DFS και Εφαρμογές

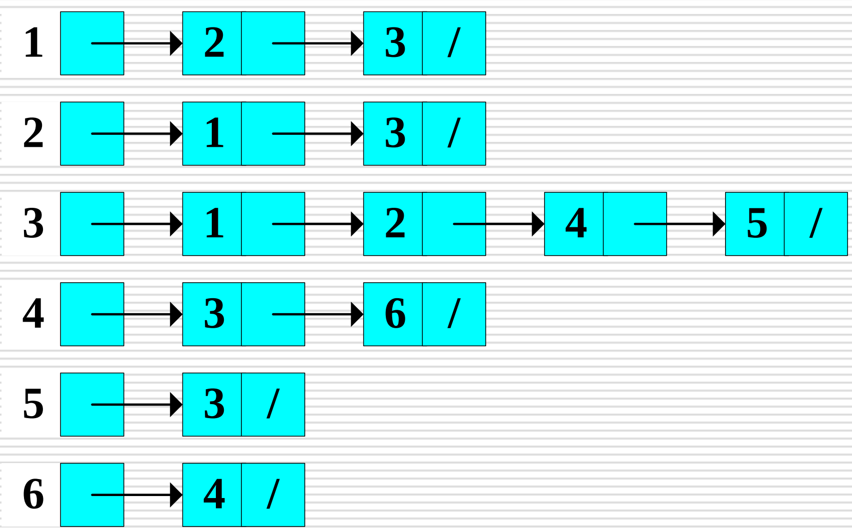
Δημήτρης Φωτάκης, Αριστείδης Παγουρτζής,
Δώρα Σούλιου, Παναγιώτης Γροντάς

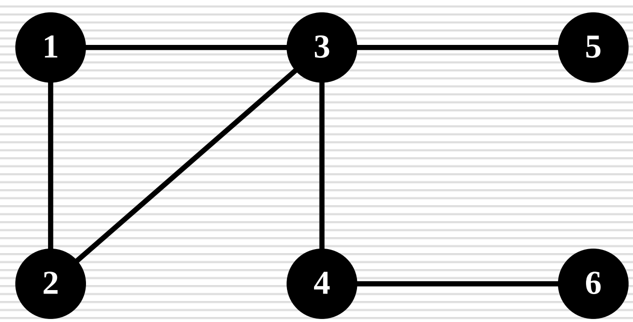
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Εθνικό Μετσόβιο Πολυτεχνείο

Αναζήτηση Κατά Βάθος (DFS)

- Εξερεύνηση νέων κορυφών με απομάκρυνση από αρχική.
- Πρώτη επίσκεψη σε ανεξερεύνητη κορυφή u:
  - Εξερεύνηση (αναδρομικά) όλων των (ανεξερεύνητων) γειτόνων της u, πριν ολοκληρώσουμε με u.





Αναζήτηση Κατά Βάθος (DFS)

- Εξερεύνηση νέων κορυφών με απομάκρυνση από αρχική.
- Πρώτη επίσκεψη σε ανεξερεύνητη κορυφή u:
  - Εξερεύνηση (αναδρομικά) όλων των (ανεξερεύνητων) γειτόνων της u, πριν ολοκληρώσουμε με u.
- Φύσει αναδρομική διαδικασία:

```
DFS(u):
  for v in adj[u]:
        if v==A:
              v=YE
              DFS(v)
  v=E
```

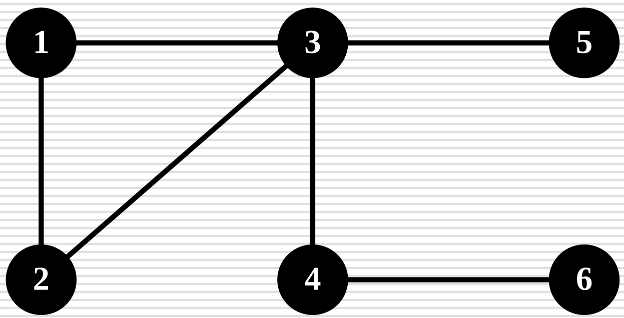
- **Τρία είδη** κορυφών:
  - Ανεξερεύνητη (A): όχι επίσκεψη ακόμη – δεν ξέρω ότι υπάρχει
  - Υπο-εξέταση (YE): επίσκεψη αλλά όχι εξάντληση λίστας γειτνίασης
  - Εξερευνημένη: εξάντληση λίστας γειτνίασης

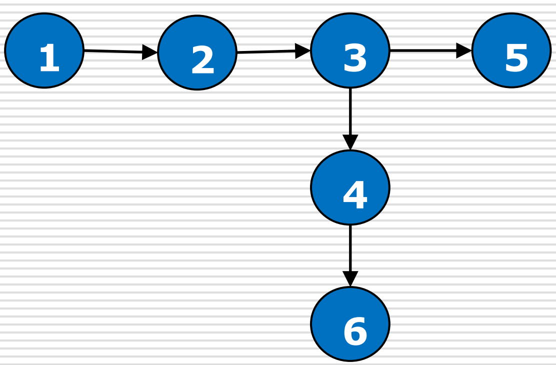
Αναζήτηση Κατά Βάθος (DFS)

- Κορυφές περνούν από παραπάνω στάδια:
  - Αρχικά όλες οι κορυφές ανεξερεύνητες.
  - Πρώτη επίσκεψη ανεξερεύνητης κορ. $\rightarrow$ υπό-εξέταση.
  - Ολοκλήρωση DFS για (ανεξερ.) γείτονες κορ. $\rightarrow$ εξερευνημένη.
- Κορυφή u τίθεται υπό-εξέταση:
  - Όλες οι κορυφές που είναι προσπελάσιμες από u και είναι ανεξερεύνητες θα τεθούν εξερευνημένες πριν u τεθεί εξερευνημένη.
  - Δηλαδή: *Σειρά(Α→ΥΕ) ≠ Σειρά(ΥΕ→Ε) (LIFO)*
  - Ενώ σε BFS: *Σειρά(Α→ΥΕ) = Σειρά(ΥΕ→Ε) (FIFO)*
- Εξέλιξη διαδικασίας αποτυπώνεται σε **DFS-δάσος** και «**χρόνους**» πρώτης επίσκεψης (ΥΕ) και αναχώρησης (Ε).
  - DFS-δάσος: ακμές πρώτης επίσκεψης, ακυκλικό.

