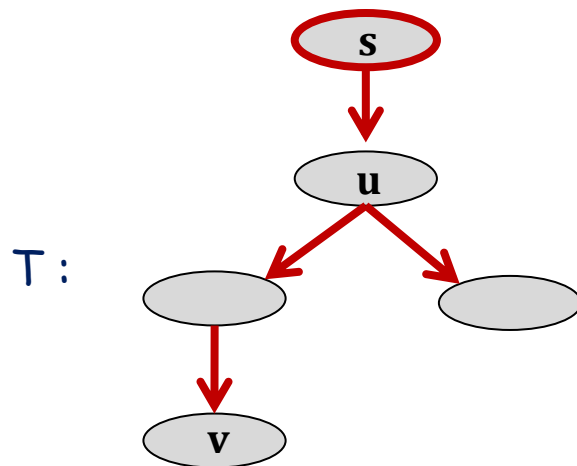
Αλγόριθμος **TS-DFS**(G)



Ορθότητα Αλγορίθμου TS-DFS(G)

Θεώρημα Λευκής-διαδρομής (White-path theorem):

Έστω T ένα DFS-δένδρο ενός (κατευθυνόμενου ή μη-κατευθυνόμενου) γραφήματος G . Ένας κόμβος v είναι **απόγονος του κόμβου u** στο DFS-δένδρο T εάν-ν στον χρόνο πρώτης επίσκεψης $d[u]$ του κόμβου u , υπάρχει διαδρομή $u \rightsquigarrow v$ στο G που αποτελείται **μόνο** από **λευκούς κόμβους**.



Ορθότητα Αλγορίθμου TS-DFS(G)

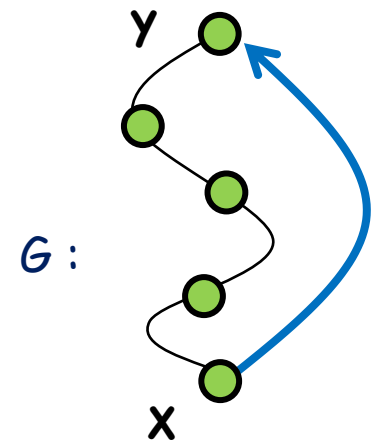
Θεώρημα 1:

Ένα κατευθυνόμενο γράφημα G είναι άκυκλο εάν-ν μια DFS διέλευση του G δεν δημιουργεί οπισθο-ακμές (back-edges).

Απόδειξη: ($\Rightarrow$) Έστω ότι το γράφημα G είναι άκυκλο και έστω (x, y) μια οπισθο-ακμή της DFS διέλευσής του.

Τότε, ο κόμβος y είναι προκάτοχος του x στο DFS-δένδρο του G .

Επομένως, υπάρχει διαδρομή $y \rightsquigarrow x$ στο G και άρα η οπισθο-ακμή (x, y) δημιουργεί κύκλο, άτοπο.



Ορθότητα Αλγορίθμου TS-DFS(G)

Θεώρημα 2:

Ένα κατευθυνόμενο γράφημα G είναι άκυκλο εάν-ν μια DFS διέλευση του G δεν δημιουργεί οπισθο-ακμές (back-edges).

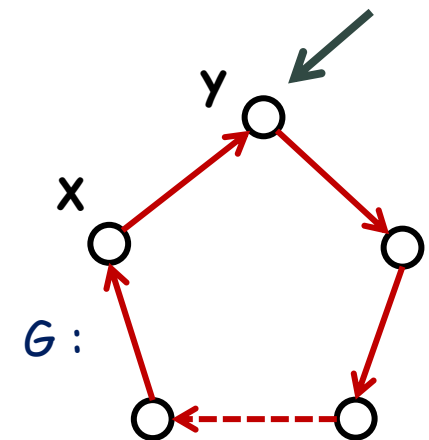
Απόδειξη: ($\Leftarrow$) Έστω το γράφημα G έχει ένα κύκλο C .

Έστω y είναι ο πρώτος κόμβος του C στον οποίο γίνεται επίσκεψη κατά την DFS διέλευση του G και έστω (x, y) η ακμή του κύκλου C .

Στον χρόνο $d[y]$ υπάρχει διαδρομή $y \rightsquigarrow x$ από λευκούς κόμβους.

Επομένως, ο κόμβος x γίνεται διάδοχος του y στο DFS-δένδρο.

Άρα, (x, y) οπισθο-ακμή, άτοπο. QED



Ορθότητα Αλγορίθμου TS-DFS(G)

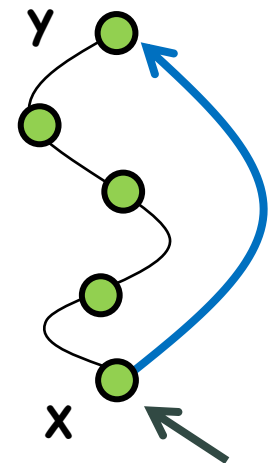
Θεώρημα 3:

Δεδομένου ενός άκυκλου κατευθυνόμενου γραφήματος G , ο αλγόριθμος TS-DFS(G) υπολογίζει μια τοπολογική ταξινόμηση των κόμβων του G .

Απόδειξη: Εκτελούμε DFS στο γράφημα G και έστω $f[u]$ είναι ο χρόνος τελευταίας επίσκεψης του κόμβου u του G , $\forall u \in V(G)$.

Αρκεί να δείξουμε ότι για κάθε ζεύγος κόμβων (x, y) , εάν $(x, y) \in E(G)$ τότε $f[y] < f[x]$.

Έστω (x, y) μια τυχαία ακμή η οποία «εξετάζεται» όταν η DFS διέλευση του G «βρίσκεται» στον κόμβο x . Τότε, ο y δεν μπορεί να είναι **Πράσινος**, διότι θα ήταν προκάτοχος του x στο **DFS-δένδρο** του G και επομένως η ακμή (x, y) θα ήταν οπισθο-ακμή, άτοπο. Άρα ο κόμβος y είναι **Λευκός** ή **Μαύρος**.



Ορθότητα Αλγορίθμου TS-DFS(G)

Θεώρημα 3:

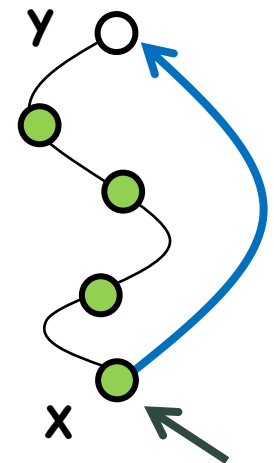
Δεδομένου ενός άκυκλου κατευθυνόμενου γραφήματος G , ο αλγόριθμος TS-DFS(G) υπολογίζει μια τοπολογική ταξινόμηση των κόμβων του G .

Απόδειξη: Εκτελούμε DFS στο γράφημα G και έστω $f[u]$ είναι ο χρόνος τελευταίας επίσκεψης του κόμβου u του G , $\forall u \in V(G)$.

Αρκεί να δείξουμε ότι για κάθε ζεύγος κόμβων (x, y) , εάν $(x, y) \in E(G)$ τότε $f[y] < f[x]$.

(1) Κόμβος y = Λευκός:

Τότε ο κόμβος y θα είναι διάδοχος του x στο DFS δένδρο και επομένως $f[y] < f[x]$.



Ορθότητα Αλγορίθμου TS-DFS(G)

Θεώρημα 3:

Δεδομένου ενός άκυκλου κατευθυνόμενου γραφήματος G , ο αλγόριθμος TS-DFS(G) υπολογίζει μια τοπολογική ταξινόμηση των κόμβων του G .

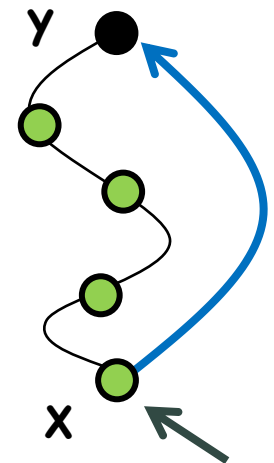
Απόδειξη: Εκτελούμε DFS στο γράφημα G και έστω $f[u]$ είναι ο χρόνος τελευταίας επίσκεψης του κόμβου u του G , $\forall u \in V(G)$.

Αρκεί να δείξουμε ότι για κάθε ζεύγος κόμβων (x, y) , εάν $(x, y) \in E(G)$ τότε $f[y] < f[x]$.

(2) Κόμβος y = Μαύρος:

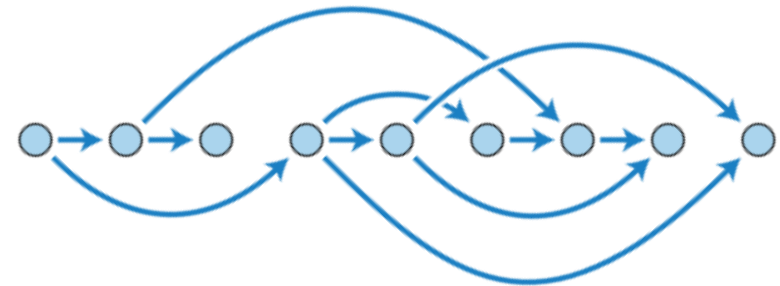
Τότε, ισχύει $f[y] < f[x]$.

Άρα, για κάθε ακμή $(x, y) \in E(G)$ ισχύει $f[y] < f[x]$. QED



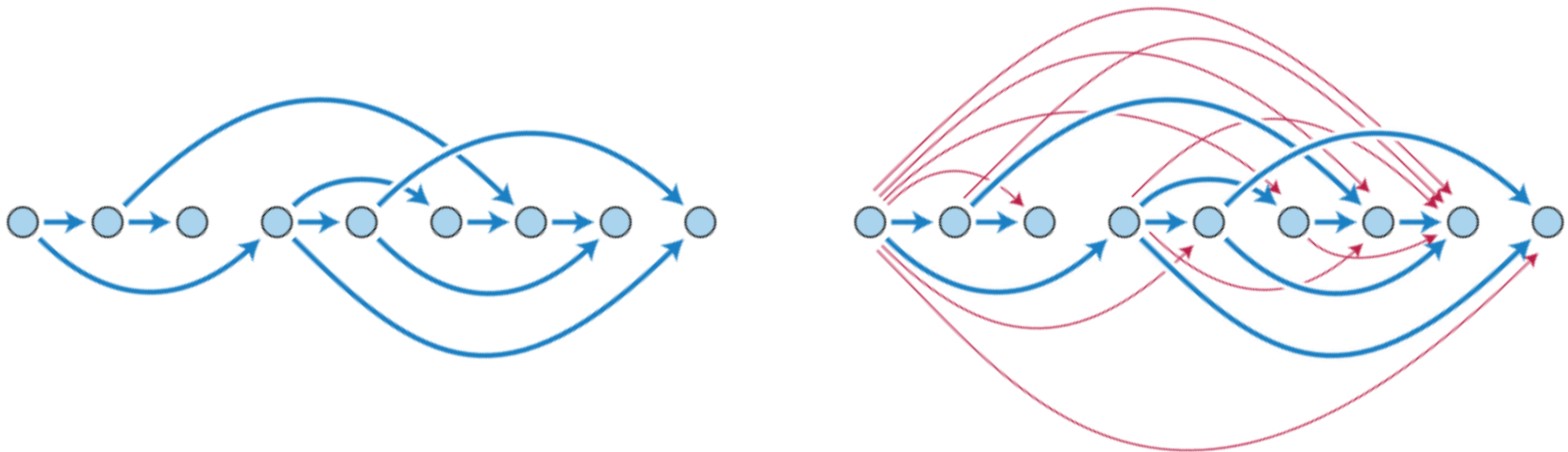
Ιδιότητες Τοπολογικής Ταξινόμησης

- Ένα κατευθυνόμενο γράφημα G είναι άκυκλο εάν-ν έχει μια τοπολογική ταξινόμηση των κόμβων του.



Ιδιότητες Τοπολογικής Ταξινόμησης

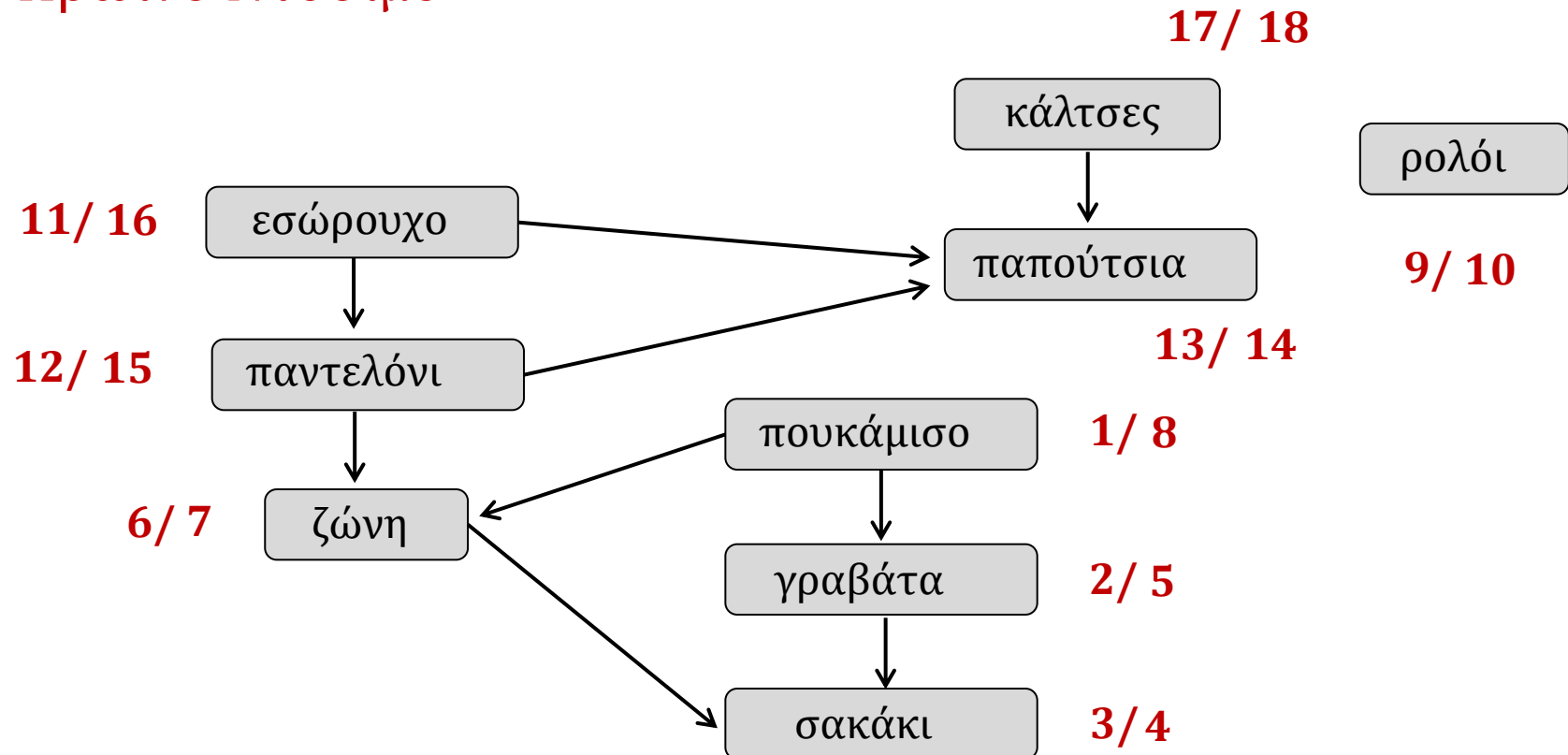
- Ένα κατευθυνόμενο γράφημα G είναι άκυκλο εάν-ν έχει μια τοπολογική ταξινόμηση των κόμβων του.



- Η **μεταβατική κλειστότητα** (transitive closure) ενός DAG G είναι ένα γράφημα H με $\max$ πλήθος ακμών που απεικονίζει την ίδια σχέση προσβασιμότητας (reachability relation). Το γράφημα H έχει την ακμή $u \rightarrow v$ εάν ο κόμβος u έχει πρόσβαση στον κόμβο v στο G .

Εφαρμογές Τοπολογική Ταξινόμηση

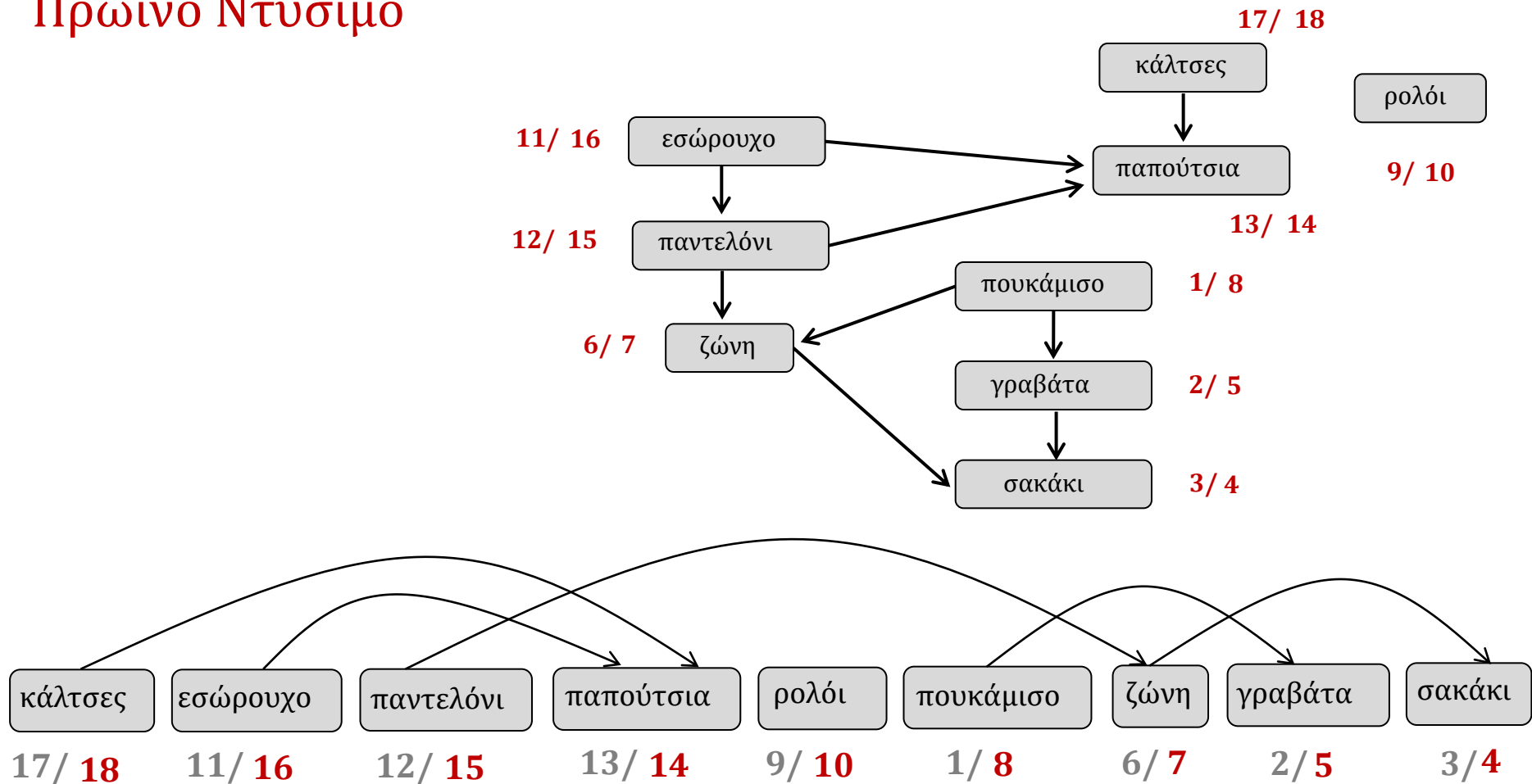
Ε Πρωινό Ντύσιμο



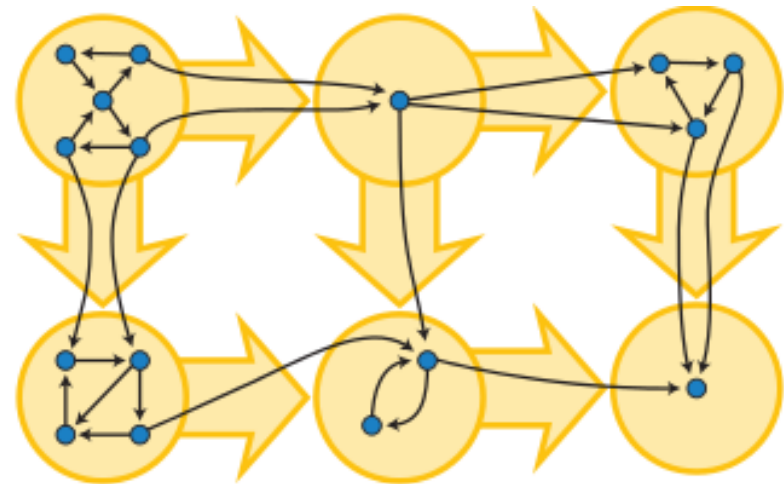
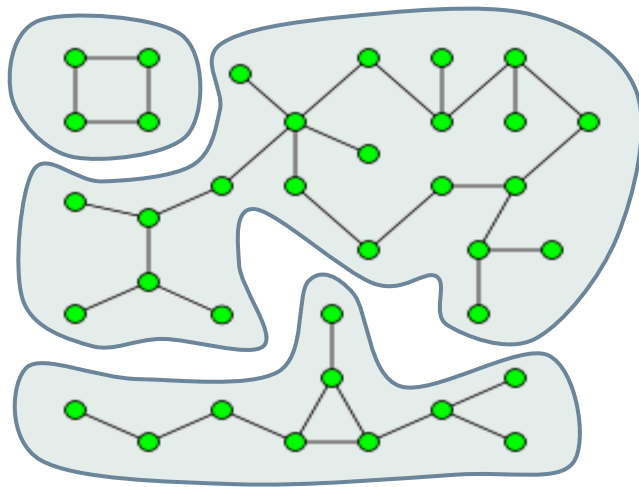
Εφαρμογές Τοπολογική Ταξινόμηση

Ε

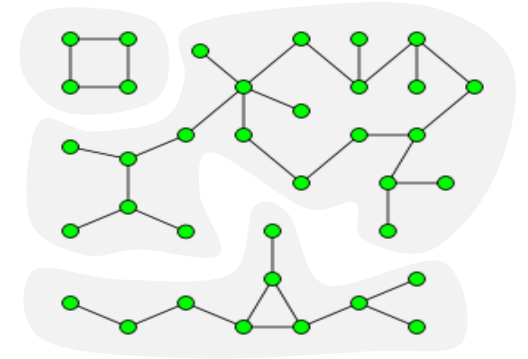
Πρωινό Ντύσιμο



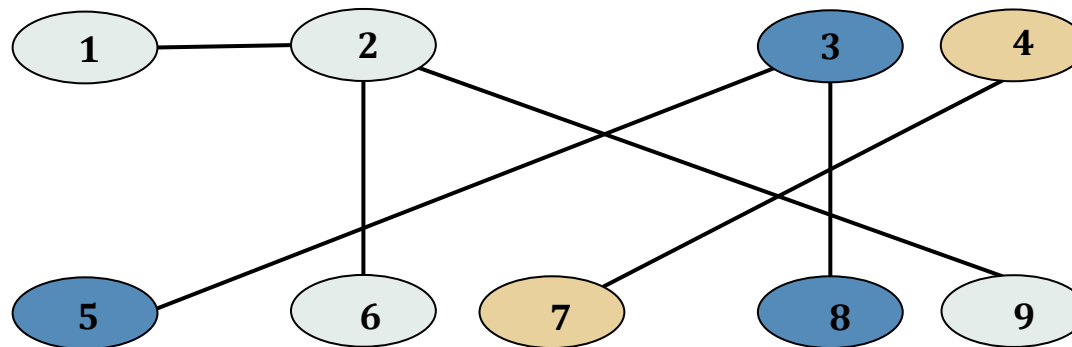
Συνεκτικές Συνιστώσες Ισχυρά Συνεκτικές Συνιστώσες