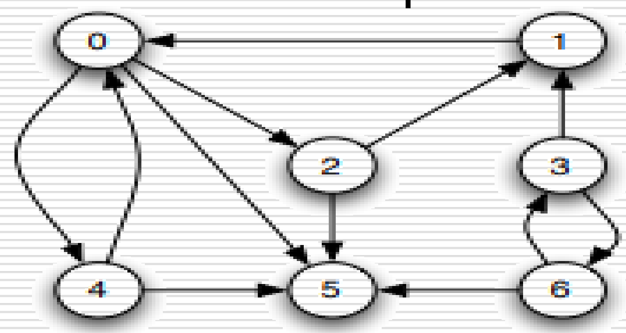
Παραδείγματα DFS δάσους

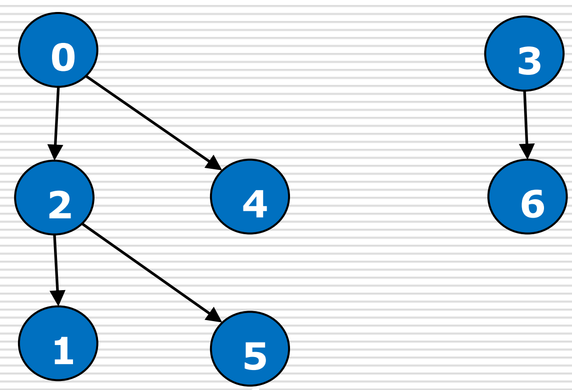
- Μη κατευθυνόμενο





- Κατευθυνόμενο





Υλοποίηση

- Πίνακας κατάστασης: **m[v]** = { A, Y, E }.
- Πίνακας γονέων: **p[v]** = πατέρας v στο DFS-δάσος.
- «Χρόνοι» πρώτης επίσκεψης **d[v]** και αναχώρησης **f[v]**.

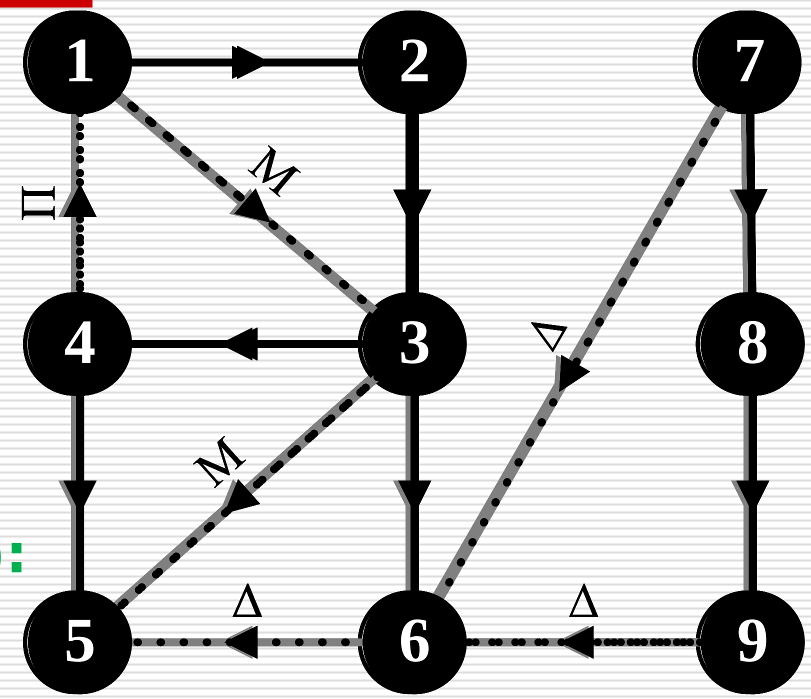
$$
\begin{array}{l}
\mathrm{DFS\_Init}(G(V,E)) \\
\quad t \leftarrow 0; \\
\quad \textbf{for all } v \in V \textbf{ do} \\
\qquad m[v] \leftarrow \mathrm{A};\ p[v] \leftarrow \mathrm{NULL}; \\
\quad \textbf{for all } v \in V \textbf{ do} \\
\qquad \textbf{if } m[v] = \mathrm{A} \textbf{ then } \mathrm{DFS}(v);
\end{array}
\qquad
\begin{array}{l}
\mathrm{DFS}(v) \\
\quad m[v] \leftarrow \mathrm{Y};\ d[v] \leftarrow {+}{+}t; \\
\quad \textbf{for all } u \in L[v] \textbf{ do} \\
\qquad \textbf{if } m[u] = \mathrm{A} \textbf{ then} \\
\qquad\quad p[u] \leftarrow v;\ \mathrm{DFS}(u); \\
\quad m[v] \leftarrow \mathrm{E};\ f[v] \leftarrow {+}{+}t;
\end{array}
$$

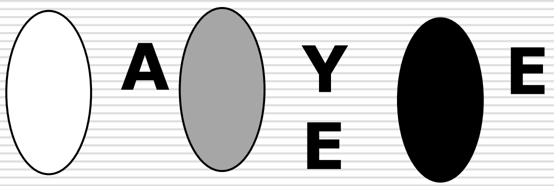
- Χρόνος εκτέλεσης **Θ(n + m)**.
- DFS σε (α) δέντρο, (β) πλήρες γράφημα, (γ) κύκλο.

Παράδειγμα – Κατηγορίες Ακμών

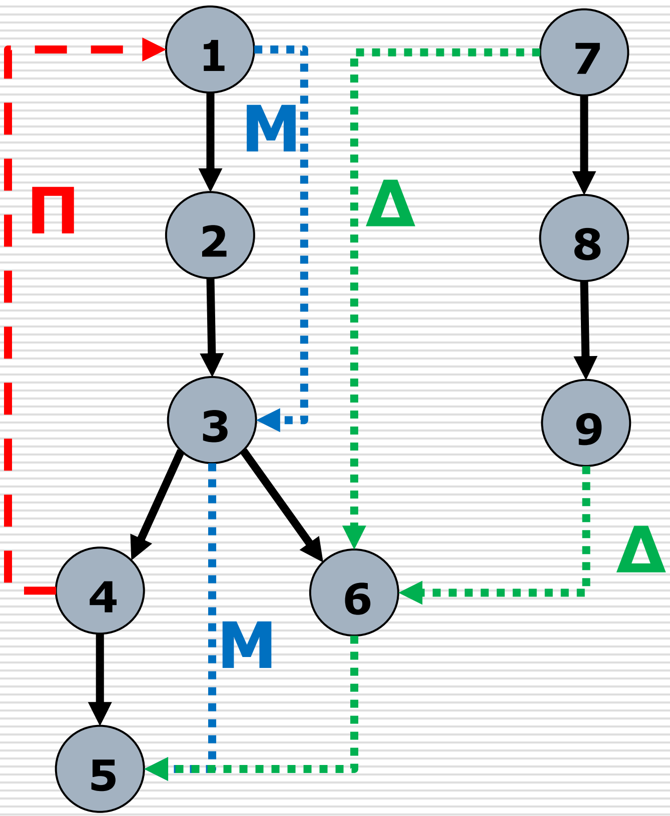
Κατηγορία Ακμής *(u,v)*:

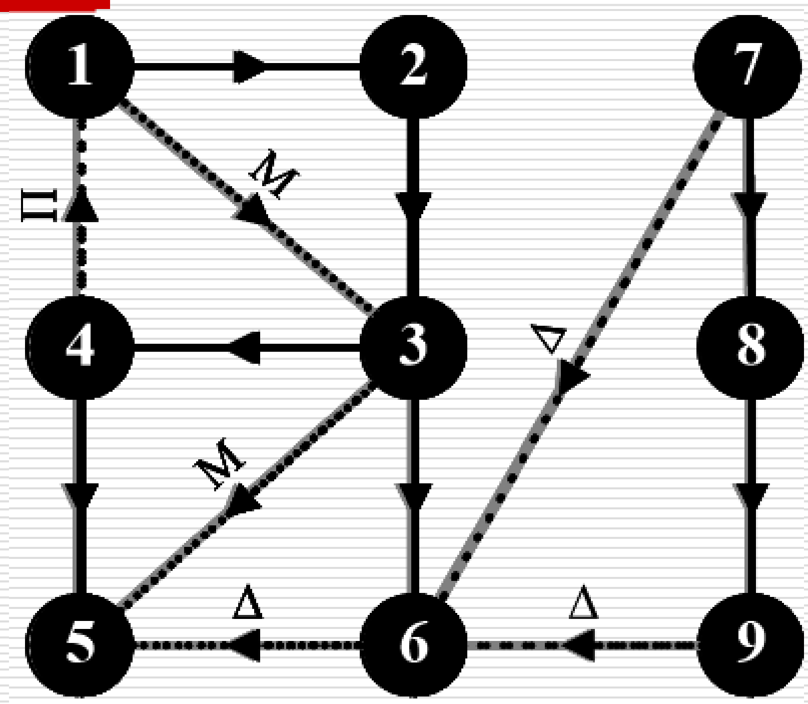
- u υπό-εξέταση
- **Ακμή Δάσους / δέντρου:**
  - Όταν v ανεξερεύνητη.
- **Πίσω ακμή (ανιούσα):**
  - Όταν v υπό-εξέταση
  - Υποδηλώνει κύκλο.
- **Μπροστά ακμή (κατιούσα):**
  - Όταν v εξερευνημένη *και v απόγονος u στο δέντρο*.
- **Ακμή διασταύρωσης (εγκάρσιες):**
  - όταν v εξερευνημένη *και v* ***όχι*** *απόγονος u στο δέντρο*.





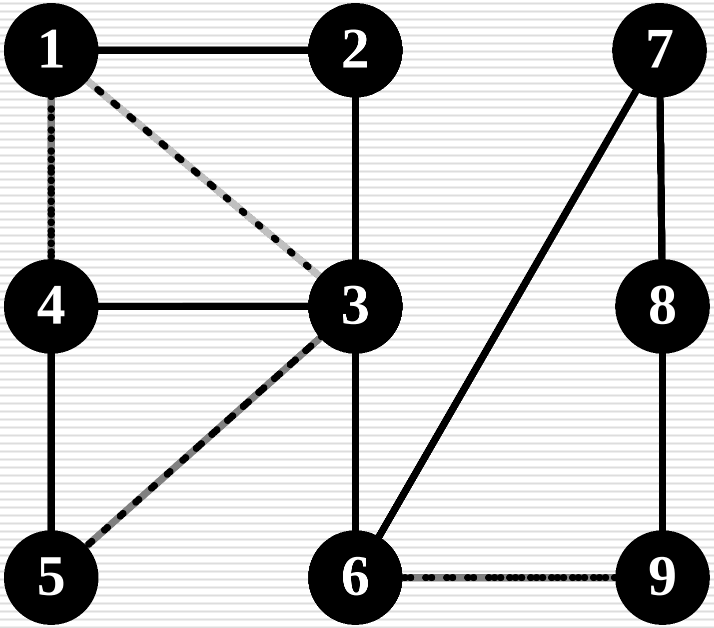
Παράδειγμα – Κατηγορίες Ακμών (2)





Σε μη-κατευθυνόμενο γράφημα

- DFS παράγει μόνο ακμές δέντρου και πίσω ακμές.
  - Ακμή {v, u} με d[v] < d[u] (πρώτα πρώτη επίσκεψη σε v).
  - Πρώτα v ΥΕ, μετά u ΥΕ, μετά u Εξερ, τέλος v Εξερ.
  - Αν κατεύθυνση (v, u) εξερευνήθηκε πρώτη, τότε {v, u} ακμή δέντρου.
  - Αν κατεύθυνση (u, v) εξερευνήθηκε πρώτη, τότε {v, u} πίσω ακμή.



Μερικές Ιδιότητες

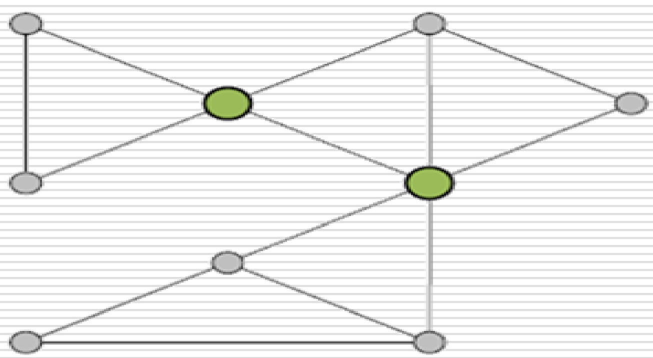
- Για μη-κατευθυνόμενα γραφήματα, DFS υπολογίζει συνεκτικές συνιστώσες (όπως και BFS).
- Αν v απόγονος u στο DFS-δάσος, $[d[v], f[v]] \subset [d[u], f[u]]$
  Αν v όχι απόγονος u στο DFS-δάσος, $[d[v], f[v]] \cap [d[u], f[u]] = \emptyset$
- Γράφημα ακυκλικό ανν DFS δεν παράγει πίσω ακμές.
  - Εξερεύνηση πίσω ακμής (u, v) όταν v ΥΕ $\Rightarrow$
    Μονοπάτι v $\rightarrow$ u και ακμή (u, v) $\Rightarrow$ κύκλος.
  - Έστω κύκλος C, v πρώτη κορυφή C που τίθεται ΥΕ,
    και (u, v) ακμή C που εισέρχεται στην v.
  - u απόγονος της v στο DFS-δάσος γιατί:
    - Υπάρχει v $\rightarrow$ u μονοπάτι.
    - Όλες οι άλλες κορυφές του C είναι Α όταν v γίνεται ΥΕ.
  - Άρα (u, v) πίσω ακμή.

Εφαρμογές

- «Χρόνοι» πρώτης επίσκεψης και αναχώρησης δίνουν πληροφορίες για δομή γραφήματος:
  - Σημεία κοπής και γέφυρες σε μη-κατευθυνόμενα γραφήματα.
  - Τοπολογική διάταξη σε Directed Acyclic Graphs (DAGs).
  - Ισχυρά συνεκτικές συνιστώσες σε κατευθυνόμενα γραφήματα.