Συνεκτικές Συνιστώσες



Μη-κατευθυνόμενο Γράφημα

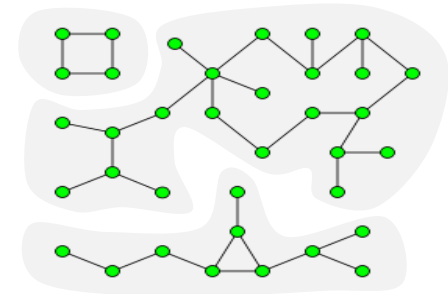
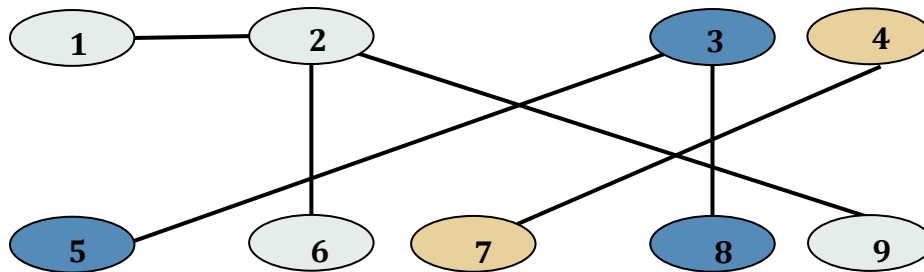


■ Συνεκτικές Συνιστώσες

Μια **συνεκτική συνιστώσα (CC)** ενός μη-κατευθυνόμενου γραφήματος G είναι ένα μεγιστοτικό σύνολο κόμβων $U \subseteq V(G)$ τέτοιο ώστε:

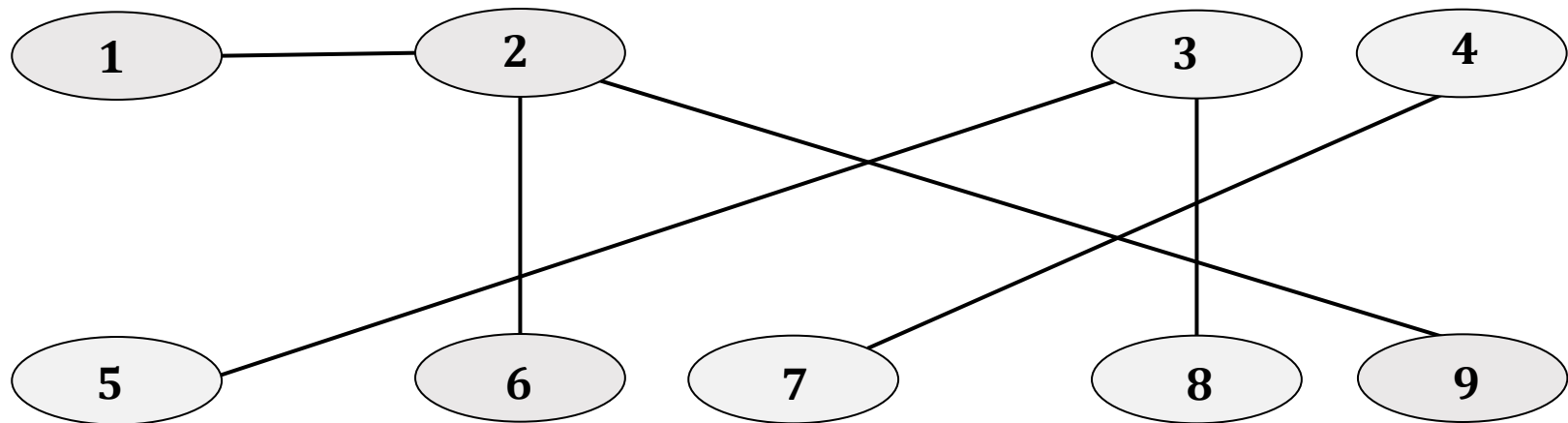
$$\forall x, y \in U \text{ ισχύει } x \rightsquigarrow y$$

όπου, με $x \rightsquigarrow y$ συμβολίζουμε την ύπαρξη στο G μιας διαδρομής $(x, z_1, z_2, \dots, z_k, y)$ από τον κόμβο x στον κόμβο y , $k \geq 0$.



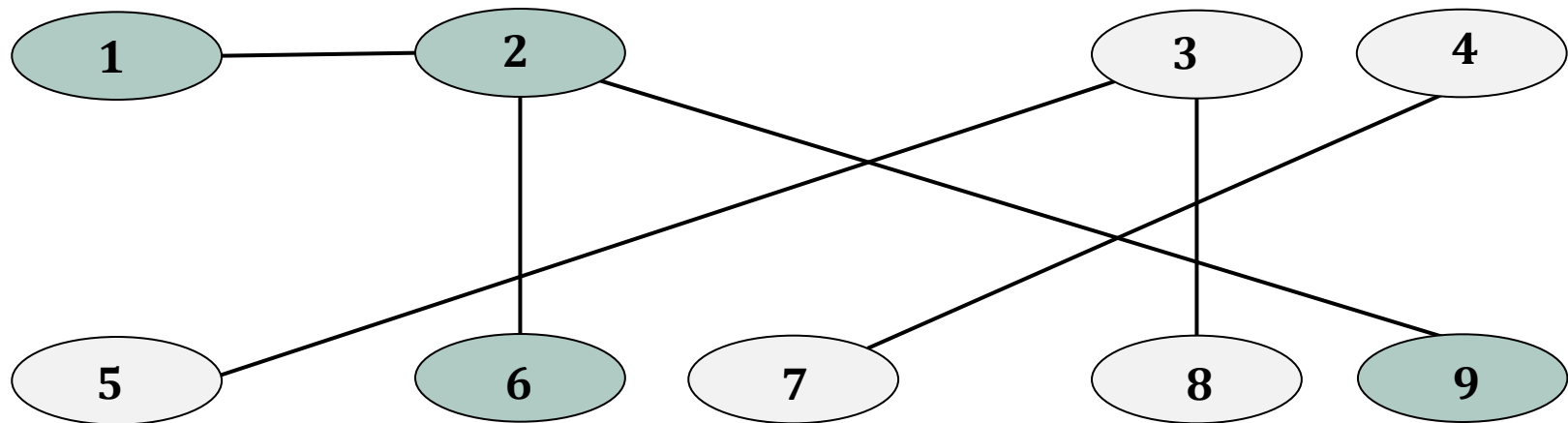


Παράδειγμα





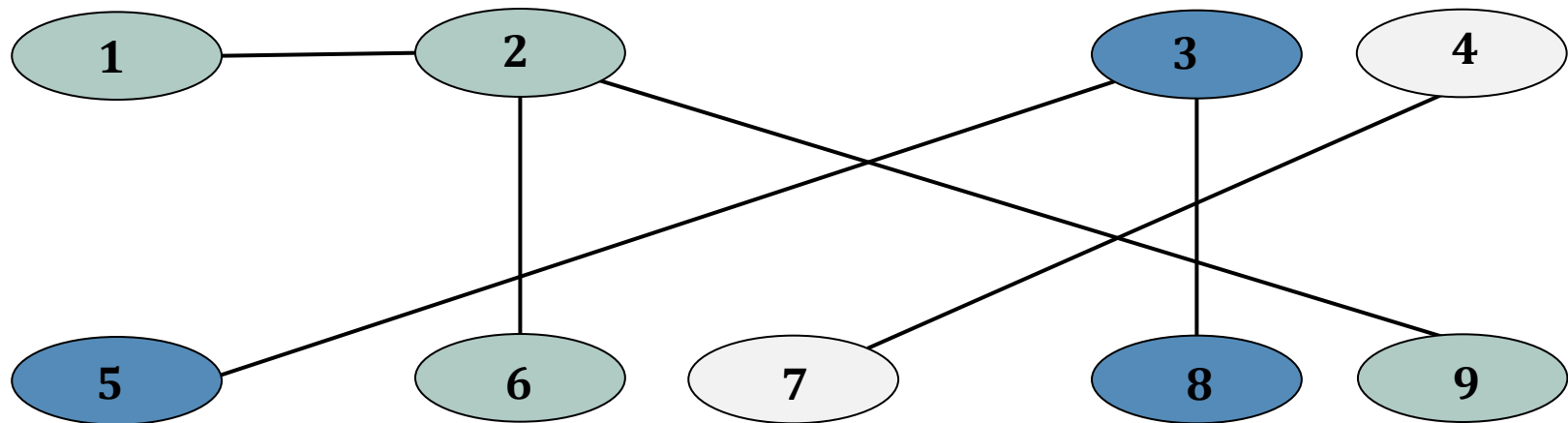
Παράδειγμα



$$CC_1 = \{1, 2, 6, 9\}$$



Παράδειγμα

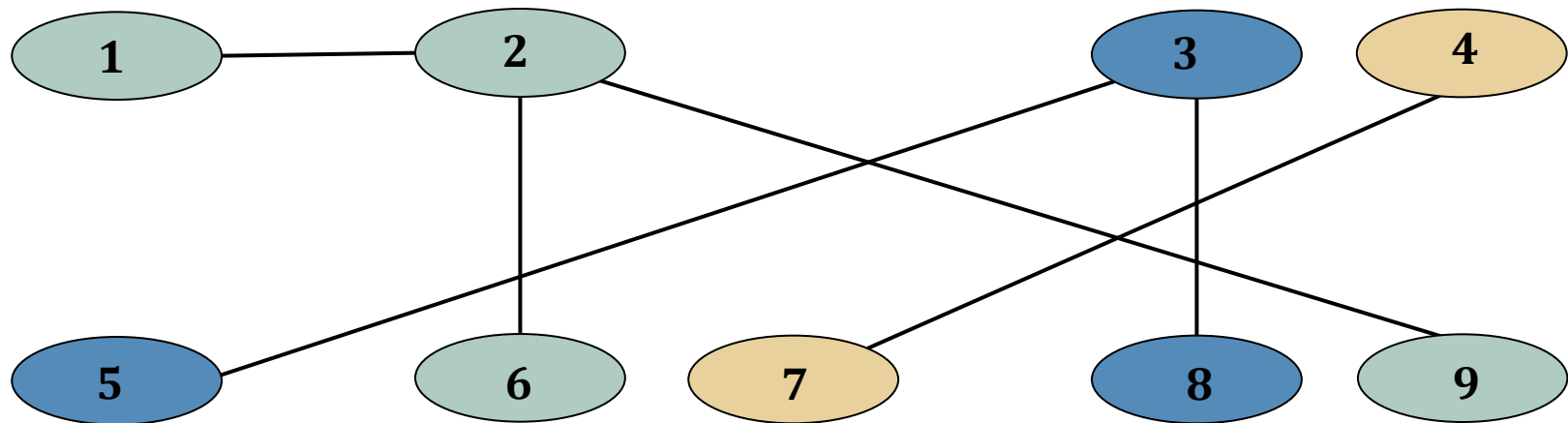


$$CC_1 = \{1, 2, 6, 9\}$$

$$CC_2 = \{3, 5, 8\}$$



Παράδειγμα



$$CC_1 = \{1, 2, 6, 9\}$$

$$CC_2 = \{3, 5, 8\}$$

$$CC_3 = \{4, 7\}$$

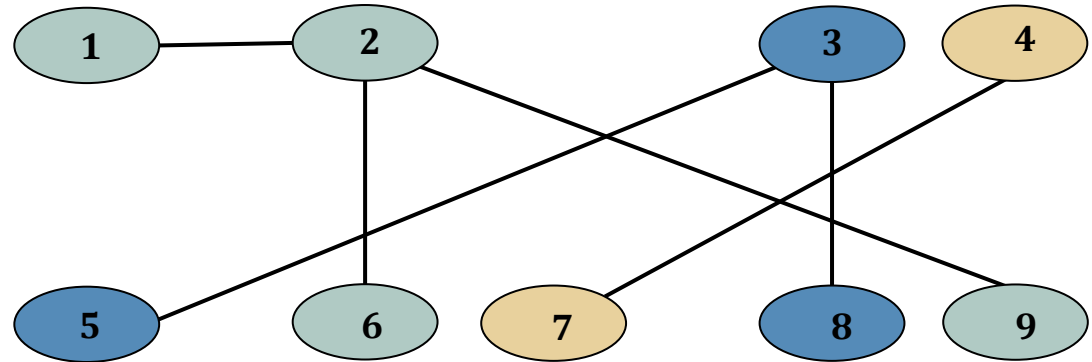
Συνεκτικές Συνιστώσες



Υπολογισμός CC

✓ BFS

✓ DFS

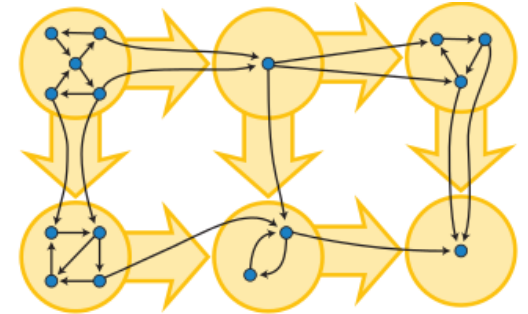


Εσωτερική Αναπαράσταση

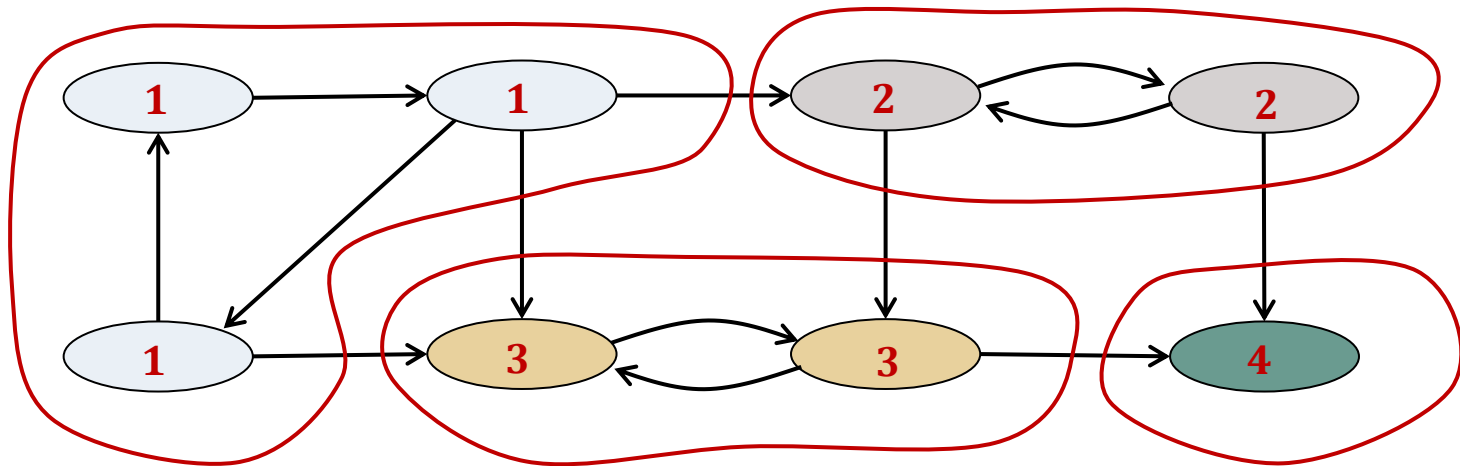
	1	2	3	4	5	6	7	8	9
CC :	1	1	2	3	2	1	3	2	1

CC :	1	1	3	4	3	1	4	3	1
------	---	---	---	---	---	---	---	---	---

Ισχυρά Συνεκτικές Συνιστώσες



Κατευθυνόμενο Γράφημα

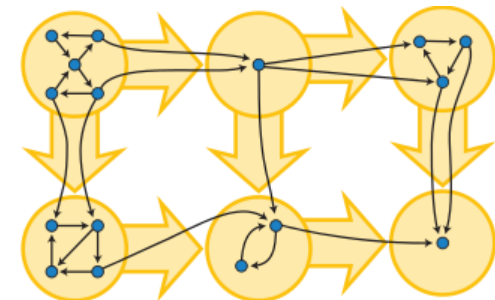
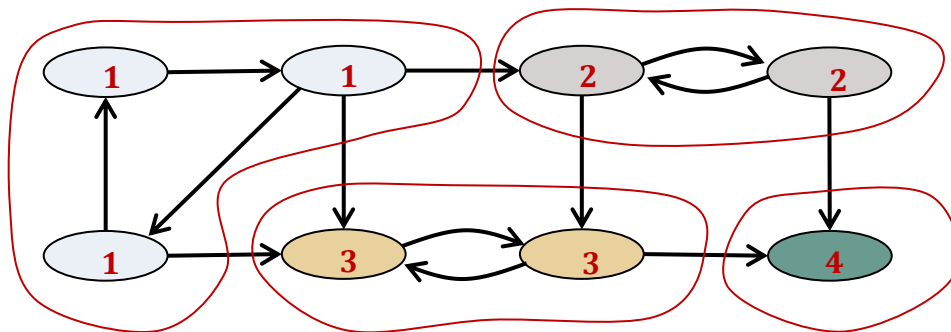


■ Ισχυρές Συνεκτικές Συνιστώσες

Μια **ισχυρή συνεκτική συνιστώσα (SCC)** ενός κατευθυνόμενου γραφήματος G είναι ένα μείζον σύνολο κόμβων $S \subseteq V(G)$ τέτοιο ώστε:

$$\forall x, y \in S \text{ ισχύει } x \rightsquigarrow y \text{ και } y \rightsquigarrow x$$

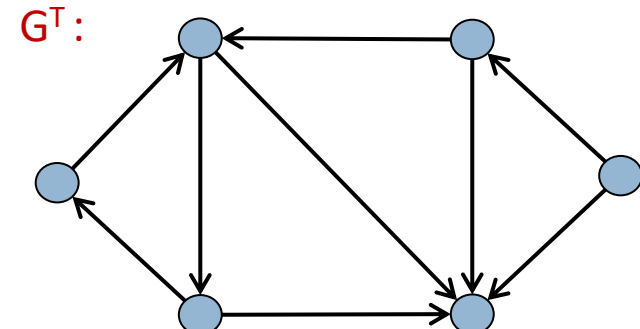
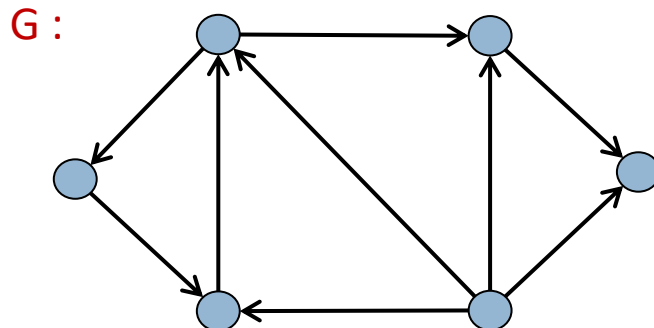
όπου, με $x \rightsquigarrow y$ συμβολίζουμε την ύπαρξη στο G μιας κατευθυνόμενης διαδρομής $(x, z_1, z_2, \dots, z_k, y)$ από τον κόμβο x στον κόμβο y , $k \geq 0$.



Ορισμός Αντίστροφου Γραφήματος

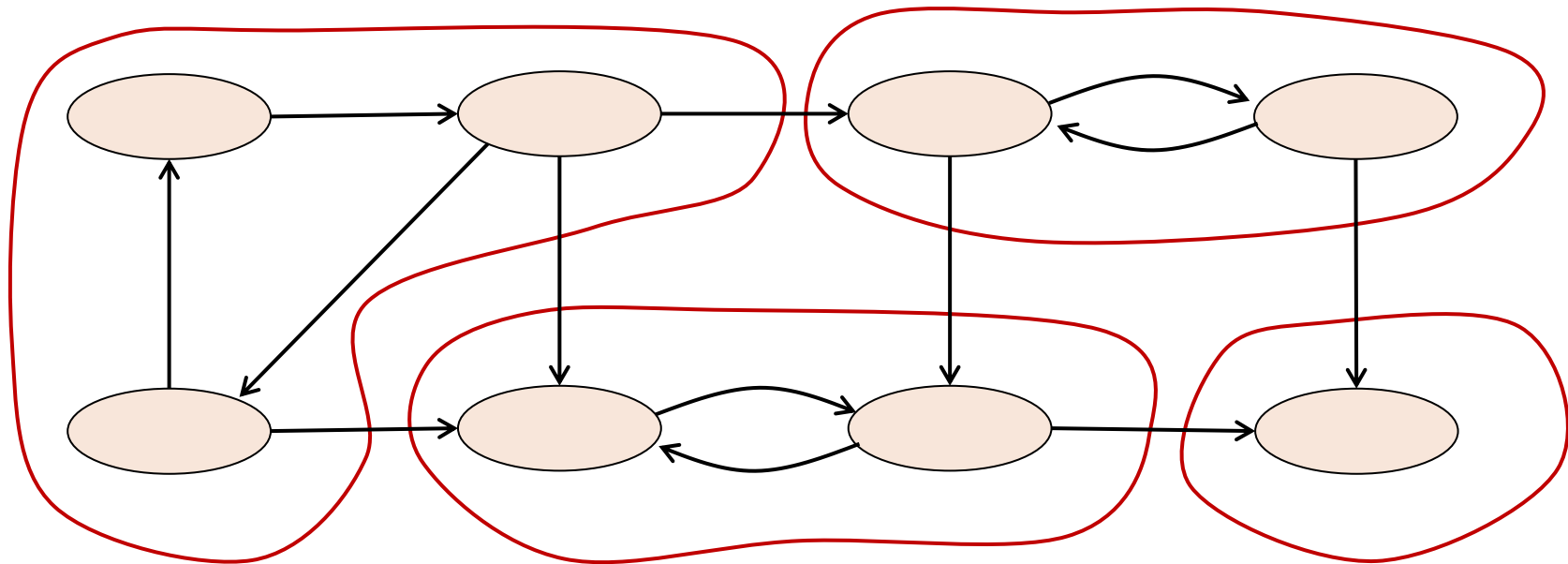
Δοθέντος ενός κατευθυνόμενου γραφήματος $G = (V, E)$, ορίζουμε το **αντίστροφο γράφημα του G** (transpose graph), το συμβολίζουμε $G^T = (V^T, E^T)$, ως εξής:

$$V^T = V \quad \text{και} \quad E^T = \{\langle a, b \rangle \mid \langle b, a \rangle \in E\}$$



Υπολογισμός SCC

Δοθέντος ενός κατευθυνόμενου γραφήματος G θέλουμε να **υπολογίσουμε** όλες τις **ισχυρές συνεκτικές συνιστώσες** του G .