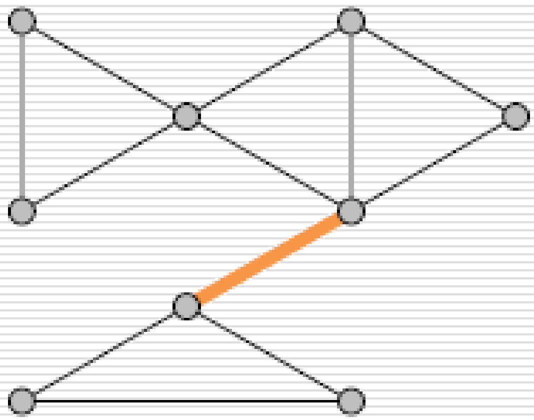
Σημεία Κοπής - Γέφυρες

- **Σημείο κοπής v (cut vertex, articulation point) μη κατευθυνόμενου γραφήματος G(V, E):**
  - αφαίρεση v αυξάνει πλήθος συνεκτικών συνιστωσών του G.



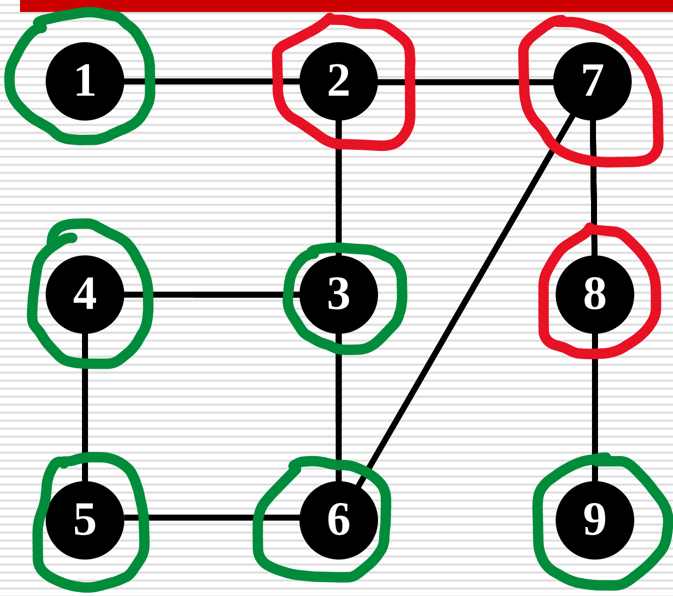
  - Brute Force: O(n(n+m))

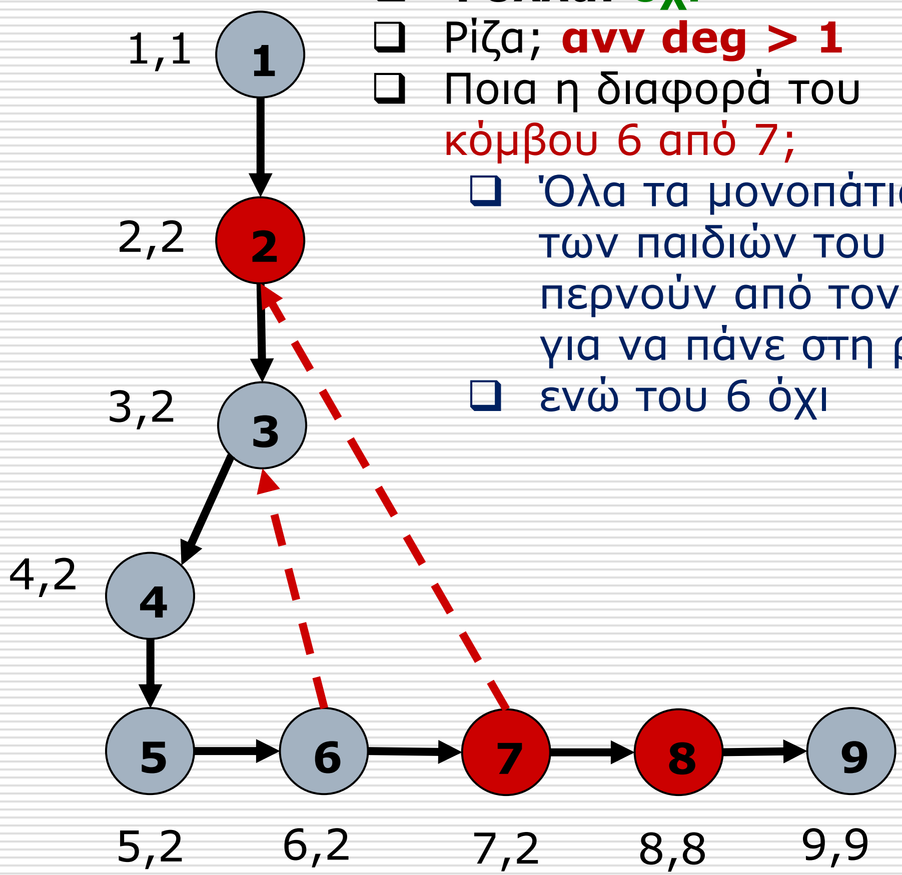
- **Γέφυρα e (bridge) μη κατευθυνόμενου γραφήματος) G(V, E):**
  - αφαίρεση e αυξάνει πλήθος συνεκτικών συνιστωσών



  - Brute Force: O(m(n+m))

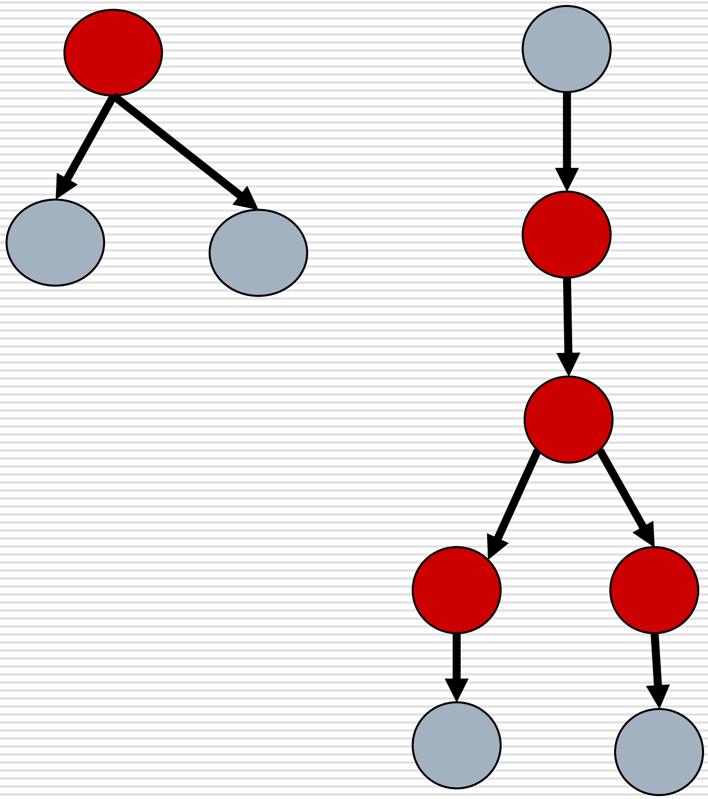
Εύρεση με DFS (γραφικά)