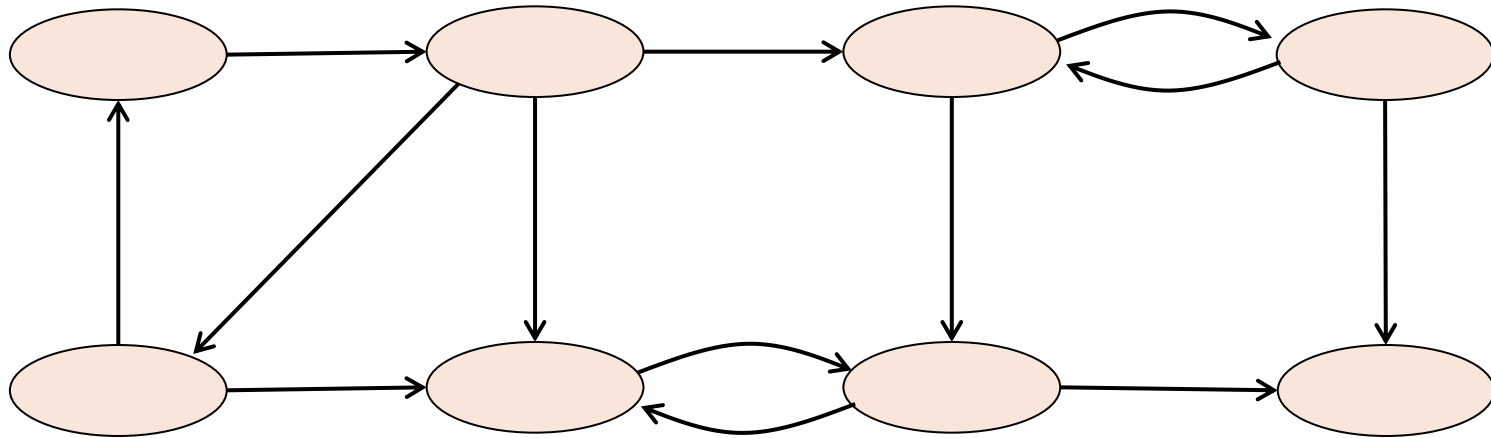
Υπολογισμός SCC

Αλγόριθμος SCC(G)

1. DFS(G) και υπολογισμός $f[u]$, $\forall u \in V(G)$
2. Υπολογισμός του αντίστροφου γραφήματος G^T
3. DFS(G^T), παίρνοντας του κόμβους
σε φθίνουσα τάξη ως προς $f[u]$
4. Υπολογισμός του DFS-δάσους
(κάθε DFS-tree του DFS-δάσους ισοδυναμεί με ένα SCC του G)

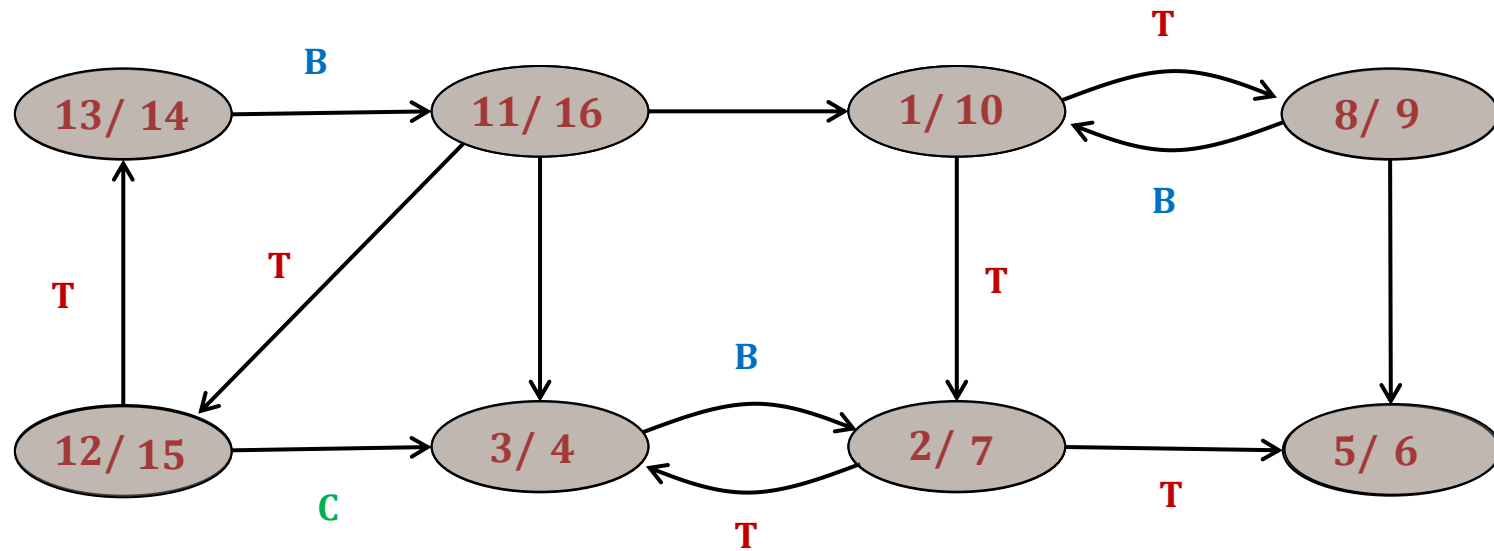
end.