- **Φύλλα: όχι**
- Ρίζα; **ανν deg > 1**
- Ποια η διαφορά του κόμβου 6 από 7;
  - Όλα τα μονοπάτια των παιδιών του 7 περνούν από τον 7 για να πάνε στη ρίζα
  - ενώ του 6 όχι

Εύρεση με DFS (γραφικά)



Είναι σημείο κοπής όταν:

Δεν υπάρχει παράκαμψη του κόμβου από κόμβους των υποδέντρων που ξεκινούν από αυτόν στο μονοπάτι τους προς τη ρίζα

Δεν υπάρχουν πίσω ακμές που να οδηγούν σε ανώτερο επίπεδο

Σημεία Κοπής

- Σημείο κοπής v (cut vertex, articulation point) μη κατευθυνόμενου γραφήματος G(V, E): αφαίρεση v αυξάνει πλήθος συνεκτικών συνιστωσών του G.
  - Ρίζα s DFS-δέντρου T είναι σημείο κοπής ανν s έχει 2 ή περισσότερα παιδιά στο T.
    - Δεν υπάρχουν ακμές διασταύρωσης: όλα τα μονοπάτια από ένα υποδέντρο στο άλλο διέρχονται μέσω s.
  - Κορυφή v ($\neq$ s) είναι σημείο κοπής ανν **υπάρχει** απόγονος u της v στο T τ.ω. όλα u – s μονοπάτια στο G διέρχονται από v.
  - Κορυφή v ($\neq$ s) **δεν** είναι σημείο κοπής ανν **κάθε** απόγονος u της v στο T μπορεί να «παρακάμψει» v (μέσω πίσω ακμής).
    - … ανν από κάθε υποδέντρο με ρίζα παιδί της v έχει πίσω ακμή που καταλήγει σε πρόγονο της v.

Σημεία Κοπής – Τροποποίηση DFS

- low(v): μικρότερο βάθος ενός γείτονα κάποιας κορυφής απόγονου της v στο DFS δέντρο T.
  - πόσο ψηλά μπορείς να φτάσεις με πίσω ακμή του υποδέντρου (αν υπάρχει).
  - Αν v φύλλο στο T:

    low(v) = **min**{***depth(v)***, ***min{ depth(u): (v, u) πίσω ακμή}***}
  - Διαφορετικά:

    low(v) = min{***depth(v)***, ***min{ depth(u): (v, u) πίσω ακμή}***,

    min{ depth(u): w απόγονος v, (w, u) πίσω ακμή }}

(το ελάχιστο low value πίσω ακμών παιδιών της v)

DFS με σημεία κοπής

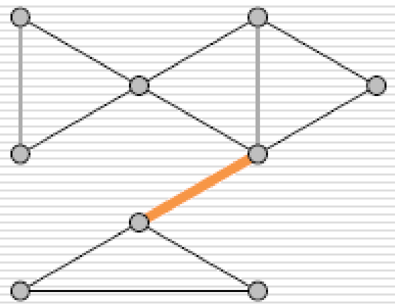
```
DFS_from(graph,v):
        t = t+1
        d[v] = t
        low[v] = t
        isAP[v]  = false
        visited[v] = True
        for u in graph[v]:
                if visited[u] == False:
                        parent[u] = v
                        DFS_from(graph,u)
                        low[v] = min(low[v],low[u])
                elif not u == parent[v]:
                        low[v] = min(low[v],low[u])
                isAP[v] = isAP[v] OR (low[u] >=  d[v])
```

ακμή δέντρου DFS

Πίσω ακμή

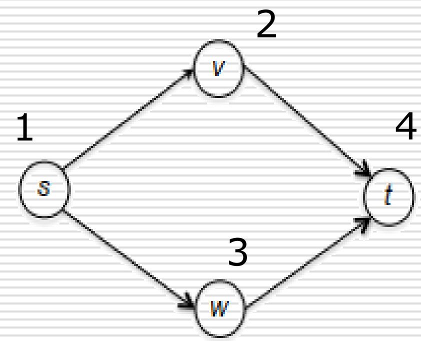
Γέφυρες

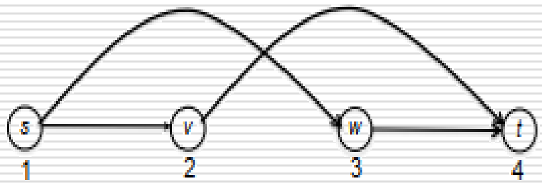
- Ακμή e (μη κατευθυνόμενου γραφήματος) είναι γέφυρα αν αφαίρεση e αυξάνει πλήθος συνεκτικών συνιστωσών
  - Ακμή e είναι γέφυρα ανν e **δεν** ανήκει σε κύκλο.
  - Ακμή e είναι γέφυρα ανν e (και τα άκρα της) αποτελούν δισυνεκτική συνιστώσα.
  - Μόνο ακμές στο DFS δέντρο μπορούν να είναι γέφυρες (απαραίτητες για συνεκτικότητα).
  - Ακμή e = {v, u} (v πατέρας u στο DFS δέντρο) είναι γέφυρα ανν low(u) > d(v).



Τοπολογική Διάταξη

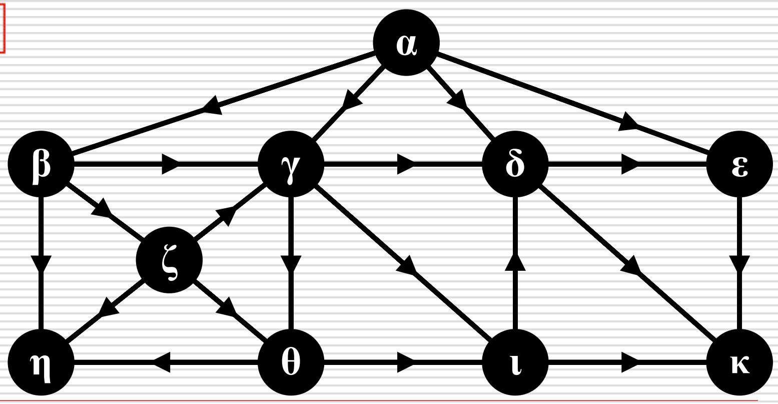
- DAG (Directed Acyclic Graph) αντιστοιχεί σε σχέση μερικής διάταξης:
  - Ακμή (u, v) ⇔ u ≤ v (δηλ. u «προηγείται» v).
- Εφαρμογές DAG
  - Σειρά υπολογισμού αριθμητικών παραστάσεων σε compilers.
  - Προγραμματισμός εργασιών σε σύνθετα έργα.
  - Επίλυση γραμμικών συστημάτων ανισώσεων (περιορισμών)
- Ύπαρξη κύκλου δεν συνάδει με «διάταξη».
- DFS ελέγχει για ύπαρξη κύκλων και υπολογίζει «σειρά» κορυφών **συμβατή** με μερική διάταξη του DAG.
  - Τοπολογική διάταξη.