Παράδειγμα Εκτέλεσης του Αλγόριθμου SCC

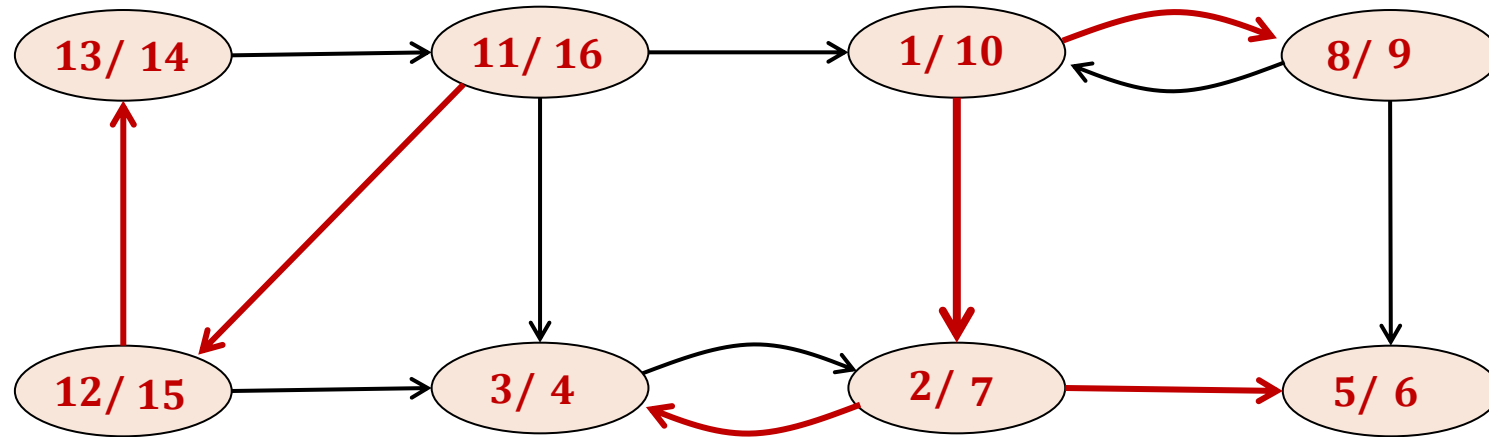


Έστω G ένα κατευθυνόμενο γράφημα

Βήμα 1: DFS διέλευση του G

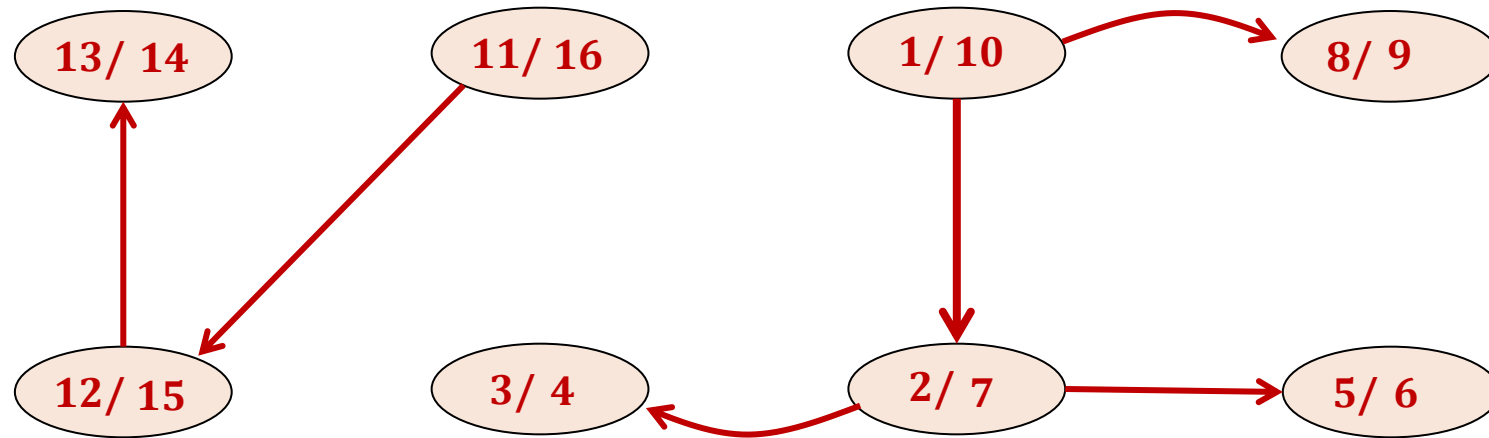


Βήμα 1: DFS διέλευση του G... DFS-δάσος



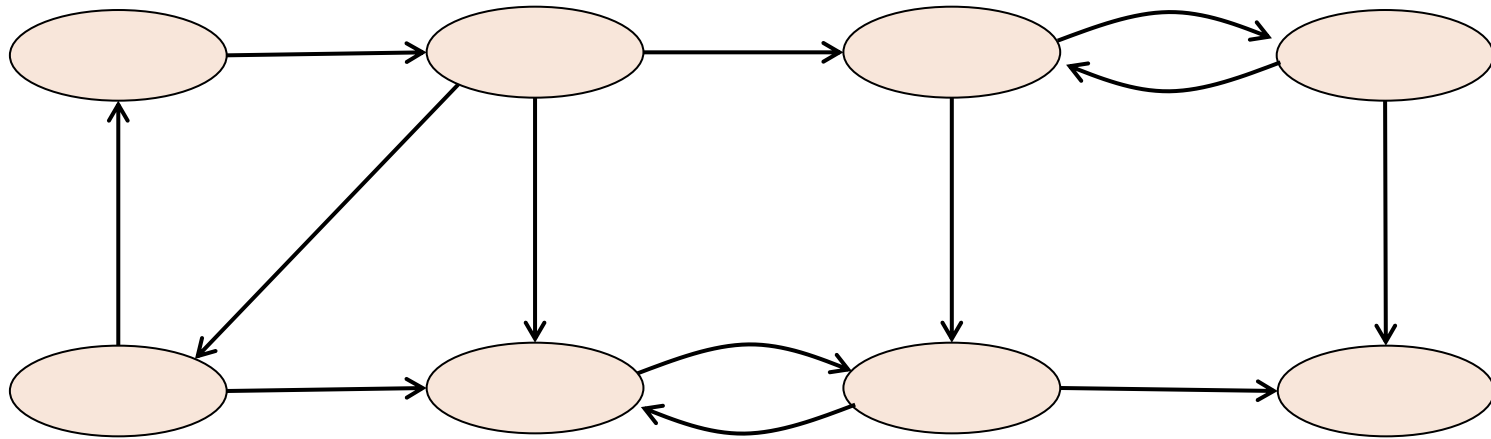
Ισχυρά Συνεκτικές Συνιστώσες

Βήμα 1: DFS διέλευση του G... DFS-δάσος



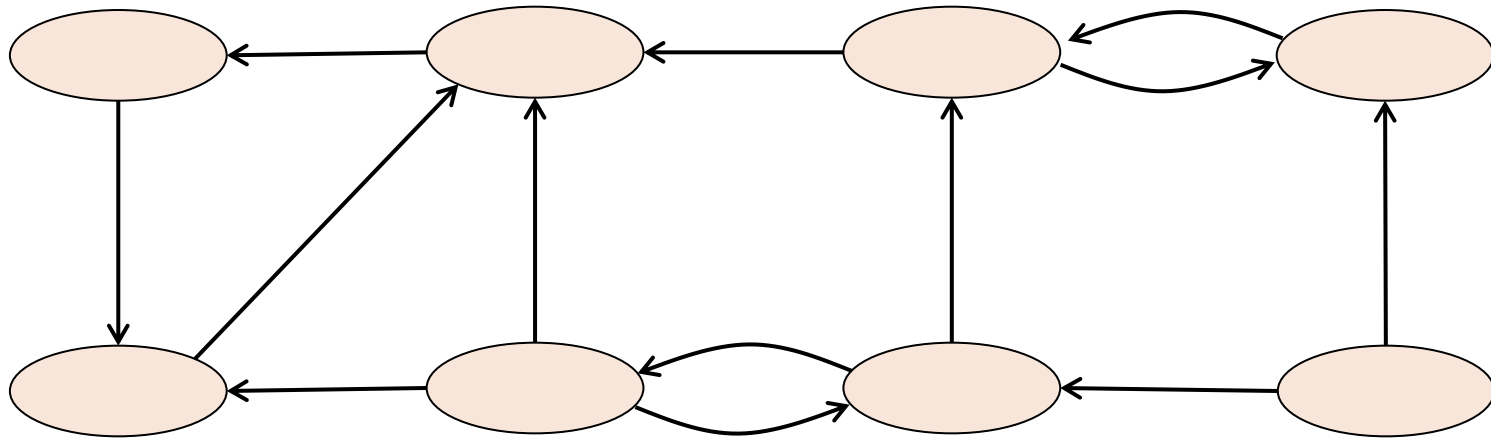


Βήμα 2: Αρχικό γράφημα G

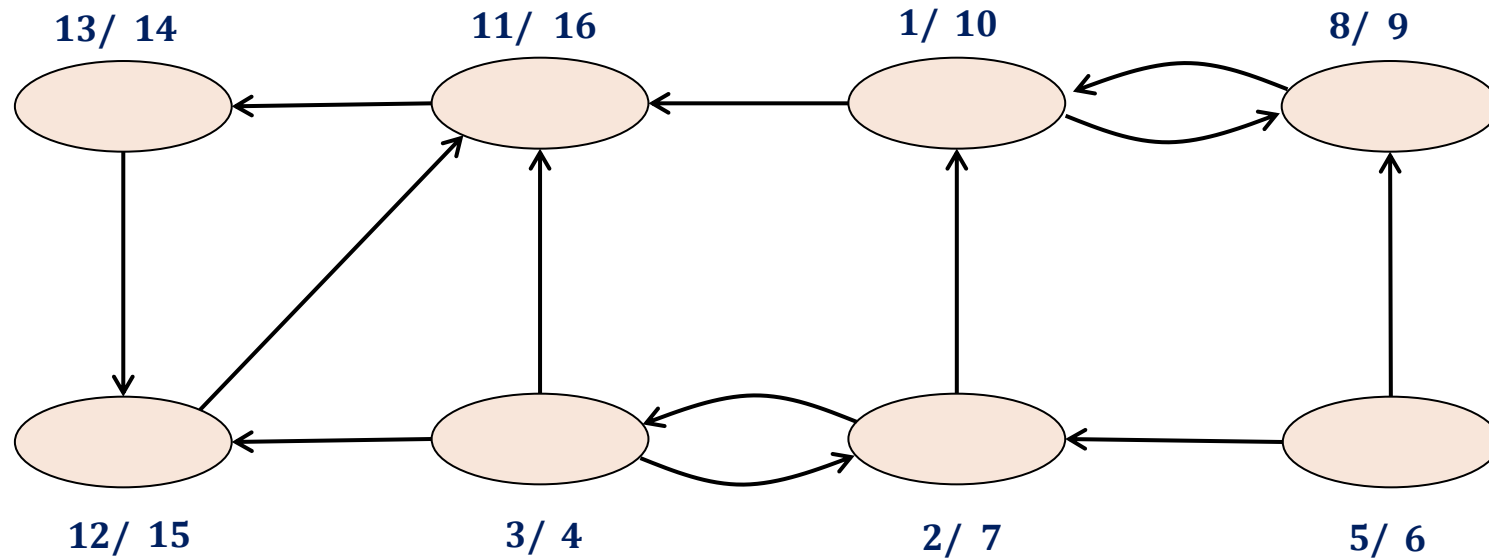




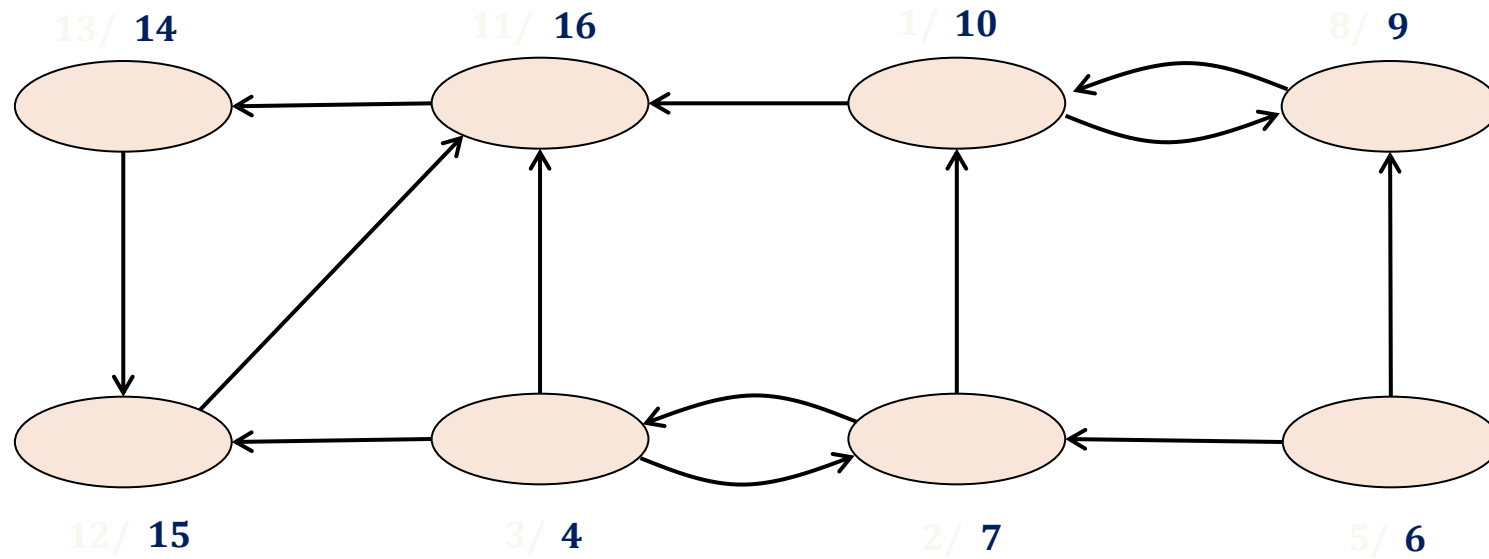
Βήμα 2: Το αντίστροφο γράφημα G^T



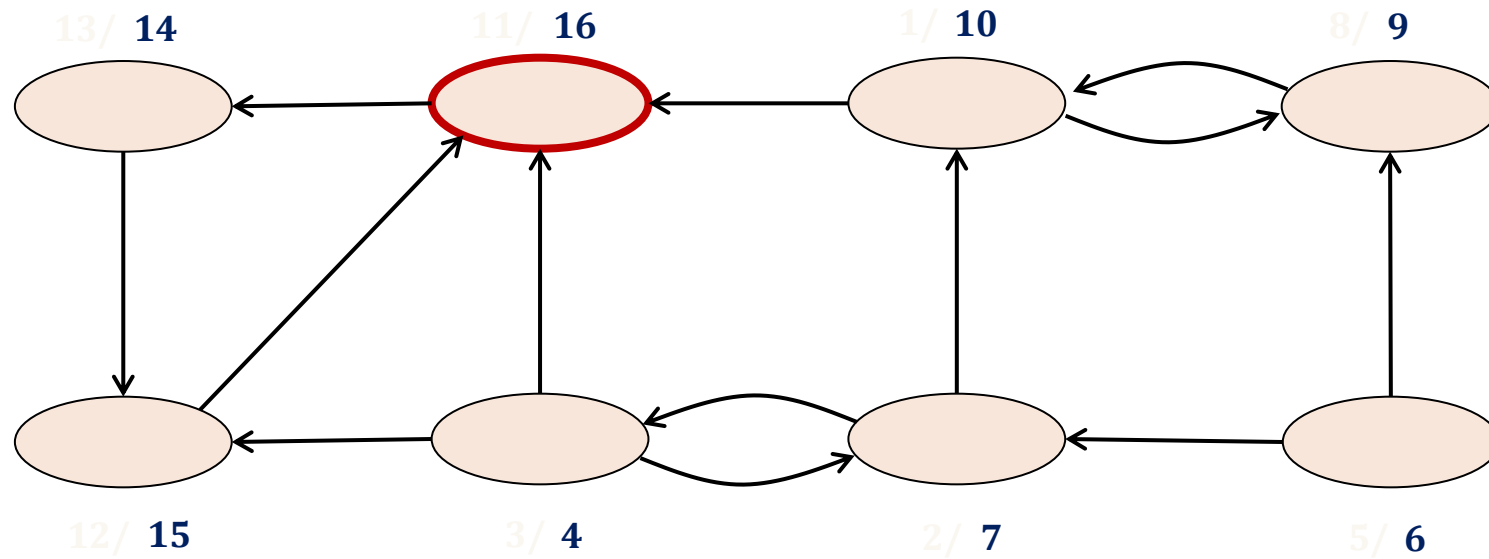
Βήμα 2: Το αντίστροφο γράφημα G^T



Βήμα 2: Το αντίστροφο γράφημα G^T

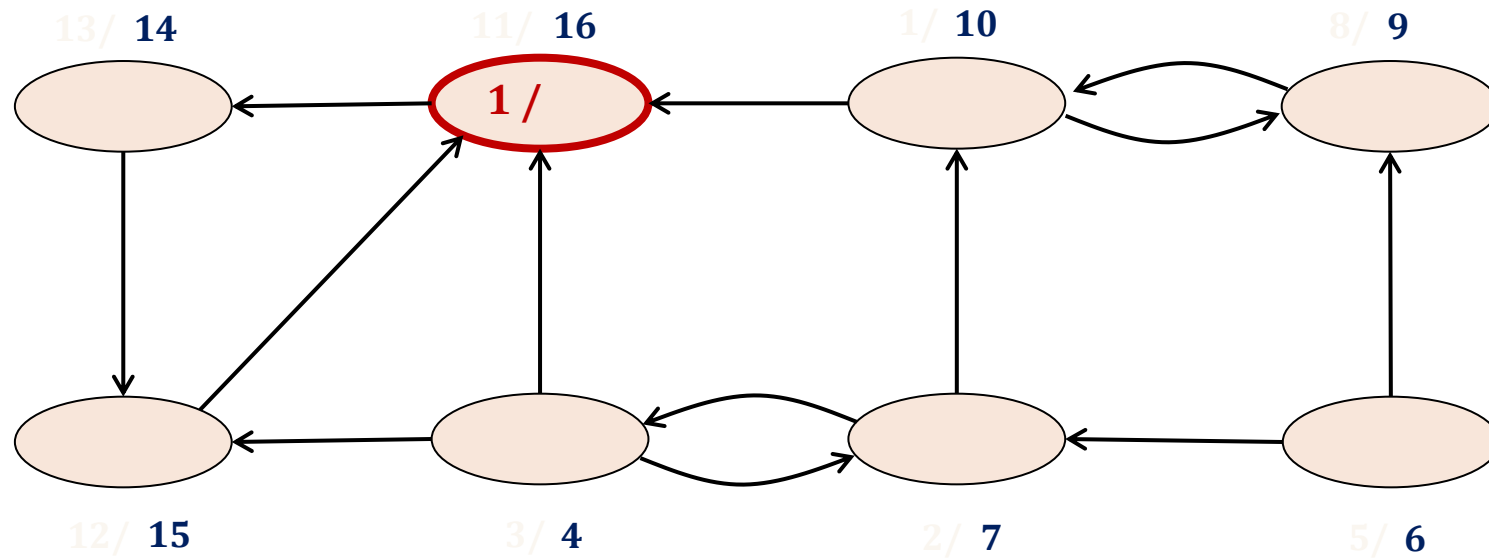


Βήμα 3: DFS διέλευση του G^T

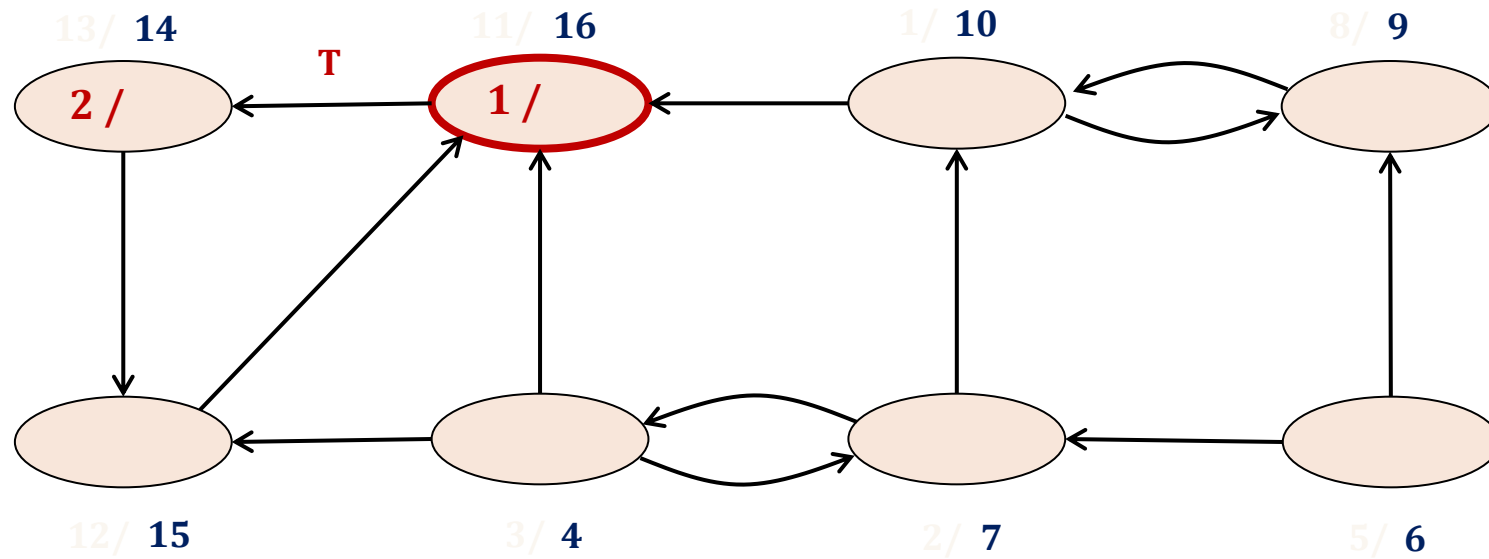


Ισχυρά Συνεκτικές Συνιστώσες

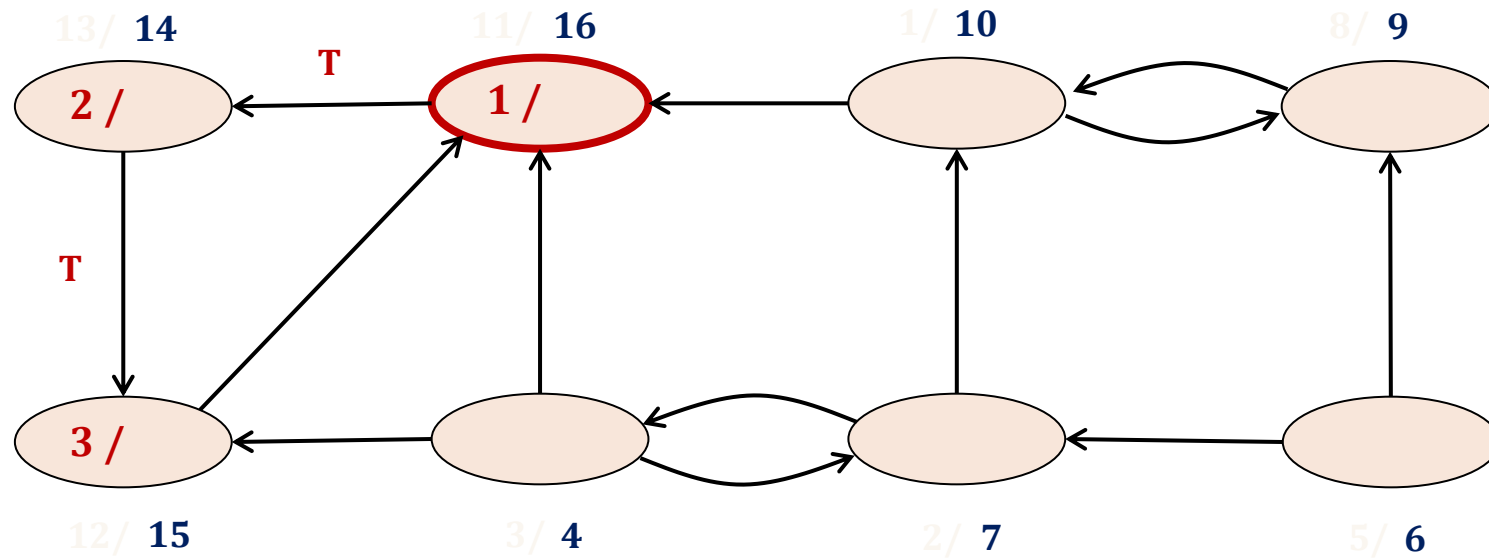
Βήμα 3: DFS διέλευση του G^T



Βήμα 3: DFS διέλευση του G^T

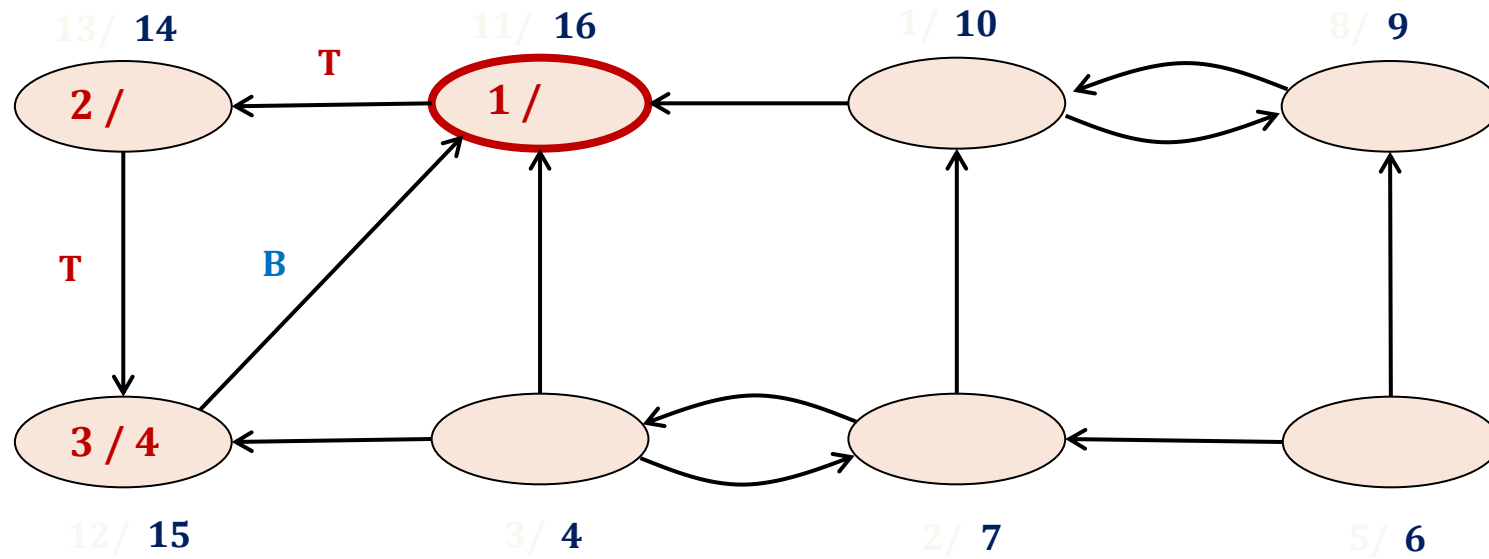


Βήμα 3: DFS διέλευση του G^T

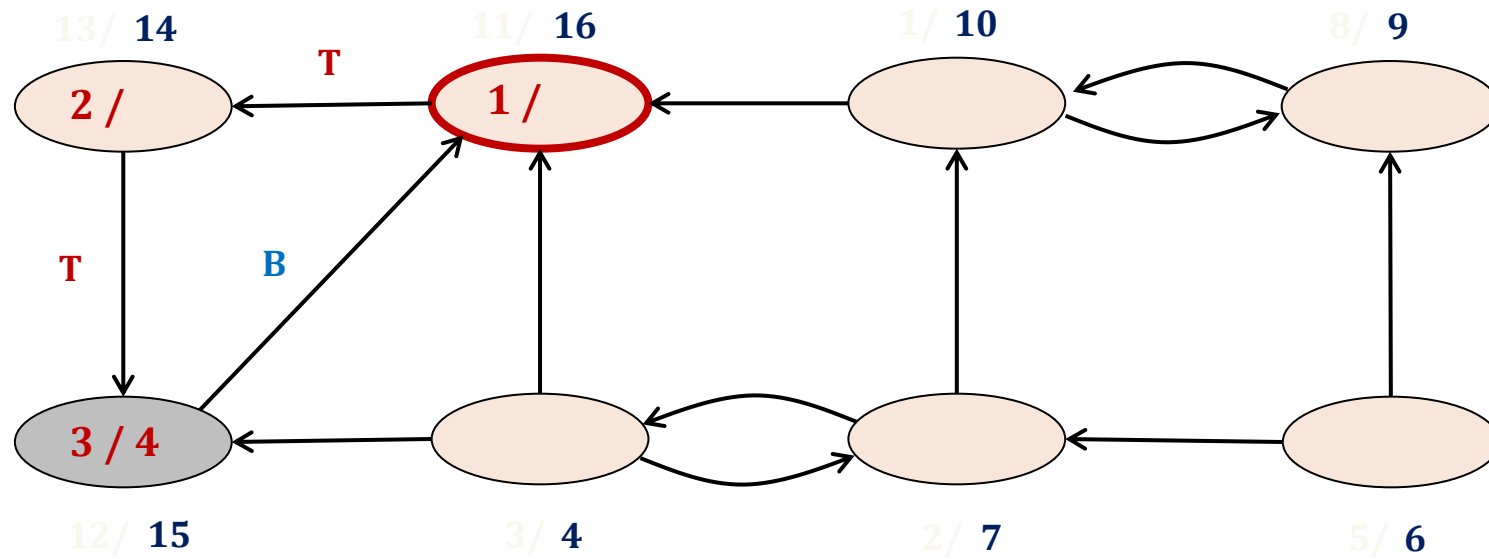


Ισχυρά Συνεκτικές Συνιστώσες

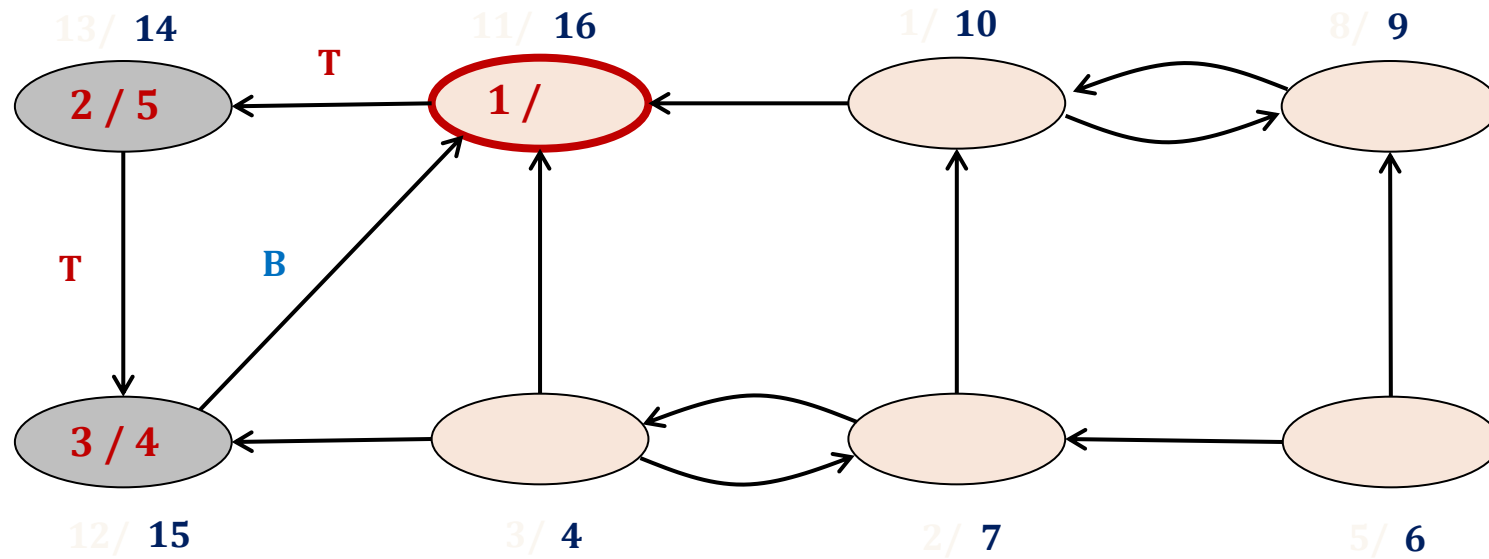
Βήμα 3: DFS διέλευση του G^T



Βήμα 3: DFS διέλευση του G^T

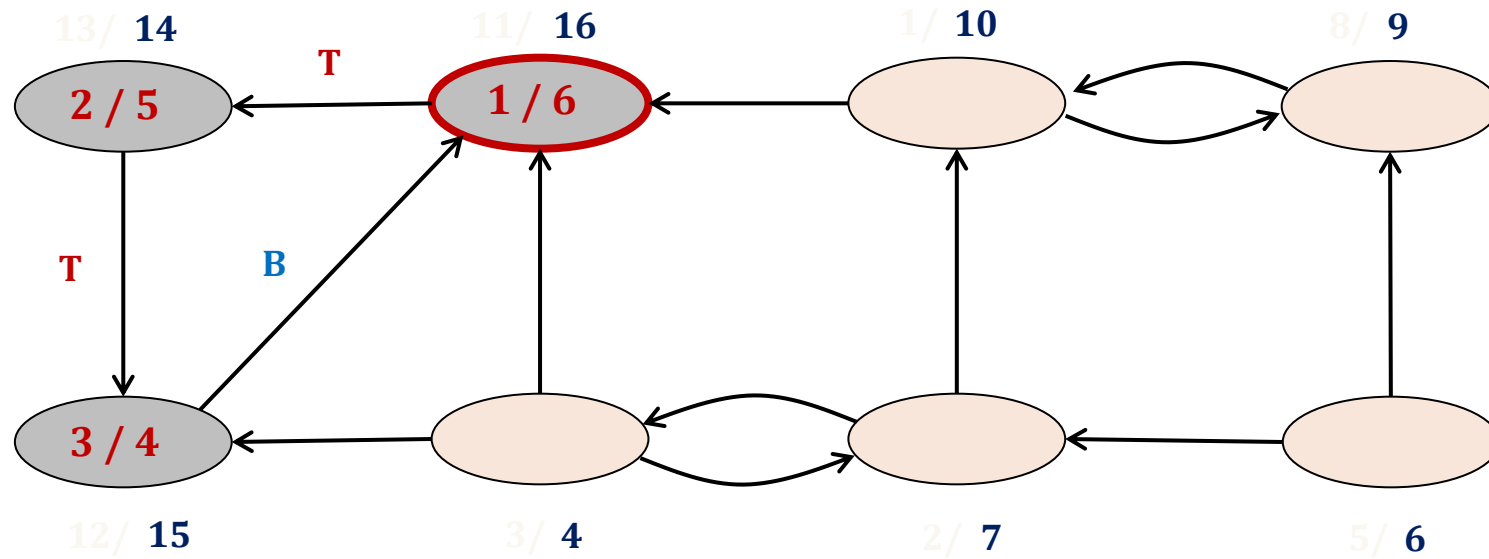


Βήμα 3: DFS διέλευση του G^T

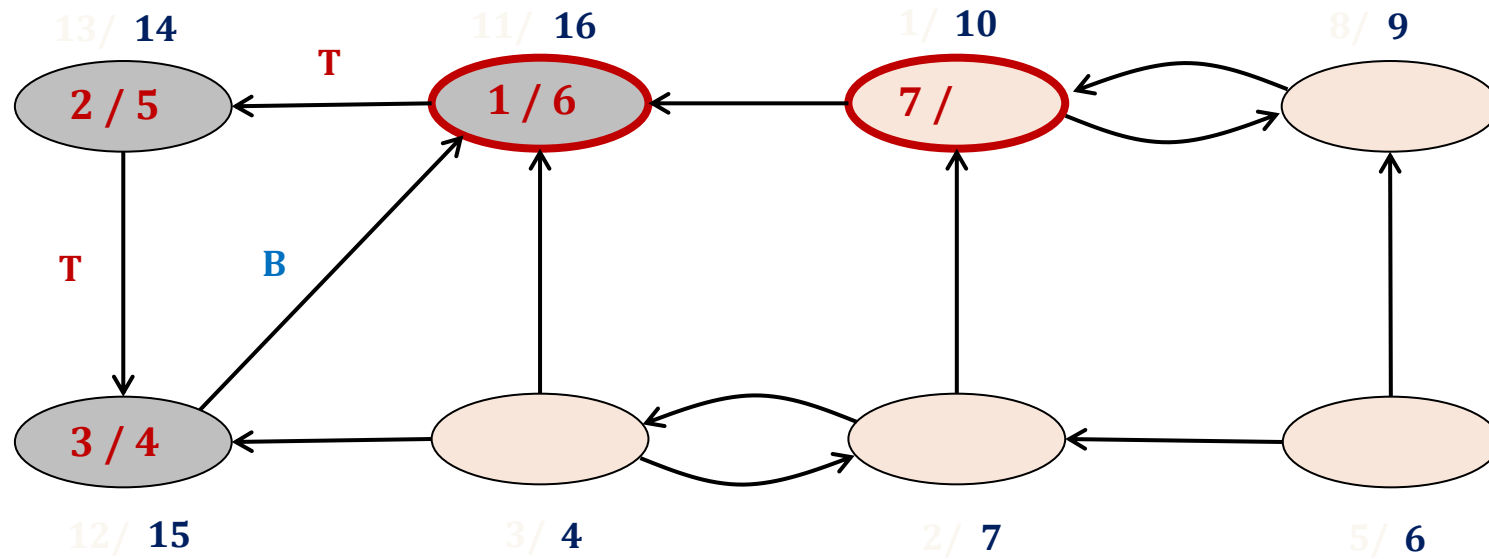


Ισχυρά Συνεκτικές Συνιστώσες

Βήμα 3: DFS διέλευση του G^T

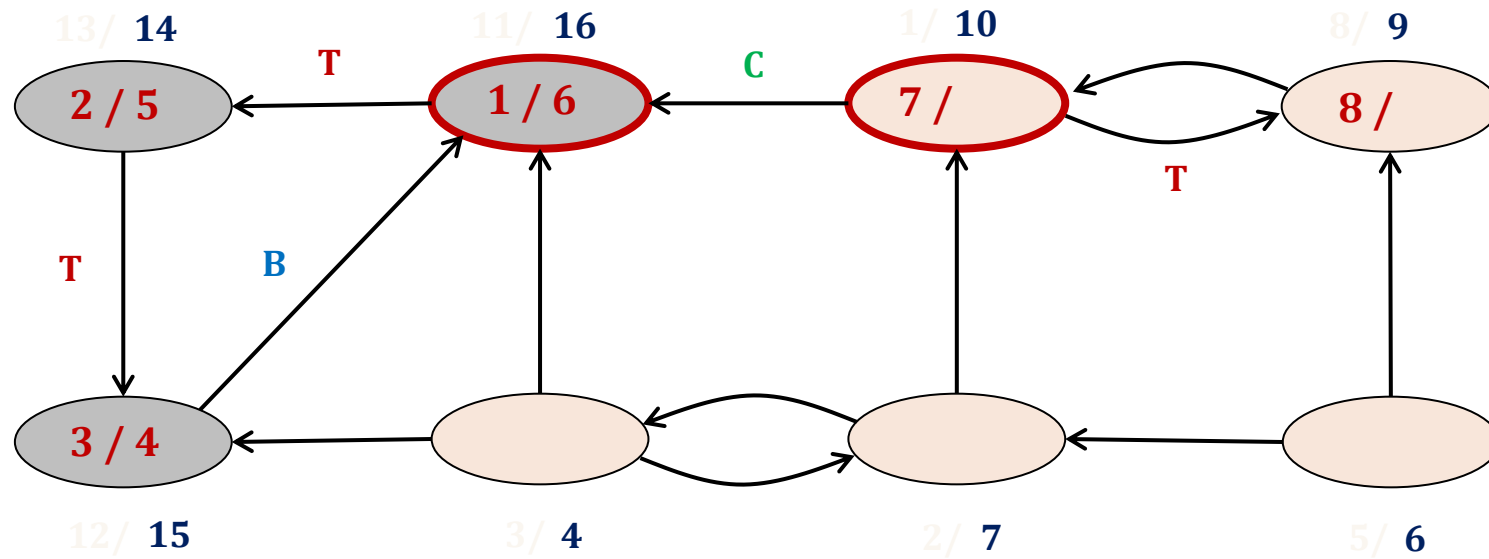


Βήμα 3: DFS διέλευση του G^T



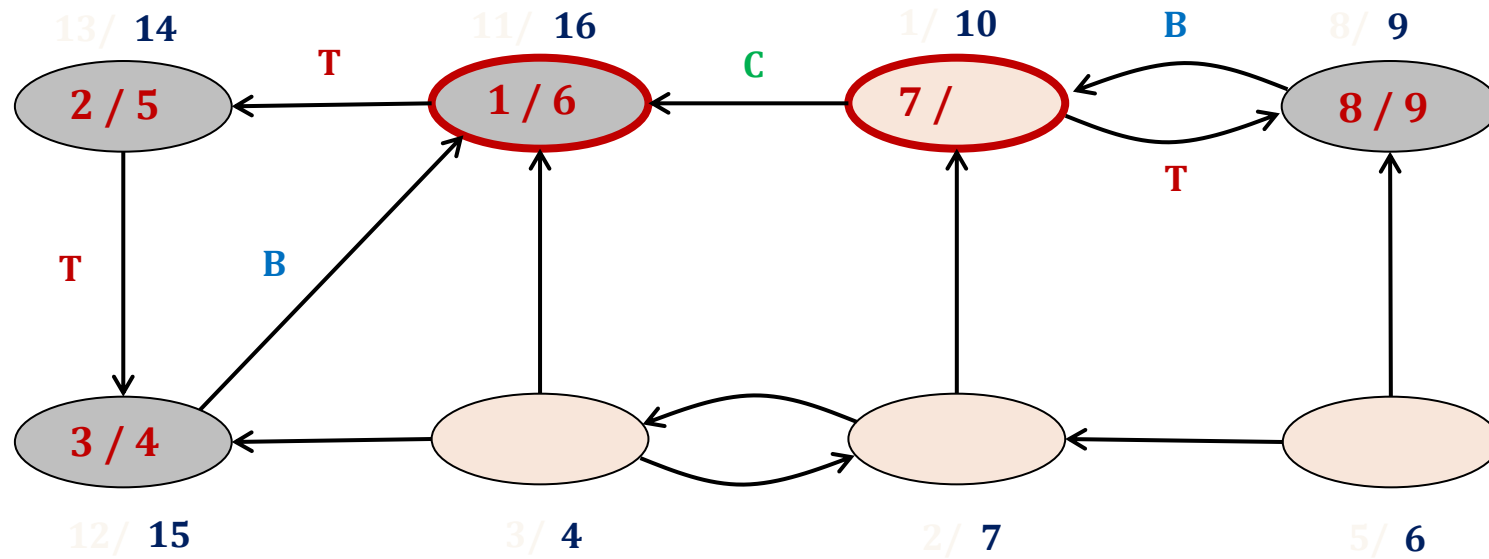
Ισχυρά Συνεκτικές Συνιστώσες

Βήμα 3: DFS διέλευση του G^T



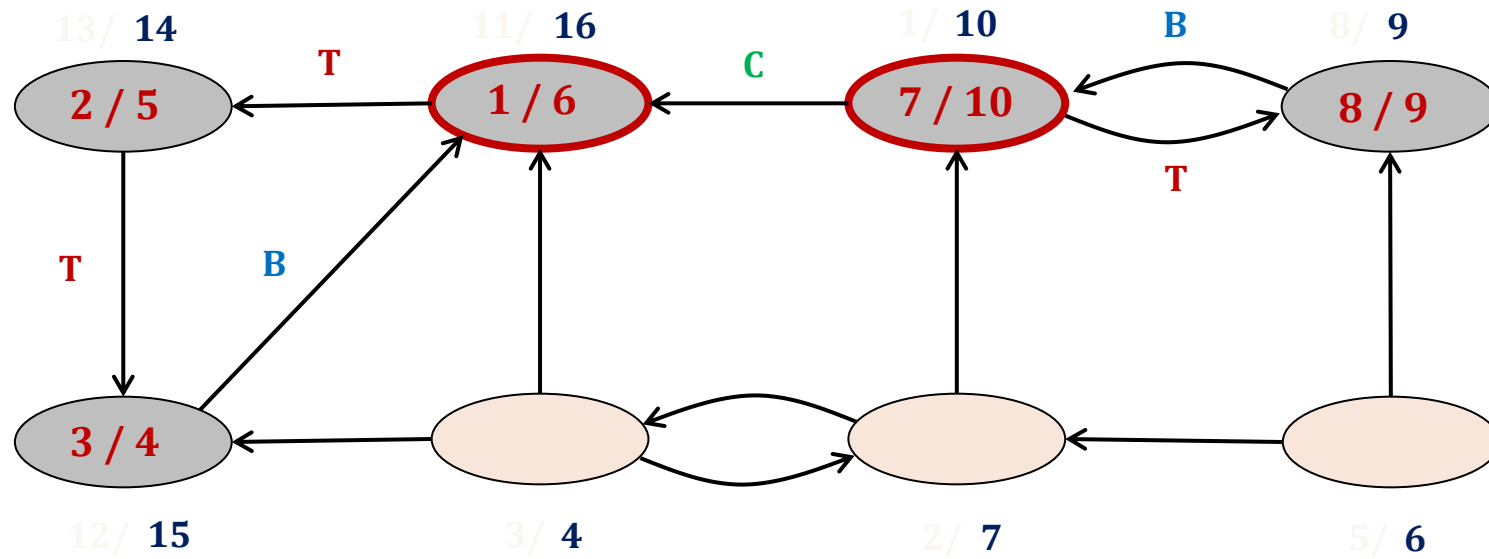
Ισχυρά Συνεκτικές Συνιστώσες

Βήμα 3: DFS διέλευση του G^T



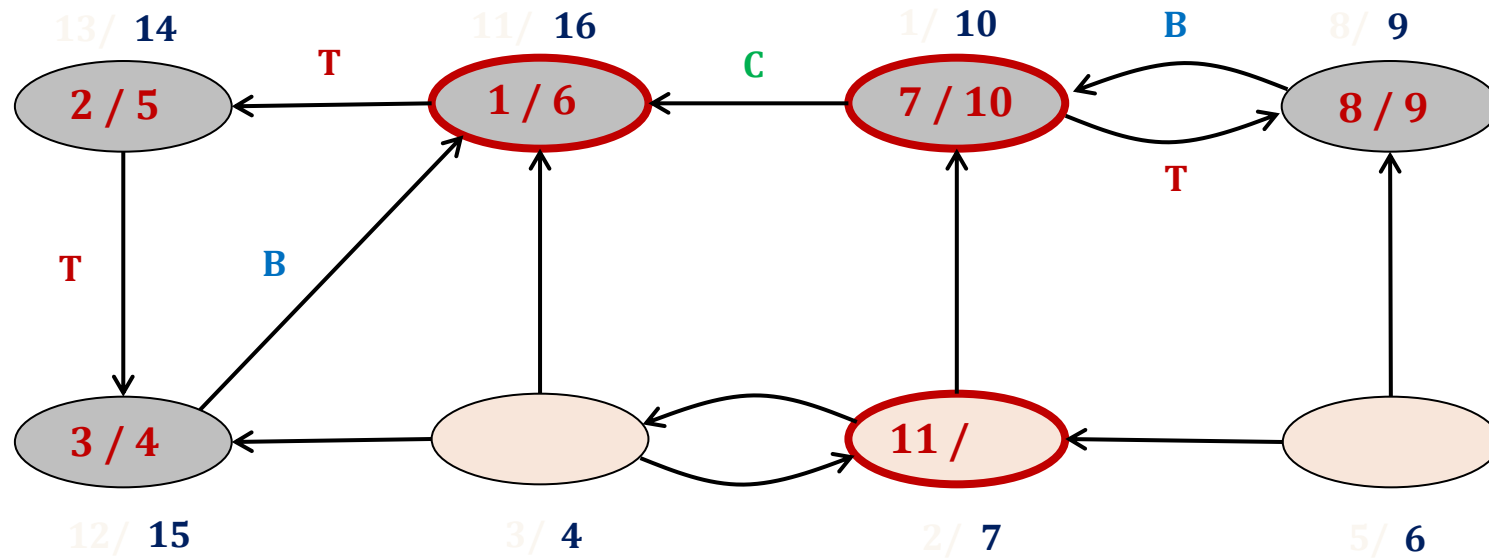
Ισχυρά Συνεκτικές Συνιστώσες

Βήμα 3: DFS διέλευση του G^T



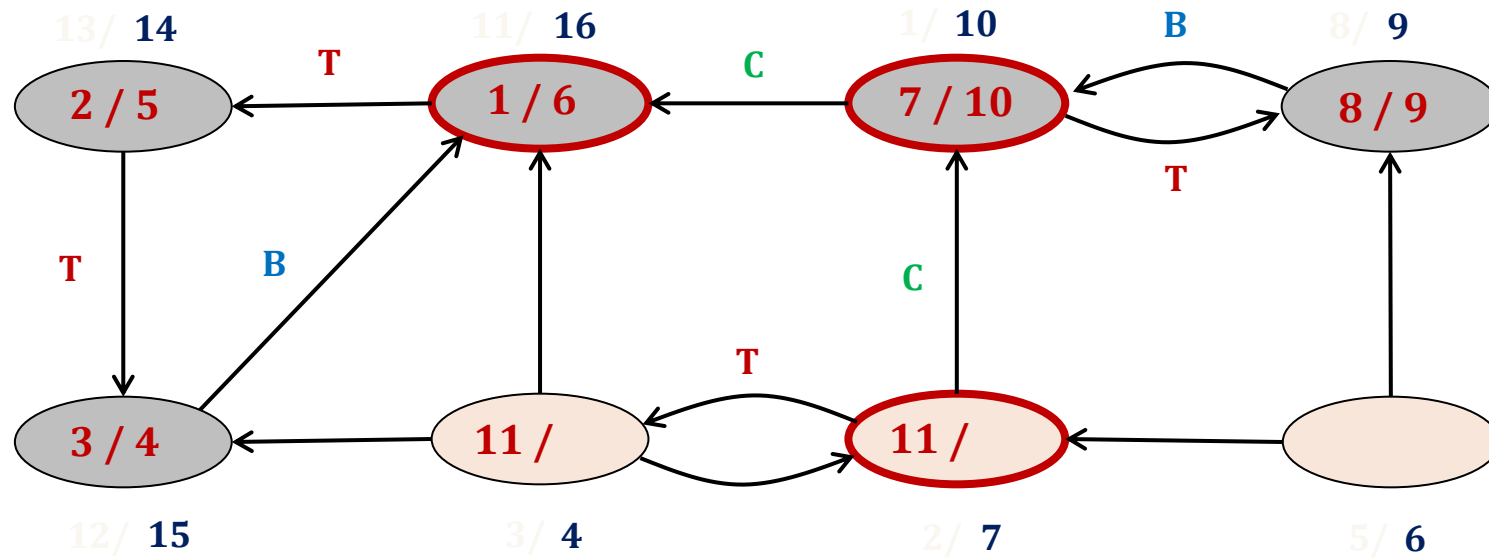
Ισχυρά Συνεκτικές Συνιστώσες

Βήμα 3: DFS διέλευση του G^T



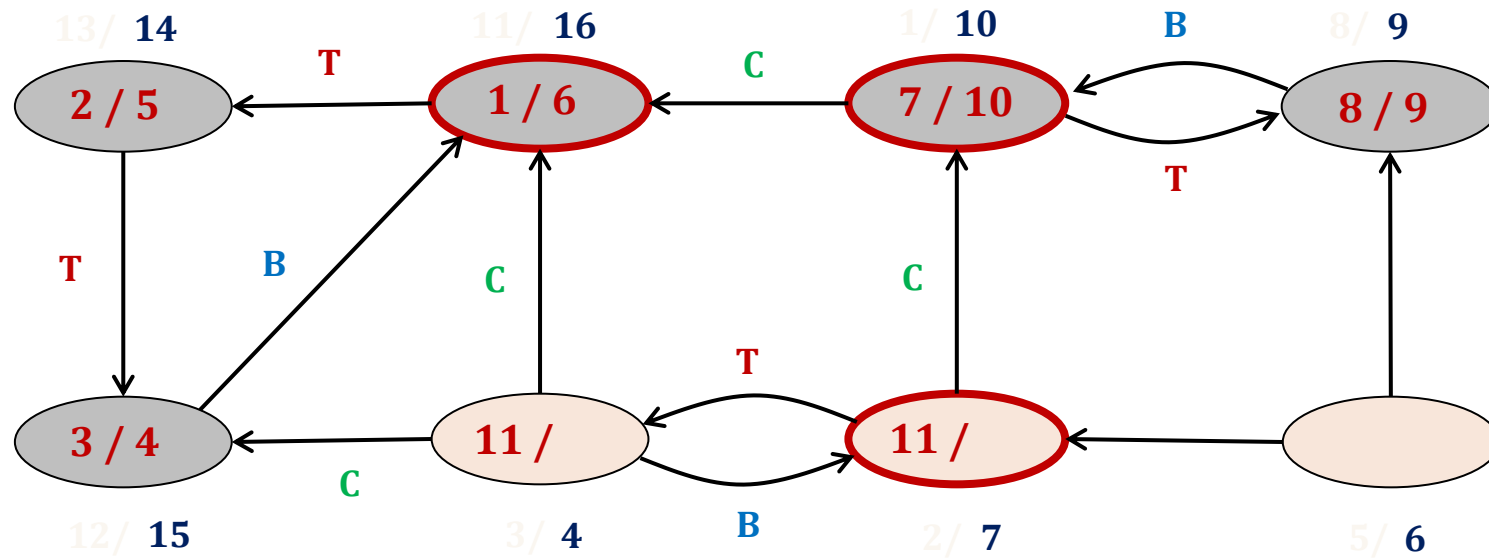
Ισχυρά Συνεκτικές Συνιστώσες

Βήμα 3: DFS διέλευση του G^T



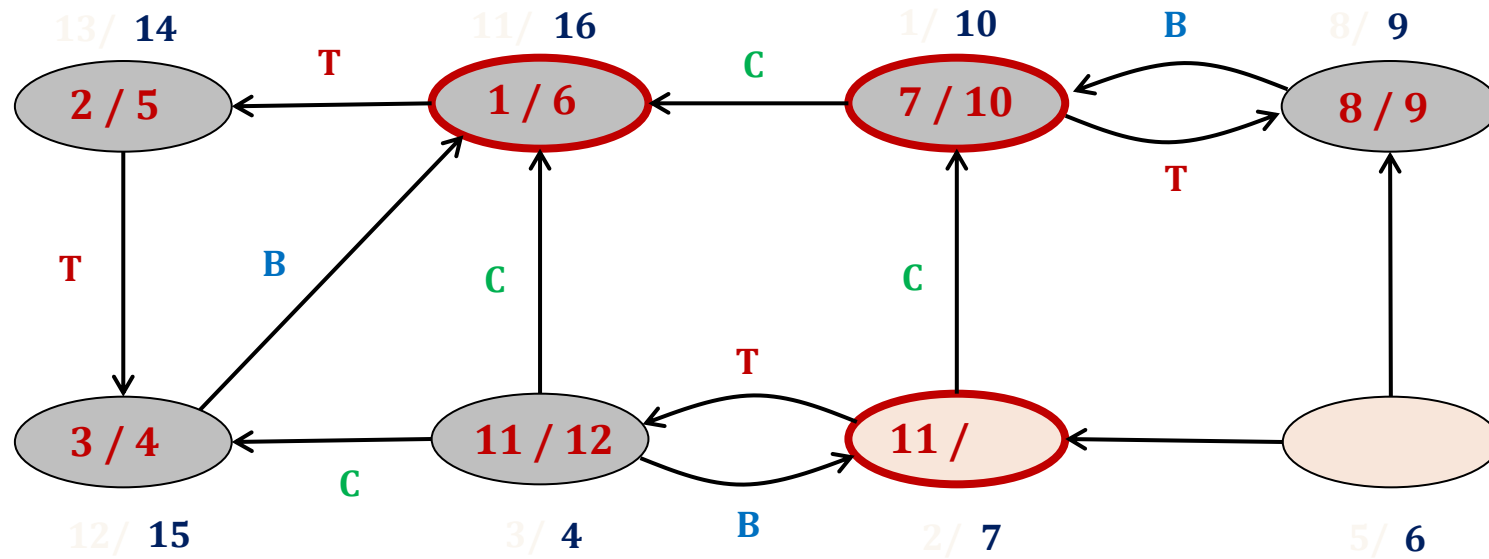
Ισχυρά Συνεκτικές Συνιστώσες

Βήμα 3: DFS διέλευση του G^T



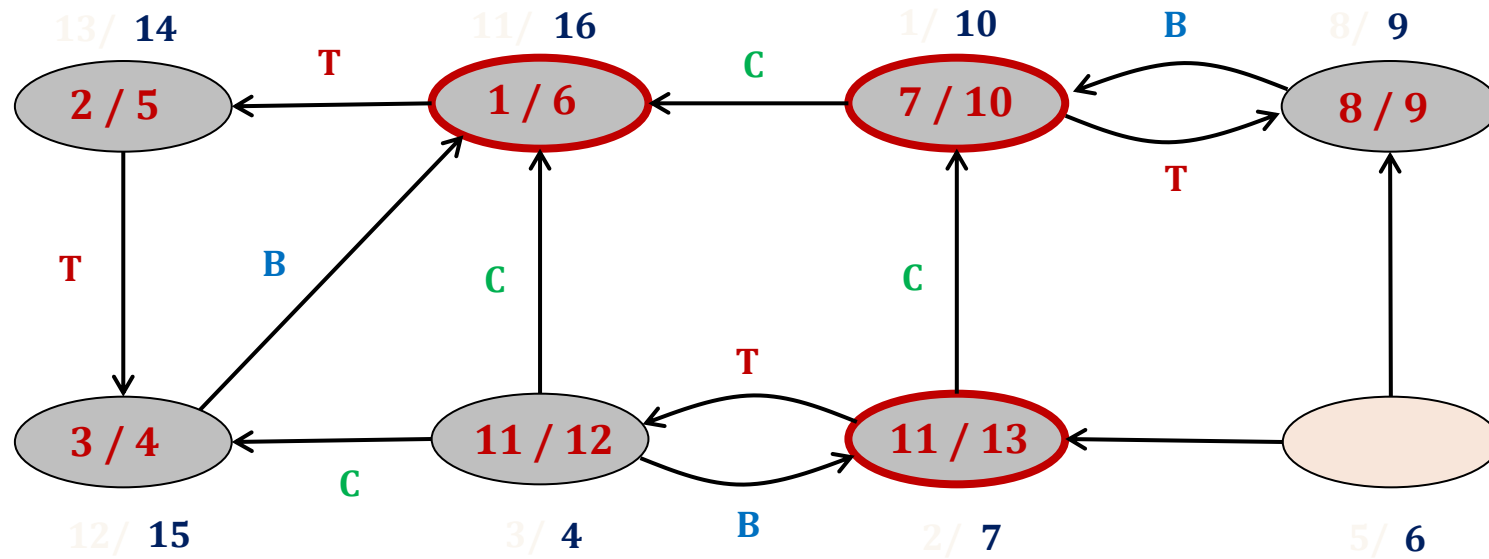