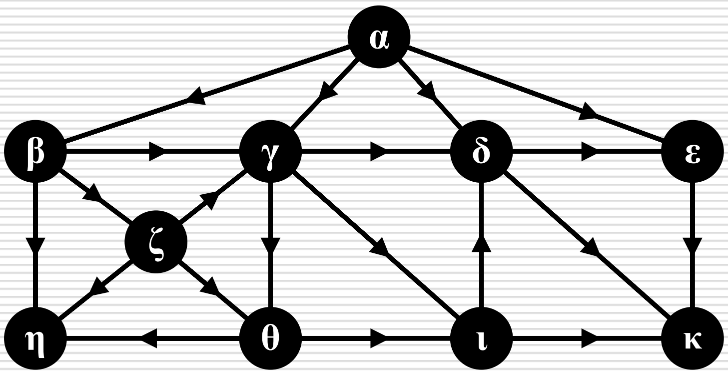


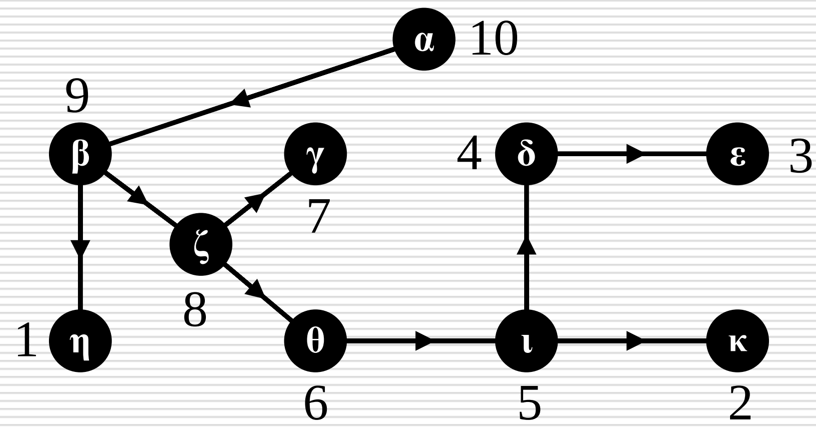
Τοπολογική Ταξινόμηση

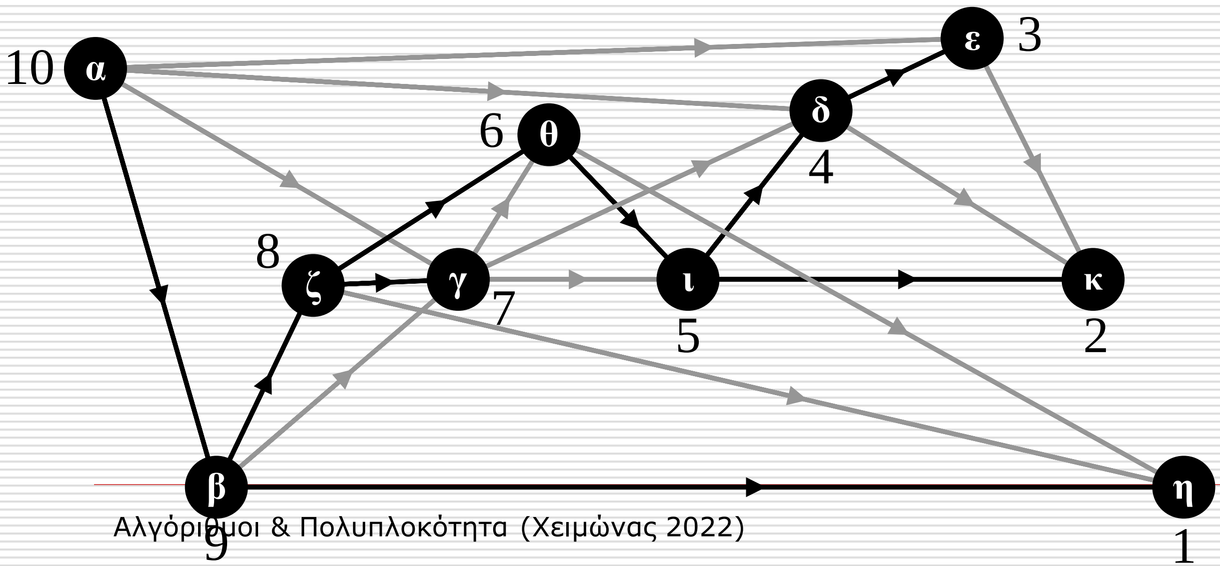
- ... μετάθεση π κορυφών κατευθυνόμενου G(V, E) ώστε
  $$\forall (u,v) \in E\,,\ \pi(u) < \pi(v)$$
  - Δηλ. κορυφές σε ευθεία ώστε όλες οι ακμές να έχουν φορά από αριστερά προς τα δεξιά.
- Βασική ιδέα: Εύρεση και σταδιακή αφαίρεση όλων των κορυφών με out-deg = 0 (ή in-deg = 0)
- Τοπολογική διάταξη ανν γράφημα ακυκλικό (DAG).
  - Κορυφές σε φθίνουσα σειρά χρόνων αναχώρησης του DFS, δηλ. $f[v_1] > f[v_2] > \cdots > f[v_n]$
  - Υλοποίηση με στοίβα: Εξερευνημένη κορυφή μπαίνει.
  - Σειρά αντιστοιχεί σε τοπολογική διάταξη.
  - Χρόνος Θ(n+m).



Παράδειγμα 1

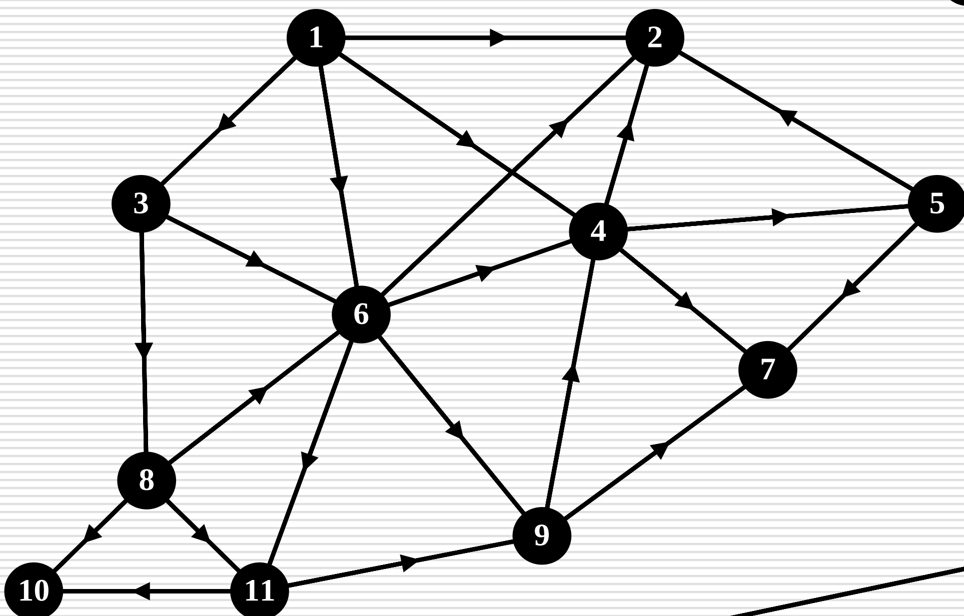


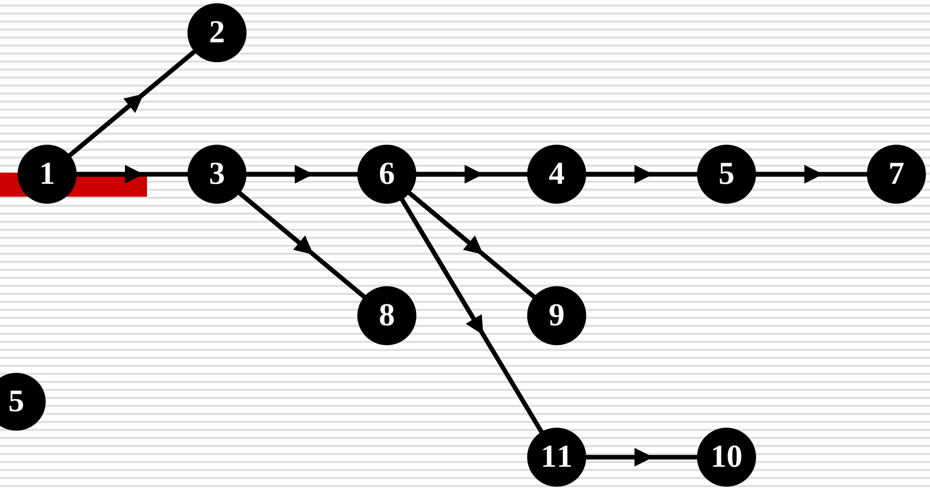


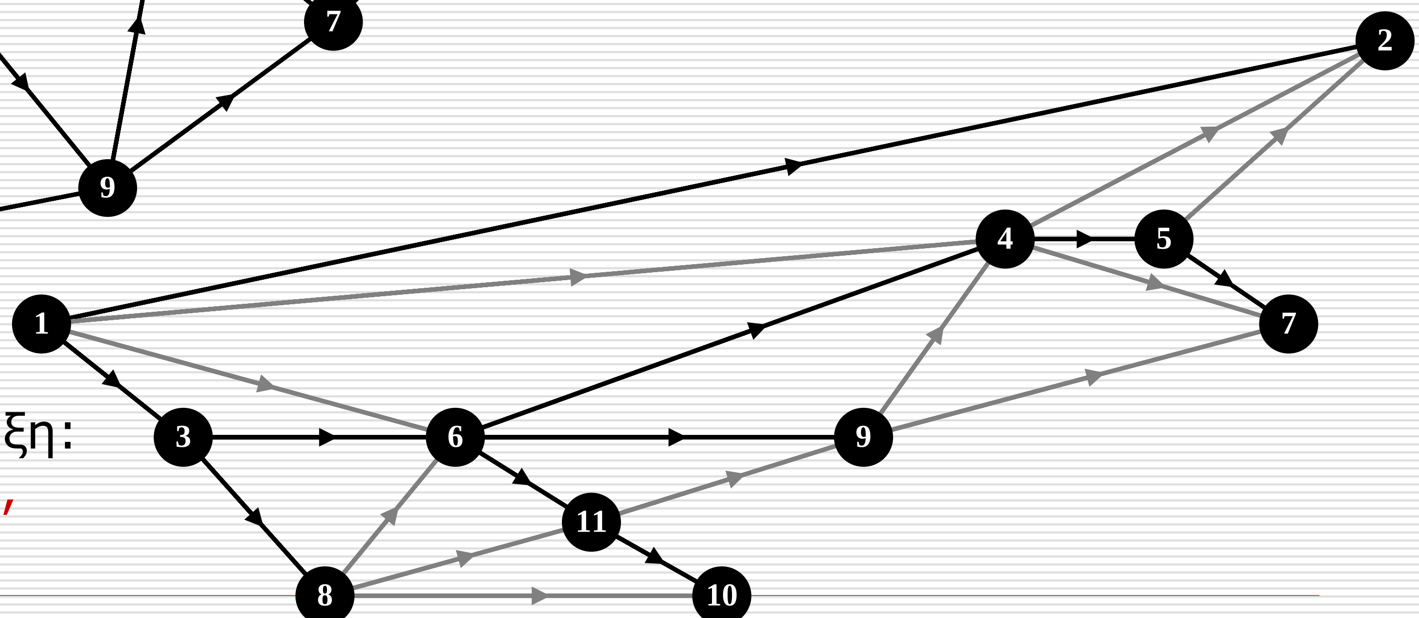


Τοπολογική διάταξη:
α, β, ζ, γ, θ, ι, δ, ε, κ, η

Παράδειγμα 2







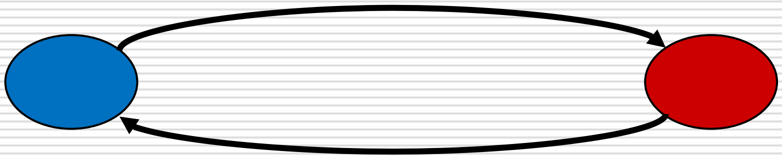
Τοπολογική διάταξη:
1, 3, 8, 6, 11, 10, 9, 4, 5, 7, 2

Τοπολογική Ταξινόμηση: Ορθότητα

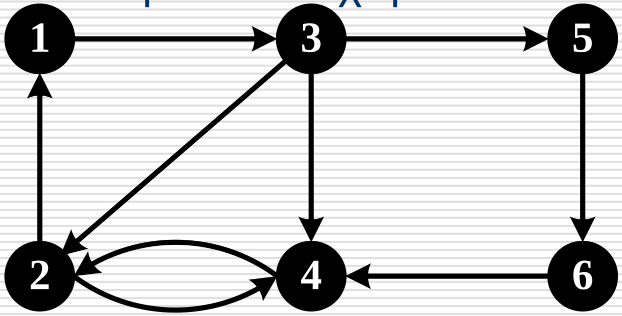
- Έστω DAG $G(V, E)$. Θδο $\forall (u, v) \in E$, $f[u] > f[v]$.
  - Εξερεύνηση (u, v) σημαίνει $u = Y$ και $v \in \{A, E, Y\}$:
  - **v = A**
    - v απόγονος της u στο DFS-δάσος.
    - Άρα $f[u] > f[v]$, γιατί πρώτα τίθεται $f[v]$ και μετά $f[u]$.
  - **v = E**
    - εξερεύνηση της v ολοκληρώθηκε πριν ολοκληρωθεί εξερεύνηση u, άρα $f[u] > f[v]$.
  - **v = Y**
    - αποκλείεται γιατί σημαίνει πίσω ακμή δηλ. κύκλο.