Ισχυρά Συνεκτικές Συνιστώσες

Βήμα 3: DFS διέλευση του G^T



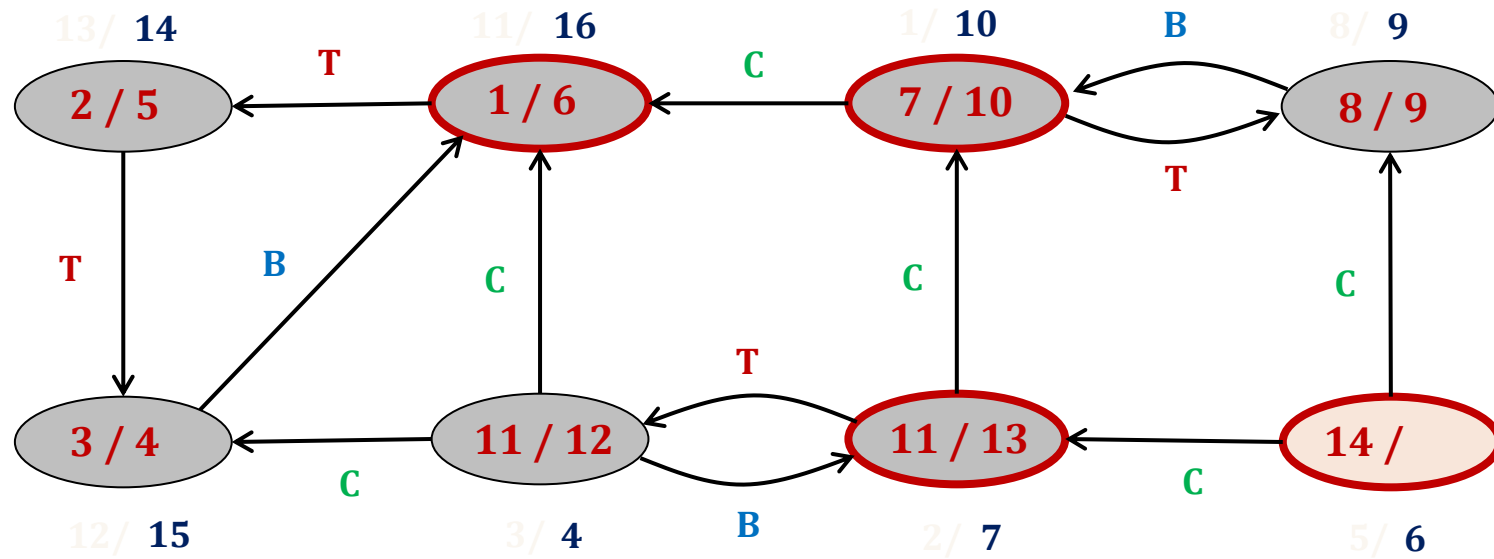
Ισχυρά Συνεκτικές Συνιστώσες

Βήμα 3: DFS διέλευση του G^T



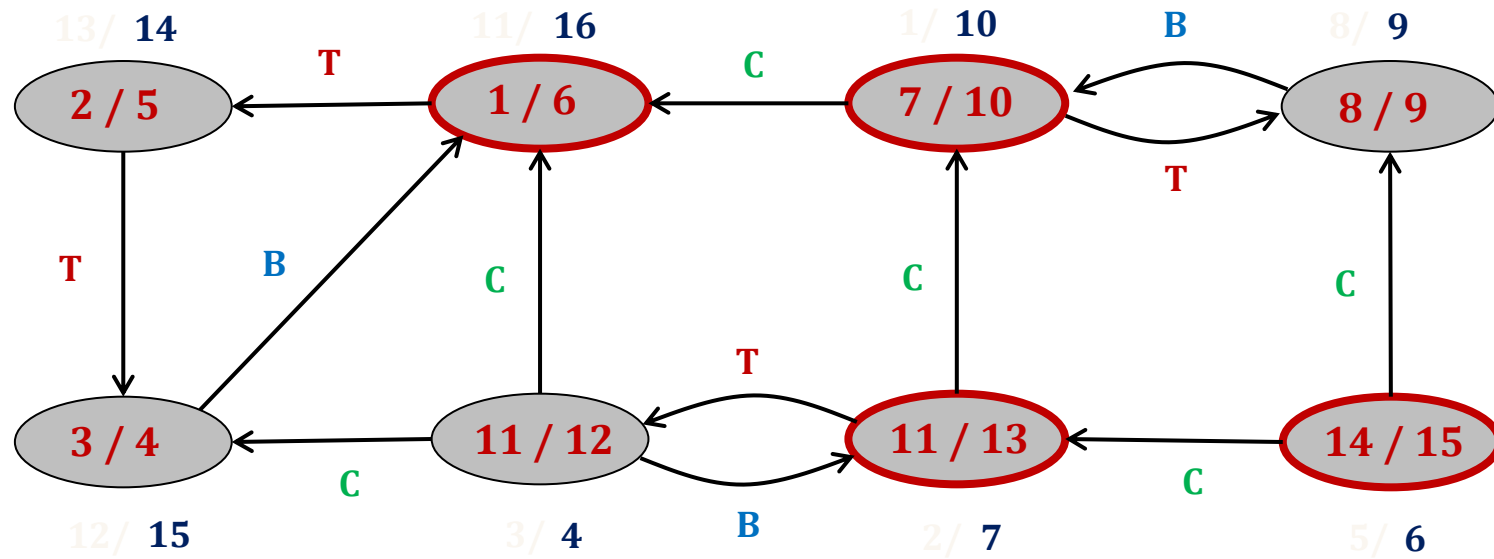
Ισχυρά Συνεκτικές Συνιστώσες

Βήμα 3: DFS διέλευση του G^T

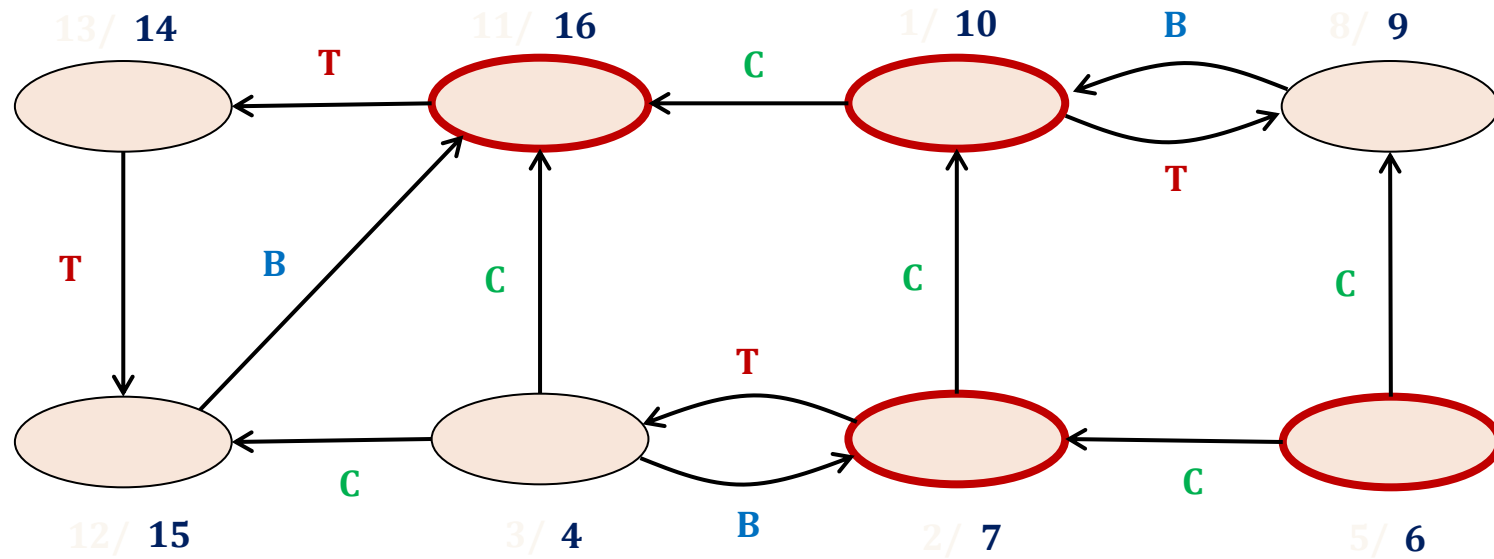


Ισχυρά Συνεκτικές Συνιστώσες

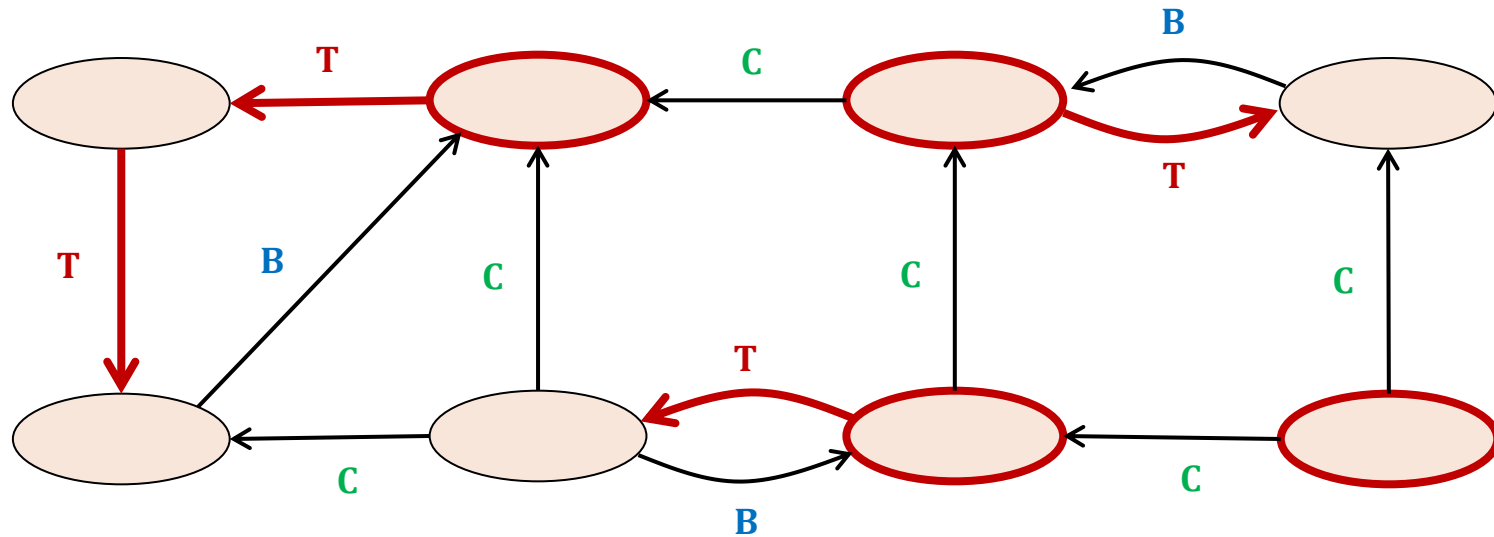
Βήμα 3: DFS διέλευση του G^T



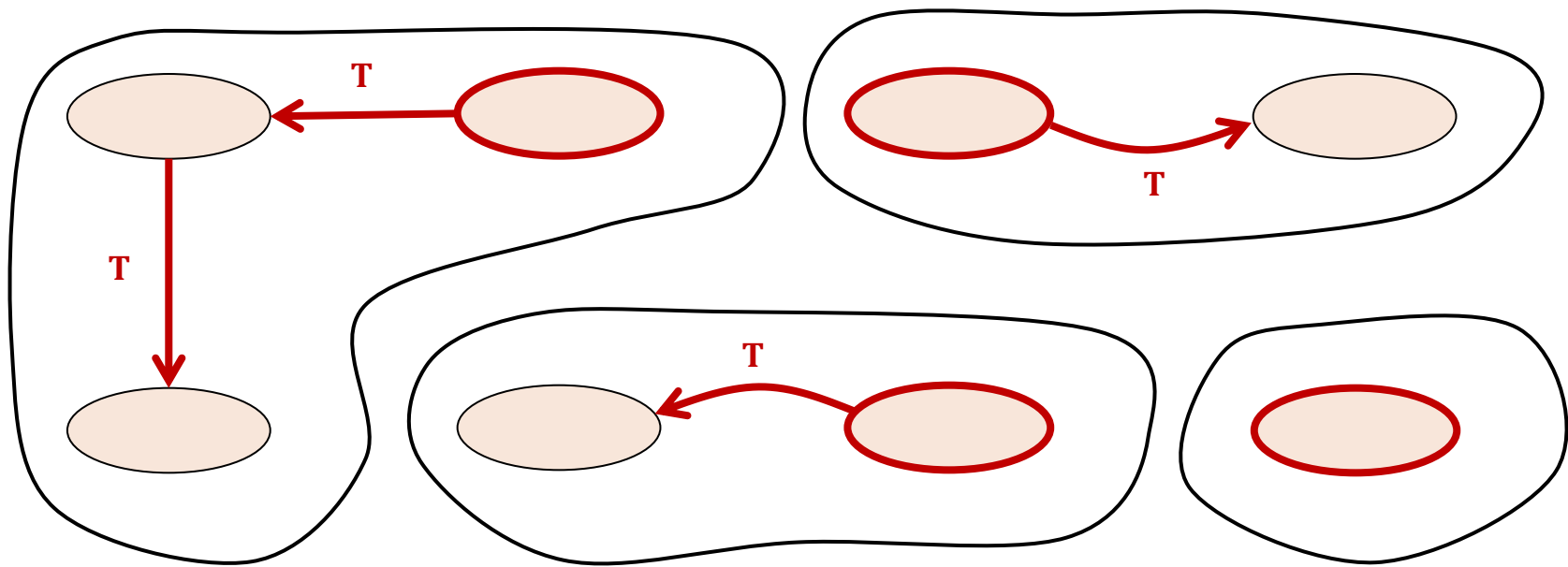
Βήμα 3: DFS διέλευση του G^T



Βήμα 4: Υπολογισμός του DFS-δάσους

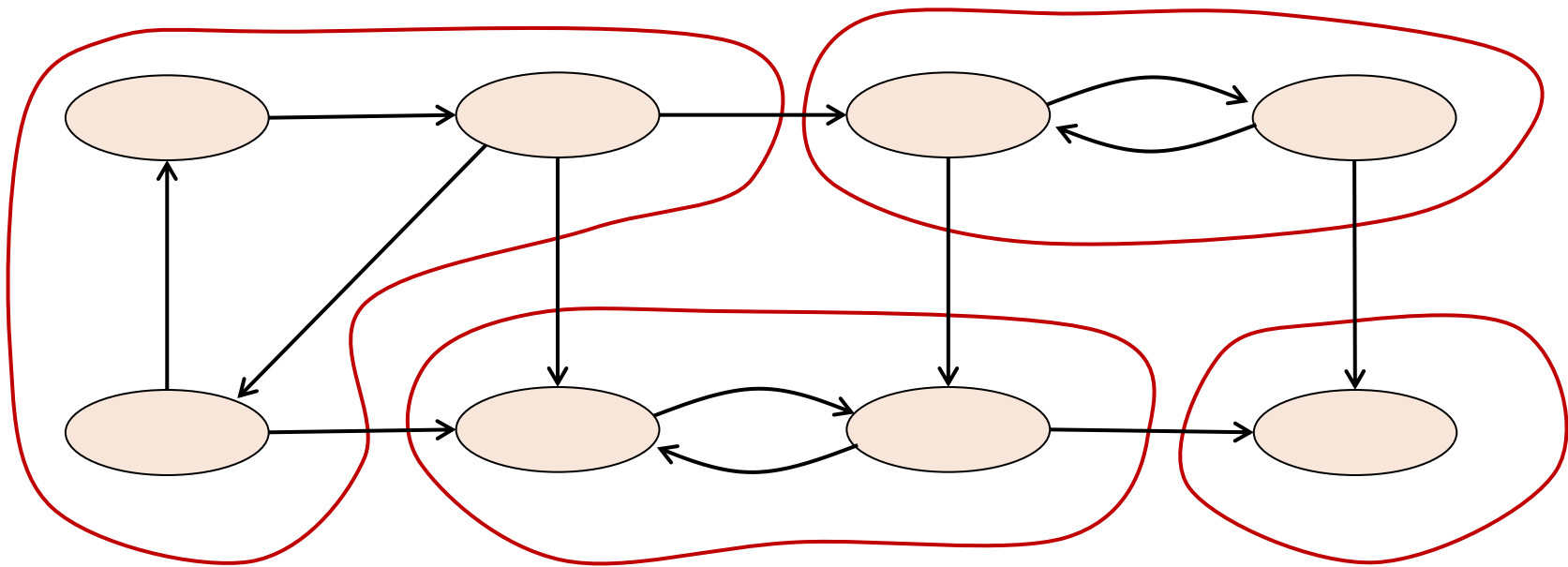


Βήμα 4: Υπολογισμός του DFS-δάσους



Ισχυρά Συνεκτικές Συνιστώσες

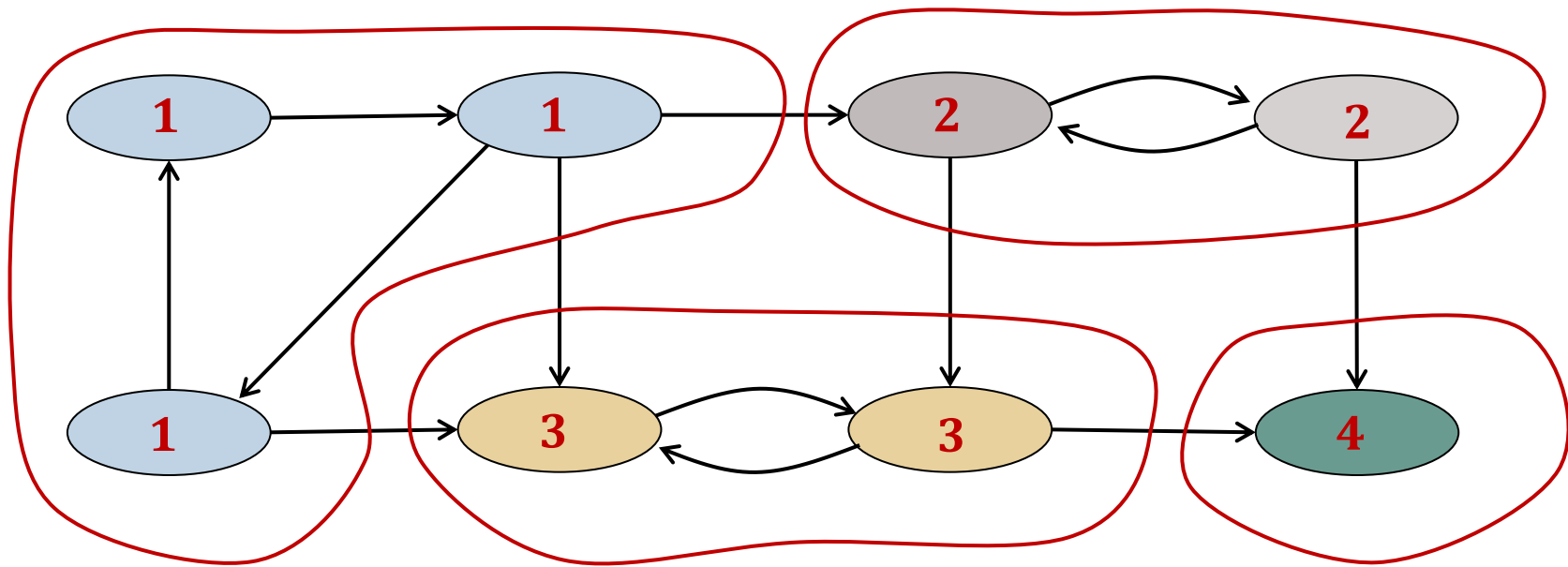
SCC του γραφήματος G



Οι Ισχυρές Συνεκτικές Συνιστώσες του γραφήματος G

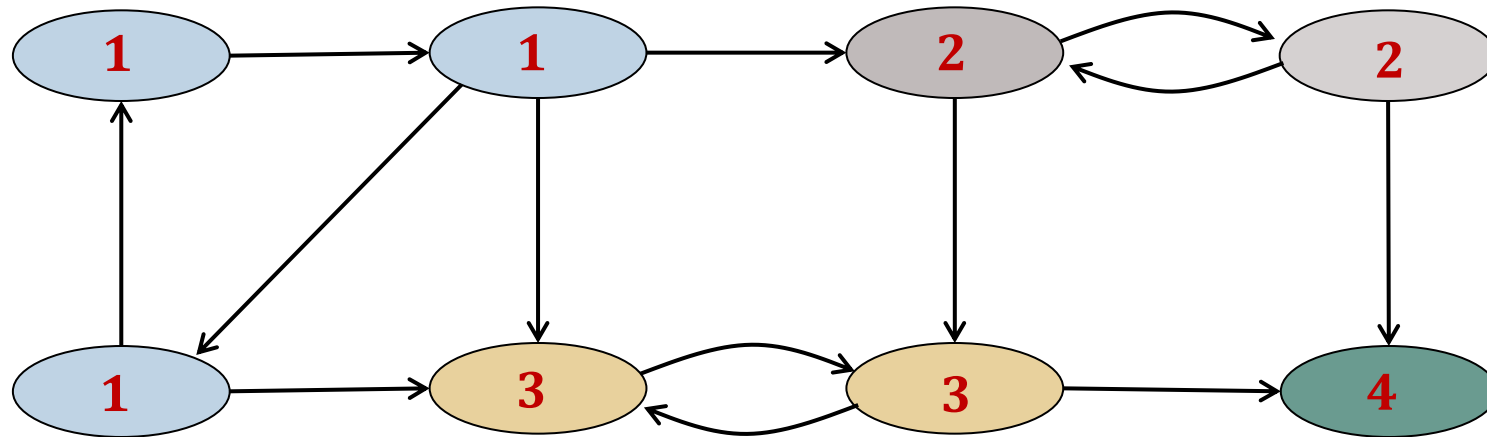
Ισχυρά Συνεκτικές Συνιστώσες

SCC του γραφήματος G



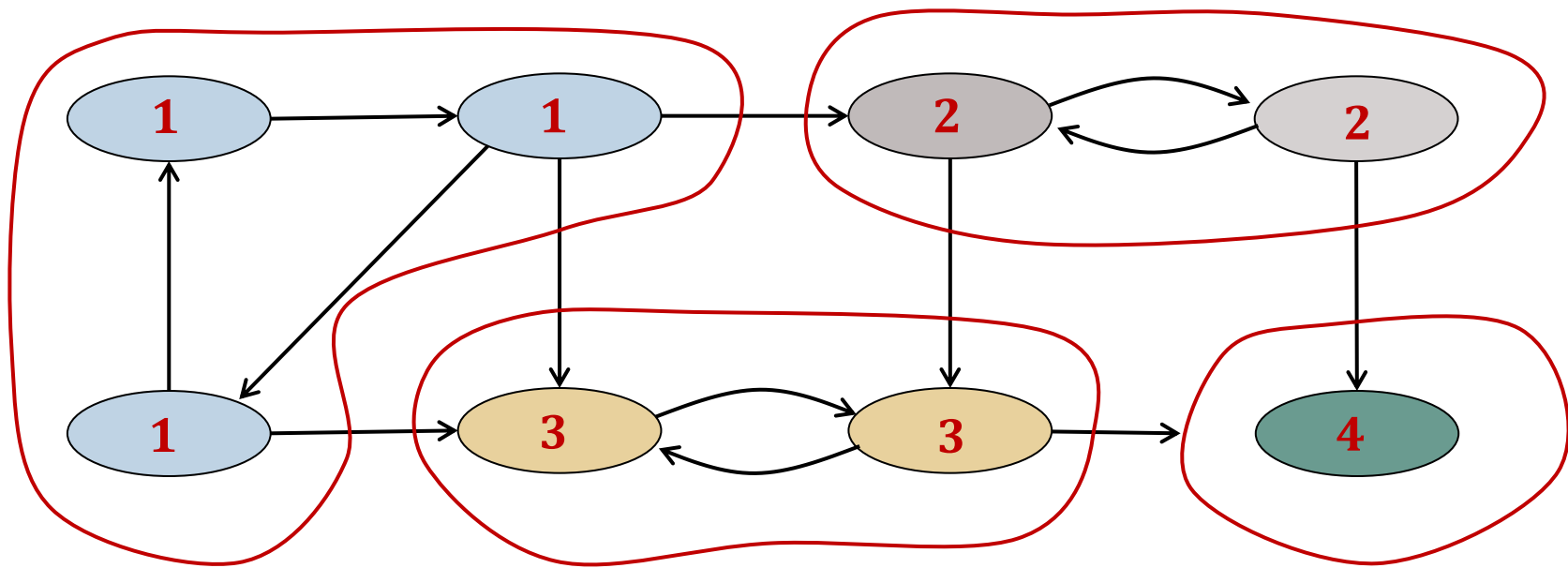
Οι Ισχυρές Συνεκτικές Συνιστώσες του γραφήματος G

SCC του γραφήματος G



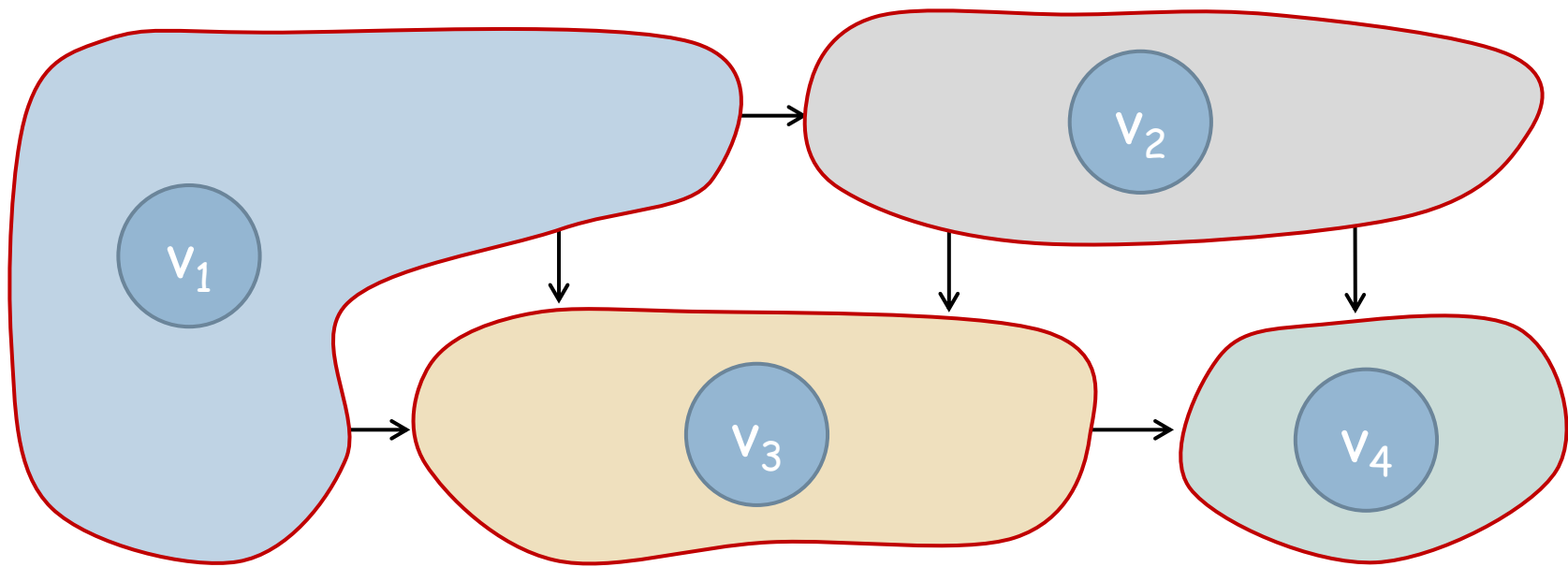
Οι Ισχυρές Συνεκτικές Συνιστώσες του γραφήματος G

Γράφημα Ισχυρών Συνιστωσών $H(G)$



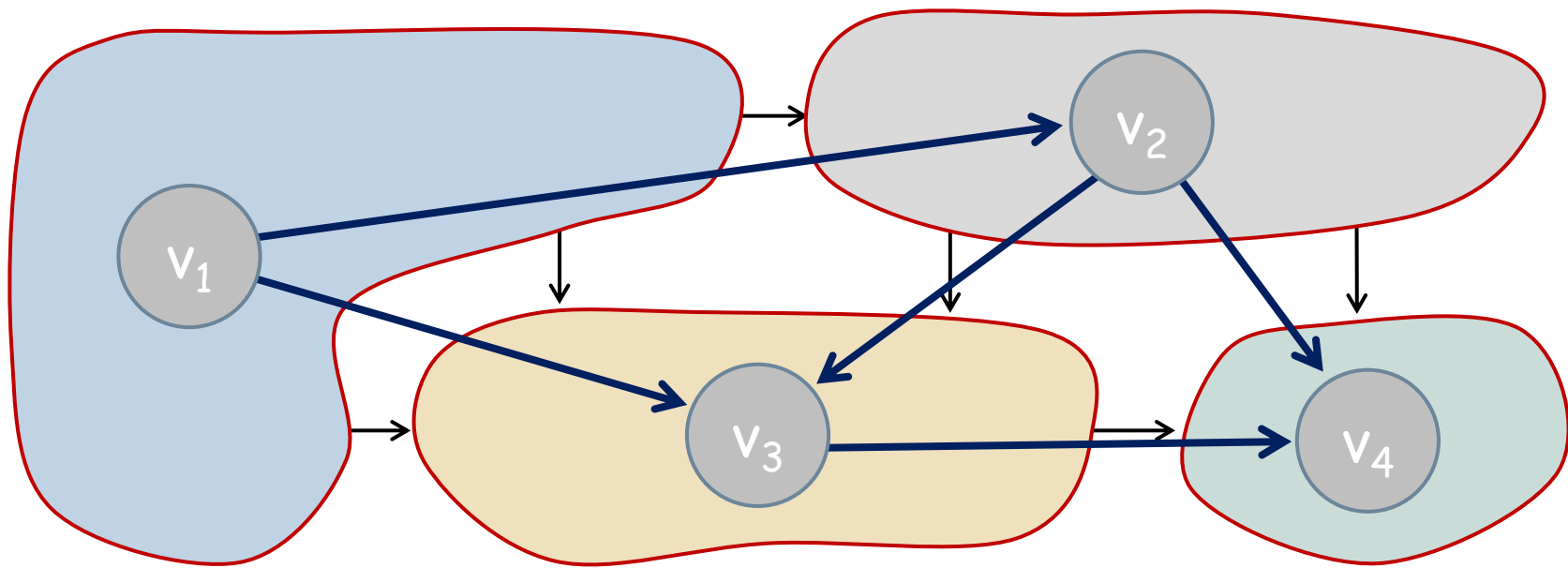
Οι Ισχυρές Συνεκτικές Συνιστώσες του γραφήματος G

Γράφημα Ισχυρών Συνιστωσών $H(G)$



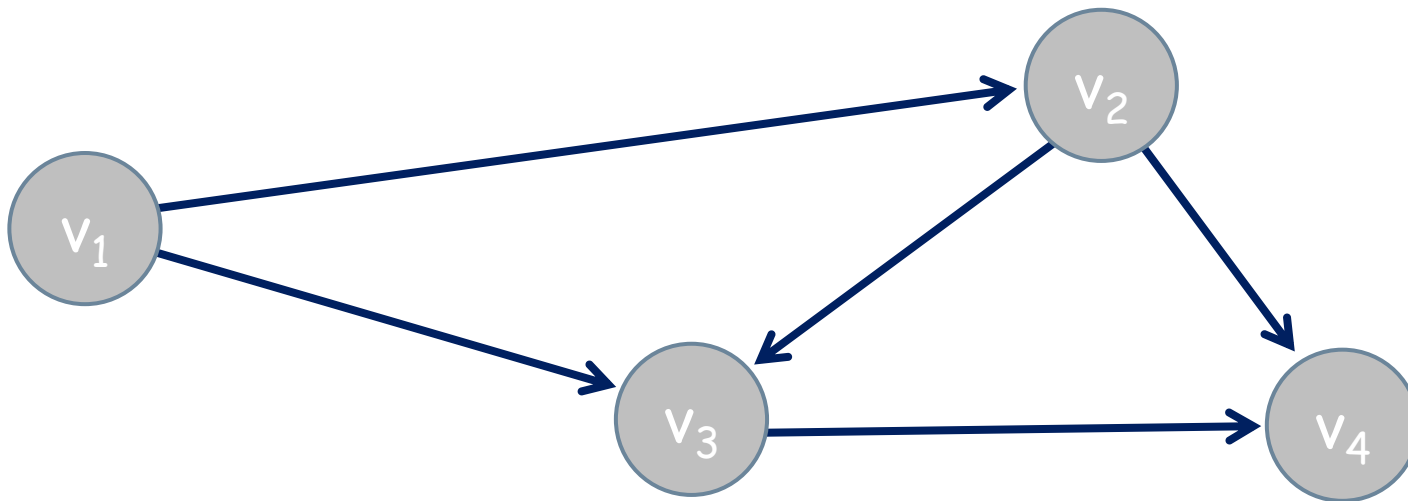
Κατασκευή γραφήματος $H(G)$

Γράφημα Ισχυρών Συνιστωσών $H(G)$



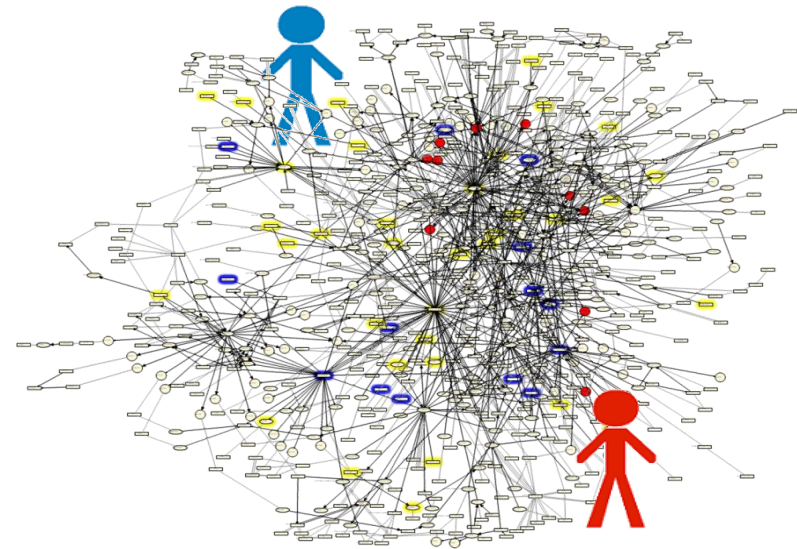
Κατασκευή γραφήματος $H(G)$

Γράφημα Ισχυρών Συνιστωσών $H(G)$



Γραφήματος $H(G)$... Άκυκλο Κατευθυνόμενο Γράφημα

Ε Σύστημα Κοινωνικής Δικτύωσης



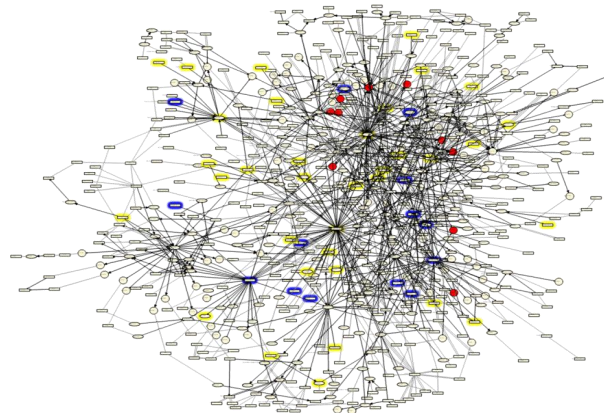
Ε

Σύστημα Κοινωνικής Δικτύωσης

Η εταιρία MS-Social με κύρια δραστηριότητα σε υπηρεσίες «κοινωνικής δικτύωσης», διαθέτει το λογισμικό κοινωνικής δικτύωσης Net-book[©] στο οποίο έχουν εγγραφεί $n \geq 1$ χρήστες (users)

$$X_1, X_2, \dots, X_n$$

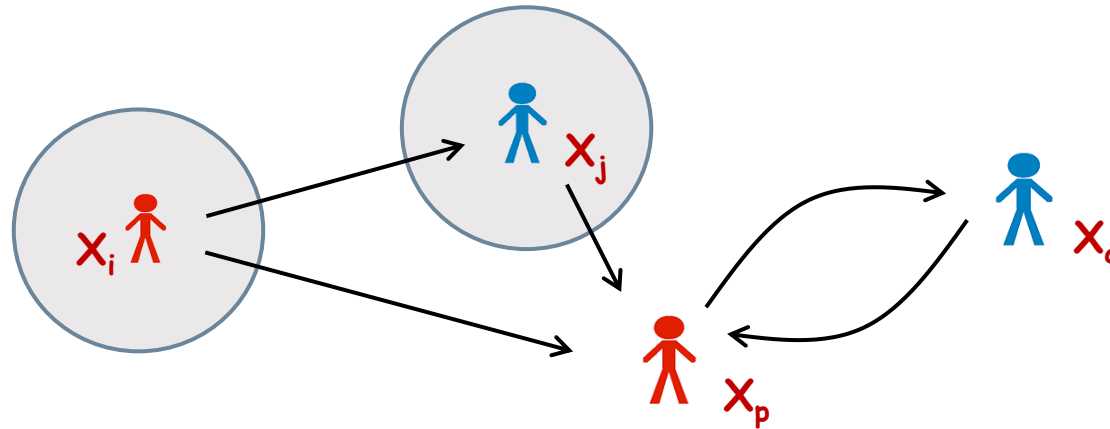
Οι χρήστες συνδέονται μεταξύ του με **άμεση ή έμμεση σχέση γνωριμίας**, την οποία σχέση έχουν γνωστοποιήσει στο Net-book[©].



Ε Σύστημα Κοινωνικής Δικτύωσης

Ορισμοί

- Ο χρήστης X_i συνδέεται με «άμεση σχέση γνωριμίας» με τον χρήστη X_j εάν:
ο χρήστης X_i γνωρίζει άμεσα τον X_j
χωρίς αυτό να σημαίνει ότι κατ' ανάγκη και ο X_j γνωρίζει τον X_i , $1 \leq i, j \leq n$.
- Η άμεση σχέση γνωριμίας συμβολίζεται $\langle X_i, X_j \rangle$.



Ε Σύστημα Κοινωνικής Δικτύωσης

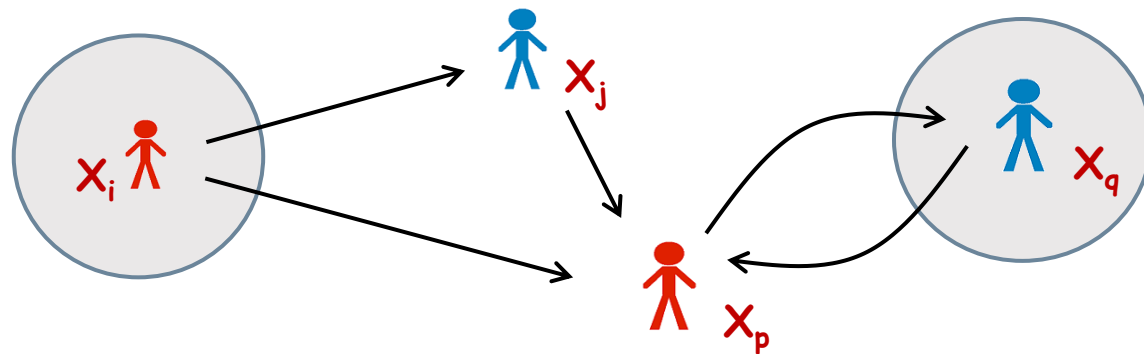
Ορισμοί

- Ο χρήστης X_i συνδέεται με «έμμεση σχέση γνωριμίας» με τον χρήστη X_j εάν:

ο χρήστης X_i γνωρίζει τον X_j μέσω κάποιου άλλου χρήστη X_k

δηλ. εάν υπάρχει μια ακολουθία από χρήστες $(X_i, X_{k1}, X_{k2}, \dots, X_{kp}, X_j)$ τέτοια ώστε:

$$\langle X_i, X_{k1} \rangle, \langle X_{k1}, X_{k2} \rangle, \dots, \langle X_{kp}, X_j \rangle, k_p \geq 1.$$



Ε

Σύστημα Κοινωνικής Δικτύωσης

Ορισμοί

- Μια ομάδα χρηστών S του Net-book[©] ονομάζεται **«ομάδα γνωριμίας»** εάν:

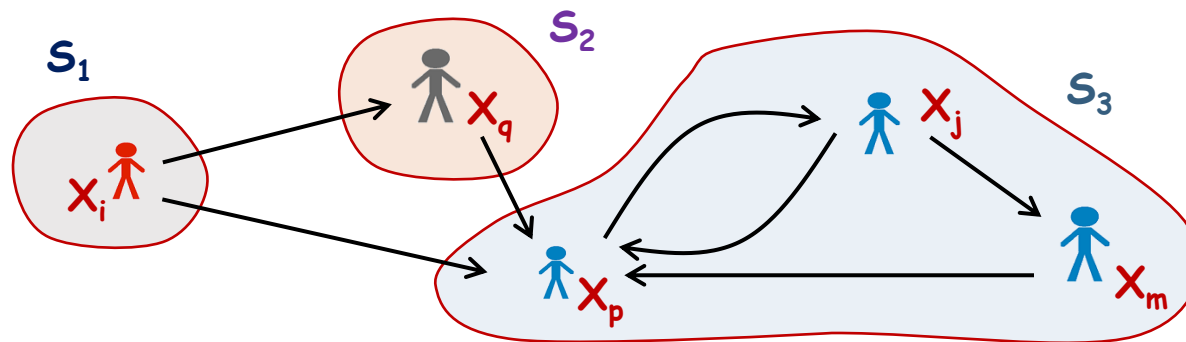
$\forall$ ζεύγος χρηστών X_i και X_j της S

ο χρήστης X_i γνωρίζει άμεσα ή έμμεσα τον χρήστη X_j

και

ο χρήστης X_j γνωρίζει άμεσα ή έμμεσα τον χρήστη X_i ,

$1 \leq i, j \leq |S|$.

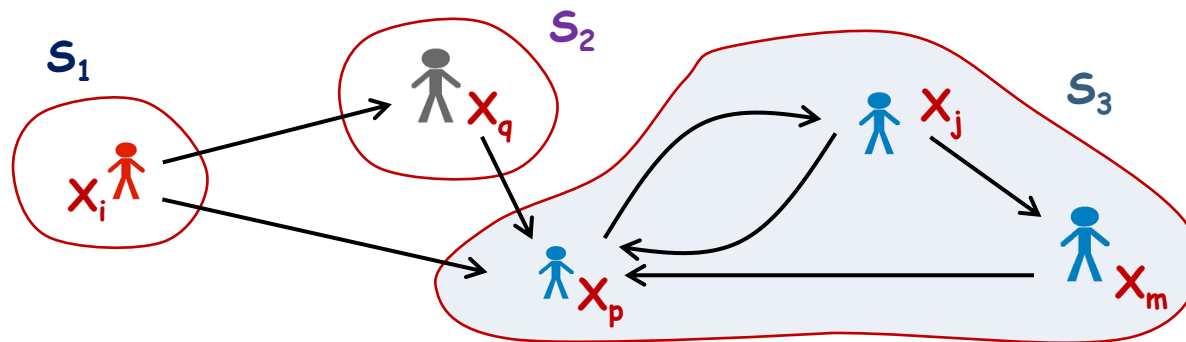


Σύστημα Κοινωνικής Δικτύωσης

- Μία ομάδα γνωριμίας S_i του Net-book[©] ονομάζεται

«Πολύ-δημοφιλής»

εάν όλοι οι χρήστες του δικτύου γνωρίζουν τα μέλη της S_i , δηλ. κάθε χρήστης του Net-book[©] έχει άμεση ή έμμεση σχέση γνωριμίας με οποιοδήποτε μέλος της ομάδας S_i , $1 \leq i \leq k$.

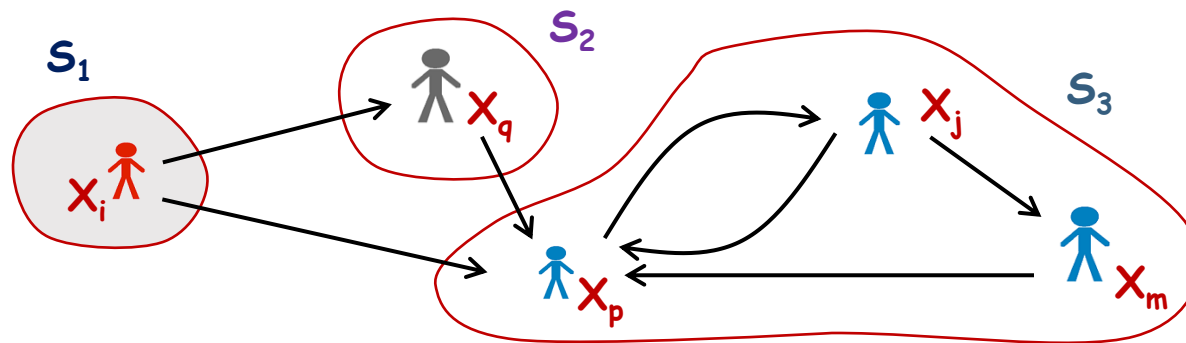


Σύστημα Κοινωνικής Δικτύωσης

- Μία ομάδα γνωριμίας S_i του Net-book[©] ονομάζεται

«Ελάχιστα-δημοφιλής»

εάν όλοι οι χρήστες του δικτύου εκτός ομάδας δεν γνωρίζουν τα μέλη της S_i , δηλ. δεν υπάρχει χρήστης του Net-book[©] εκτός της ομάδας S_i που να έχει άμεση ή έμμεση σχέση γνωριμίας με κάποιο μέλος της S_i , $1 \leq i \leq k$.



Ε

Σύστημα Κοινωνικής Δικτύωσης

Η εταιρία MS-Social θέλει για δική της χρήση, μέσω του συστήματός της, να γνωρίζει όλες της ομάδες γνωριμίας $S_1, S_2, \dots, S_k$ ($1 \leq k \leq n$) του δικτύου της Net-book[©].

Επιπλέον, η MS-Social θέλει να δημιουργήσει και να προσφέρει νέες υπηρεσίες στο σύστημα Net-book[©], όπως:

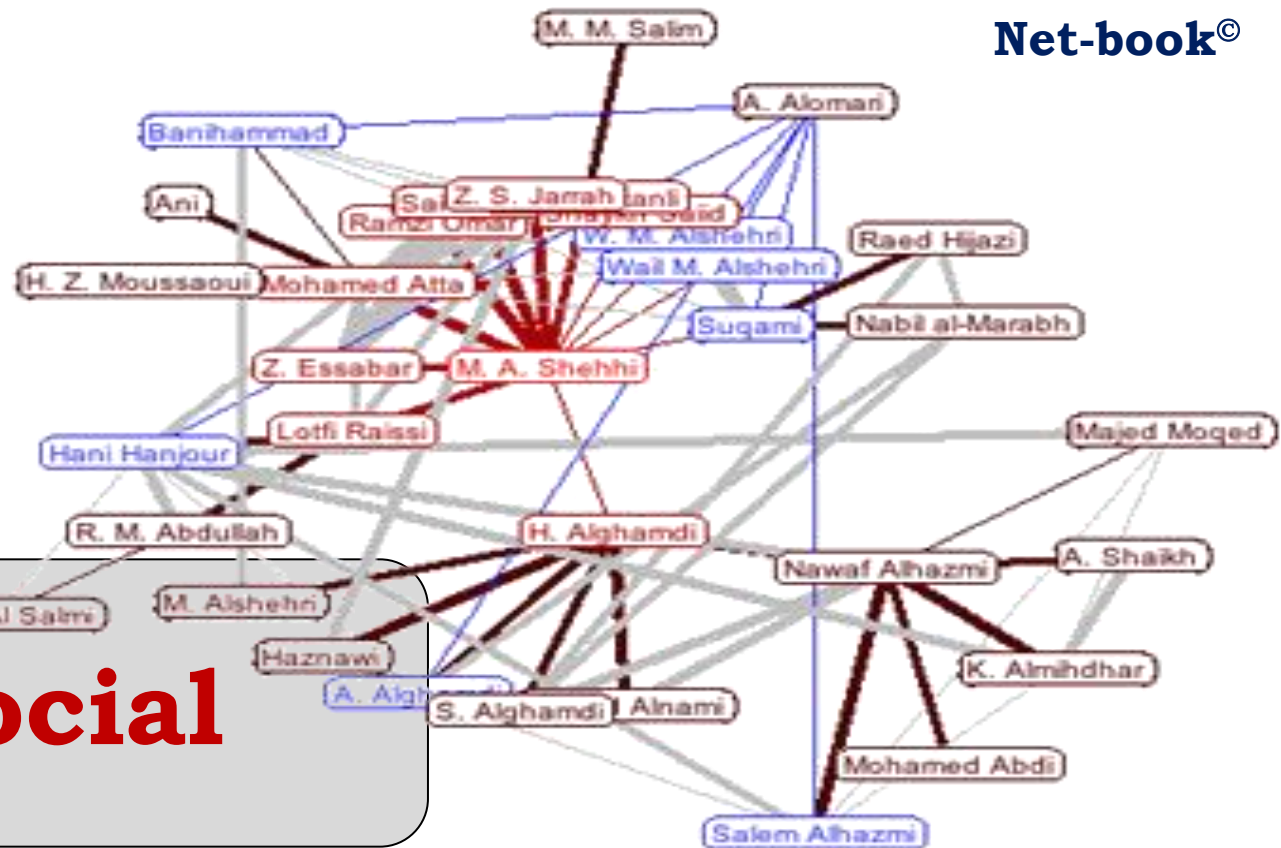
- (1) να απαντάει σε **σταθερό χρόνο** στον χρήστη X_i σε ποια από τις k ομάδες γνωριμίας ανήκει ($1 \leq i \leq n$), εμφανίζοντας id της ομάδας του,
- (2) να εμφανίζει όλα τα μέλη της ομάδα γνωριμίας του χρήστη X_i ($1 \leq i \leq n$), σε **χρόνο γραμμικό** ως προς το πλήθος των μελών της ομάδας, και
- (3) να υπολογίζει όλες τις **πολύ-δημοφιλείς** και **ελάχιστα-δημοφιλείς** ομάδες γνωριμίας του δικτύου Net-book[©] και **για κάθε τέτοια ομάδα να εμφανίζει τα μέλη της**.



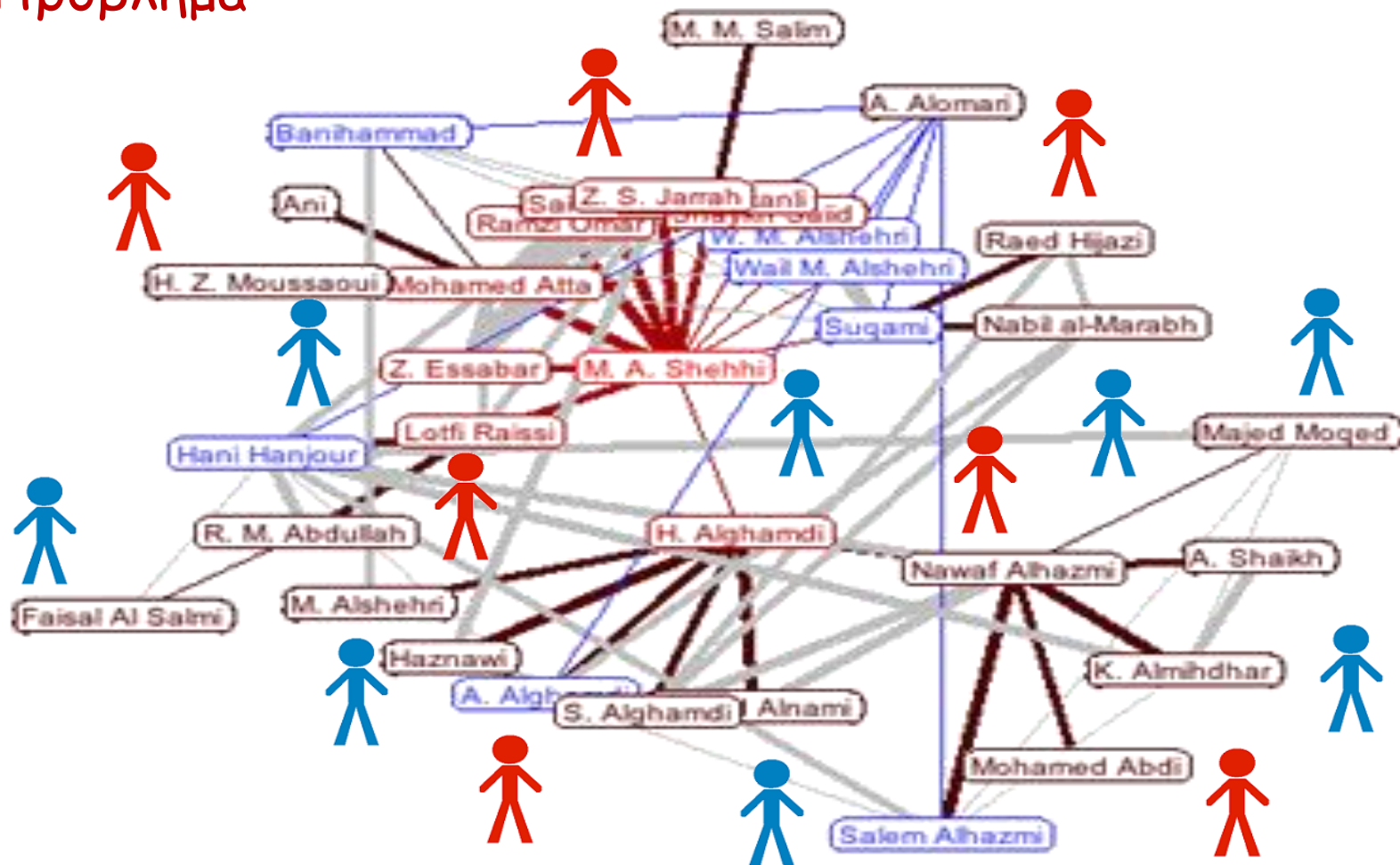
Ε Η Εταιρεία

Net-book©

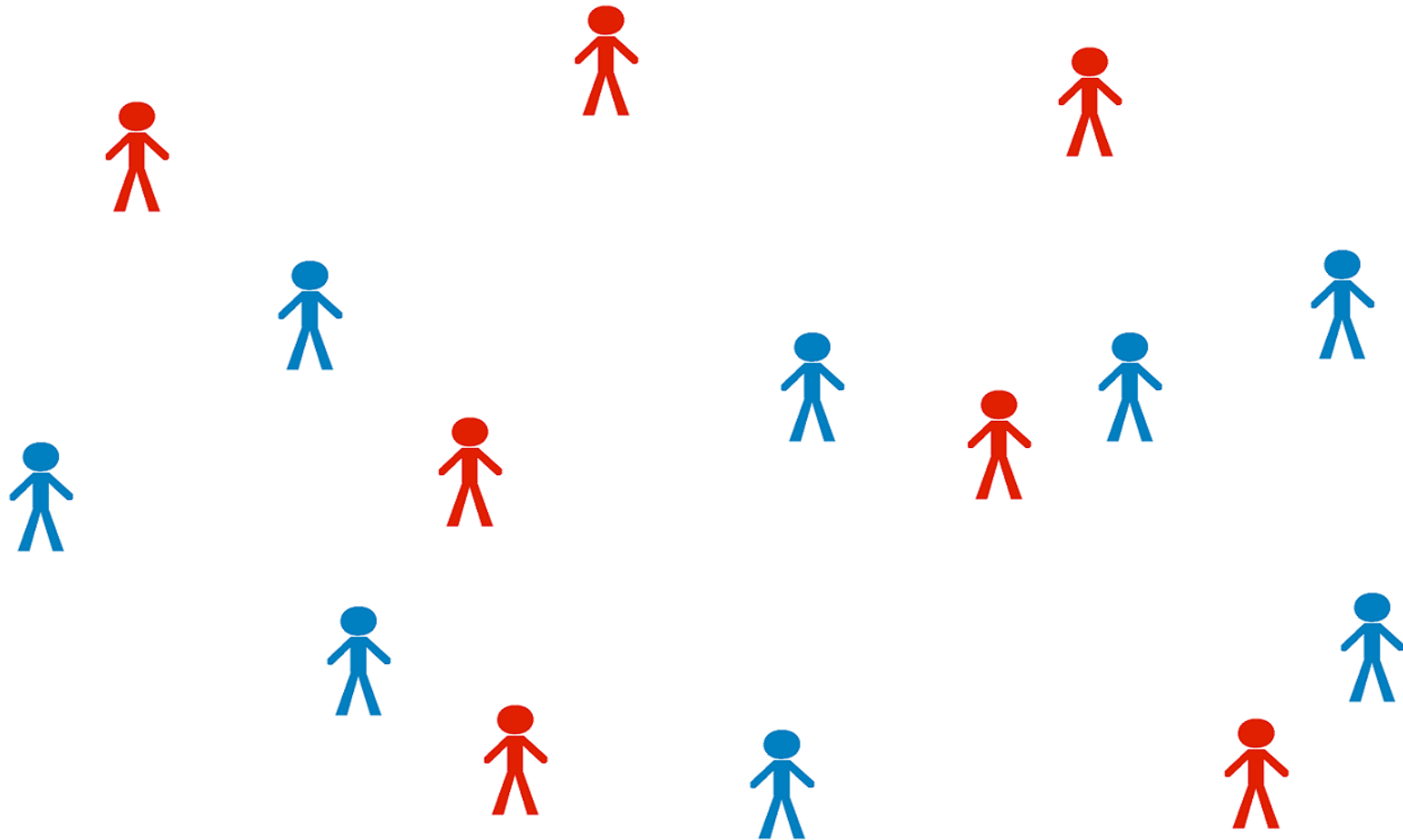
MS-Social



Ε Πρόβλημα

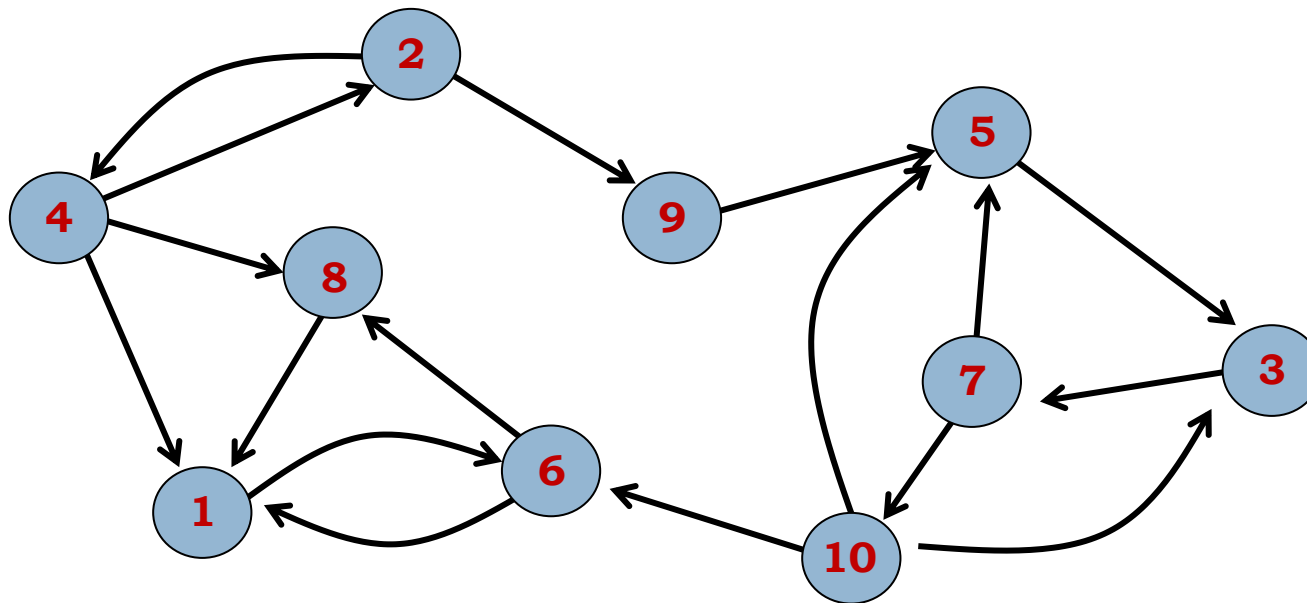


Ε Μοντελοποίηση



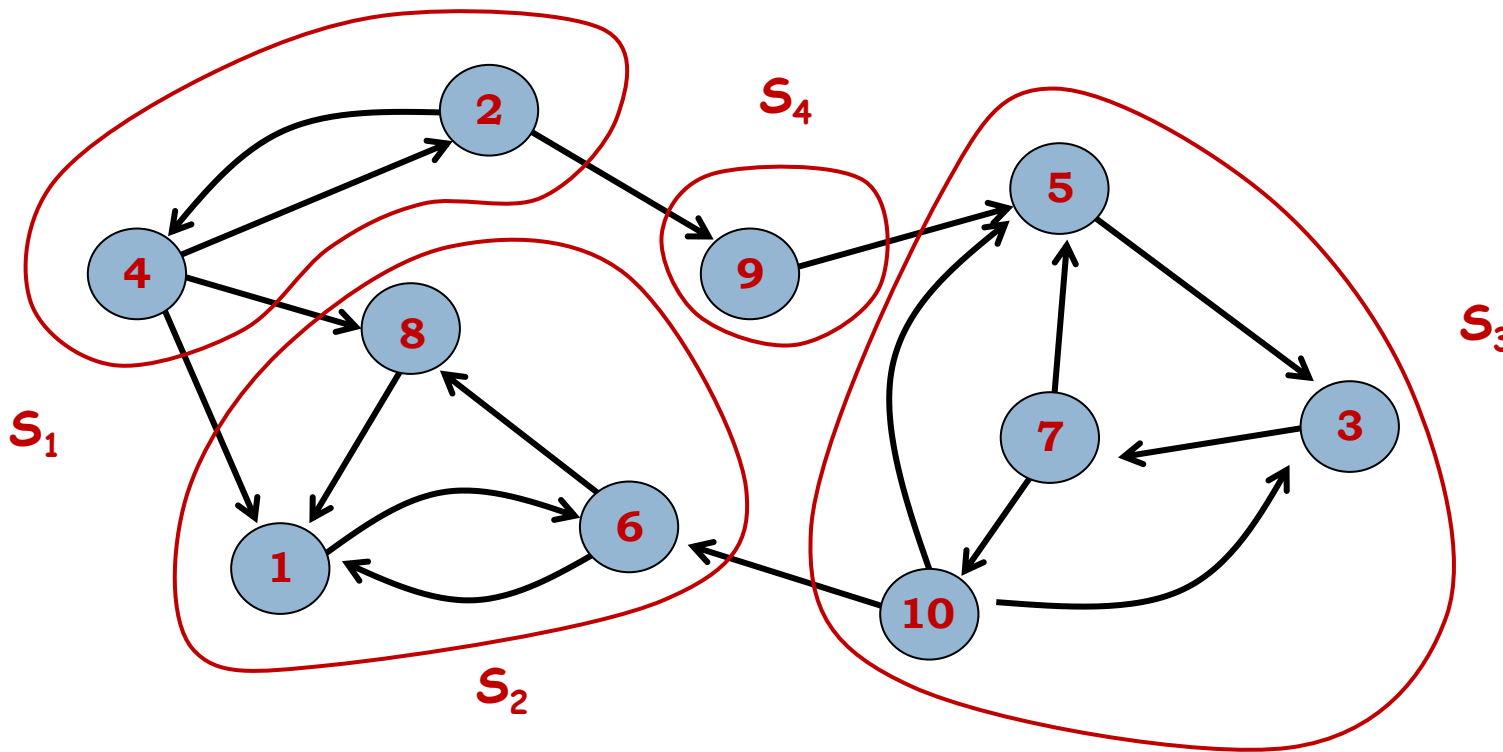
Ε Μοντελοποίηση

**Κατευθυνόμενο
Γράφημα**



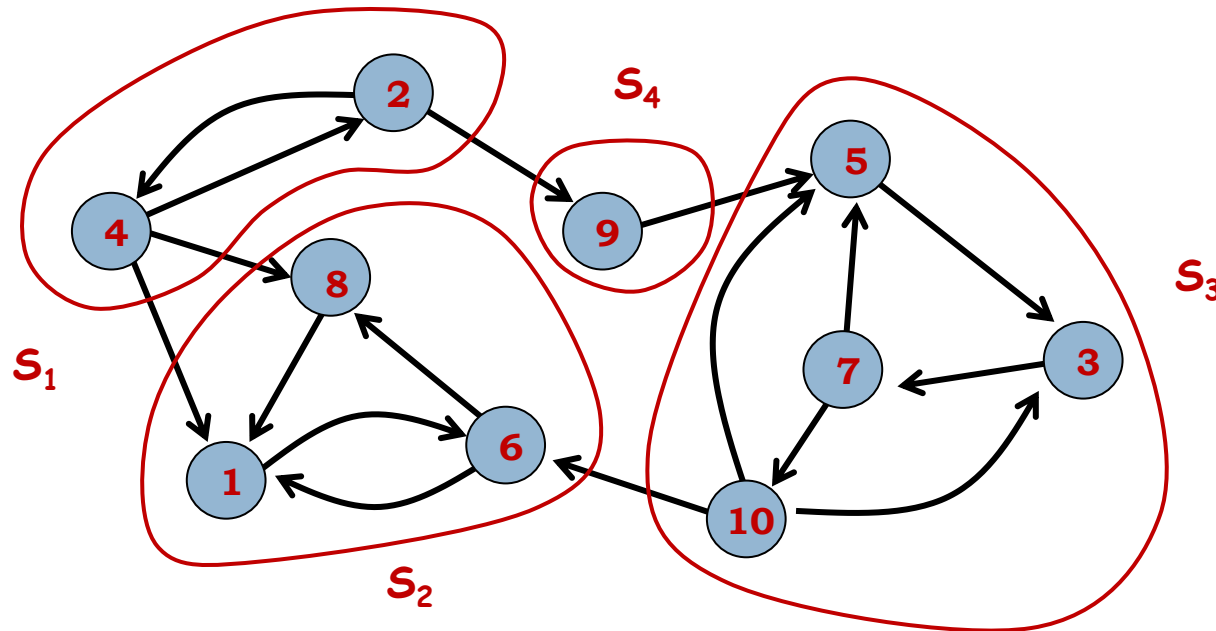
Ε Επίλυση

**Ισχυρά Συνεκτικές
Συνιστώσες**



Ε Επίλυση

**Ισχυρά Συνεκτικές
Συνιστώσες**



A:

1	2	3	4	5	6	7	8	9	10
2	1	3	1	3	2	3	2	4	3

Ε Ερωτήσεις Συστήματος

Επίλυση

Ερώτηση 1: Οι χρήστες X_i και X_j ανήκουν στην ίδια ομάδα γνωριμίας?

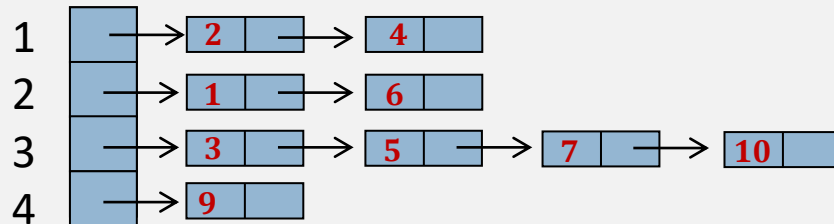
	1	2	3	4	5	6	7	8	9	10
A:	2	1	3	1	3	2	3	2	4	3

If $A[i] = A[j]$ then ΝΑΙ, ανήκουν στην ίδια ομάδα γνωριμίας S_k , όπου $k = A[i] = A[j]$.
else ΟΧΙ, δεν ανήκουν στην ίδια ομάδα γνωριμίας !!!

Ε Ερωτήσεις Συστήματος

Επίλυση

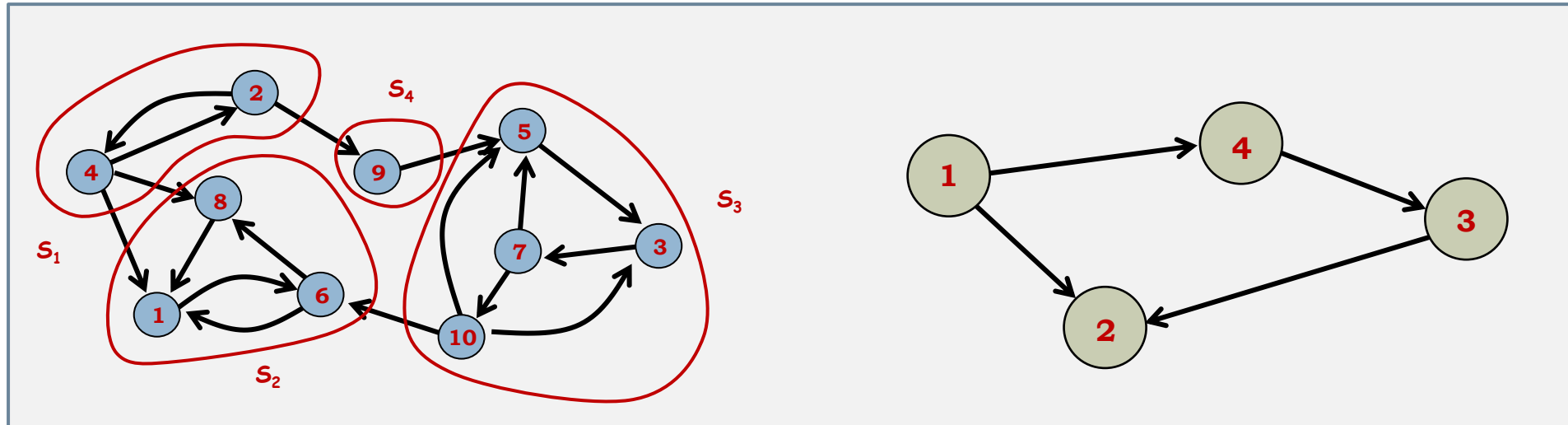
Ερώτηση 2: Ποιοι είναι οι χρήστες X_{i1} , X_{i2} , ..., X_{ip} που ανήκουν στην ίδια ομάδα γνωριμίας S_i ?



Ε Ερωτήσεις Συστήματος

Επίλυση

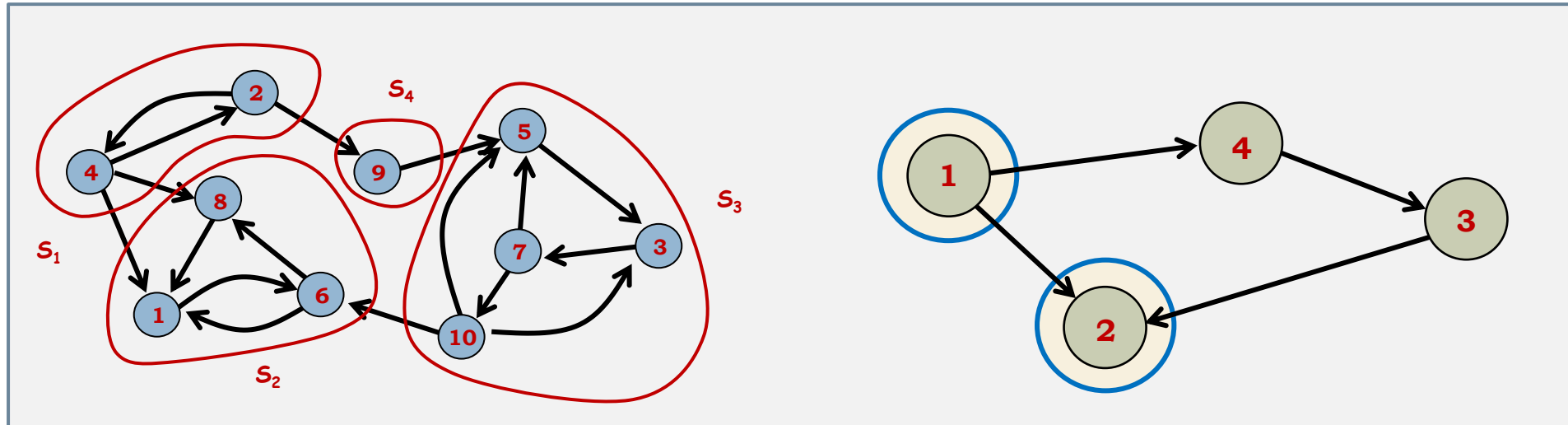
Ερώτηση 3: Ποιες είναι οι **πολύ-δημοφιλείς** και **ελάχιστα-δημοφιλείς** ομάδες γνωριμίας του δικτύου?



Ε Ερωτήσεις Συστήματος

Επίλυση

Ερώτηση 3: Ποιες είναι οι **πολύ-δημοφιλείς** και **ελάχιστα-δημοφιλείς** ομάδες γνωριμίας του δικτύου?



Ε Πολυπλοκότητα Συστήματος

- 1) DFS Διέλευση Γραφήματος
- 2) Αντίστροφο Γράφημα
- 3) Βαθμοί Κόμβων Γραφήματος

**Πολυπλοκότητα
Αλγορίθμων Συστήματος**

$$T(n) = O(n+m)$$

Ευχαριστώ για τη Συμμετοχή σας !!!



Σταύρος Δ. Νικολόπουλος

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Πανεπιστήμιο Ιωαννίνων

Webpage: www.cs.uoi.gr/~stavros