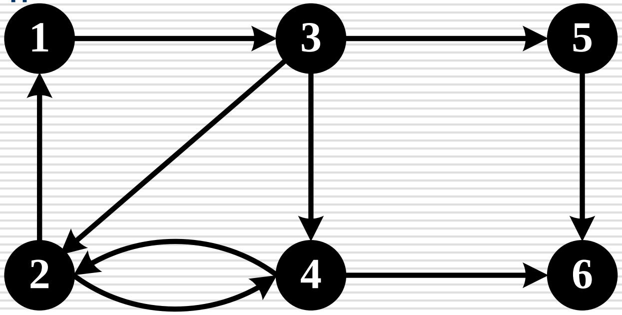
Ισχυρά Συνεκτικές Συνιστώσες

- Κατευθυνόμενο γράφημα G(V, E) **ισχυρά συνεκτικό** αν ∀u, v ∈ V, υπάρχουν u – v και v – u μονοπάτια.
  - Για κάθε ζευγάρι κορυφών ισχυρά συνεκτικού γραφήματος, υπάρχει κυκλική διαδρομή που τις περιλαμβάνει.



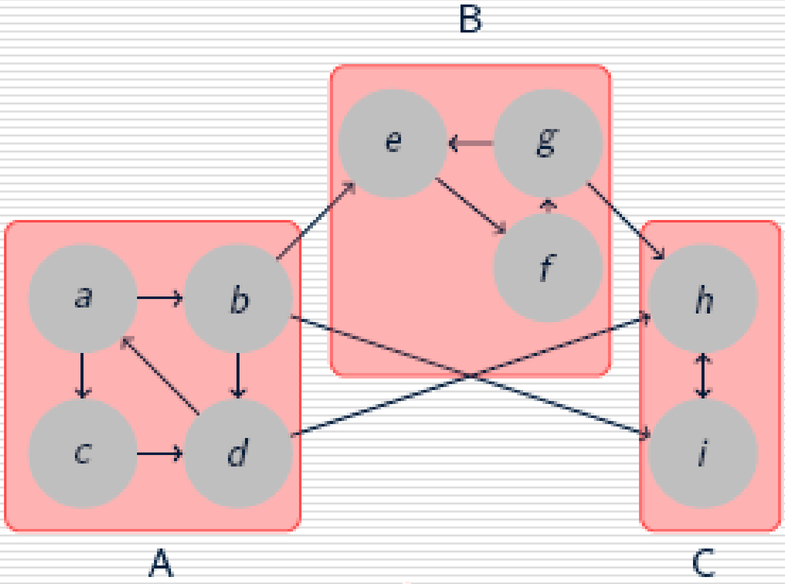
- Αν ένα κατευθυνόμενο γράφημα δεν είναι ισχυρά συνεκτικό, διαμερίζεται σε ισχυρά συνεκτικές συνιστώσες:
  - Μεγιστικά ισχυρά συνεκτικά υπογραφήματα.

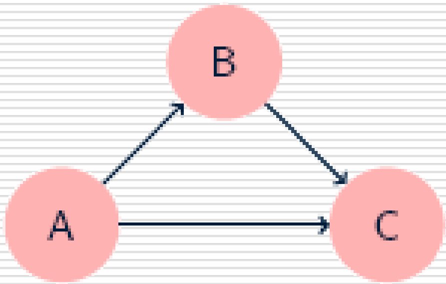




Ισχυρά Συνεκτικές Συνιστώσες (Μεταγράφημα)

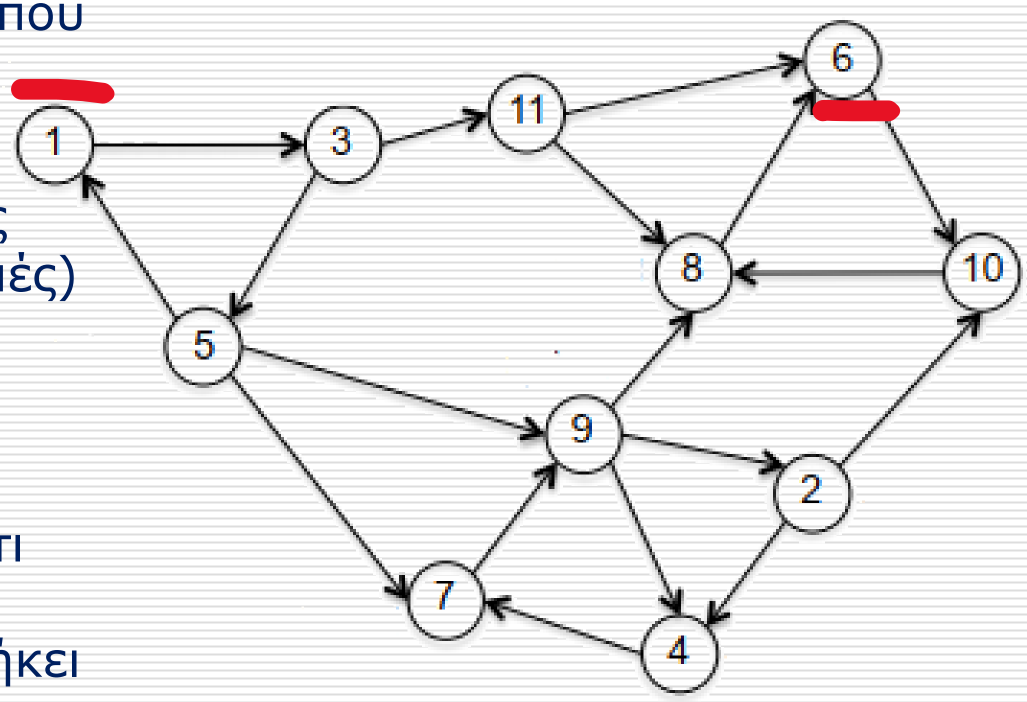
- (Κατευθυνόμενο) γράφημα ισχυρά συνεκτικών συνιστωσών (ΙΣΣ):
  - Κορυφή για κάθε ΙΣΣ
  - Ακμή (X, Y) από ΙΣΣ X προς ΙΣΣ Y ανν (αρχικά) υπάρχει ακμή (v, u) για $v \in X$ και $u \in Y$.
- Γράφημα ΙΣΣ είναι **ακυκλικό** (DAG) και μπορεί να ταξινομηθεί τοπολογικά.
- Μπορούμε να βρούμε ΙΣΣ με DFS?
  - **ΝΑΙ**, αν ξεκινήσουμε από την κατάλληλη κορυφή
  - **Πώς** θα τη βρούμε?



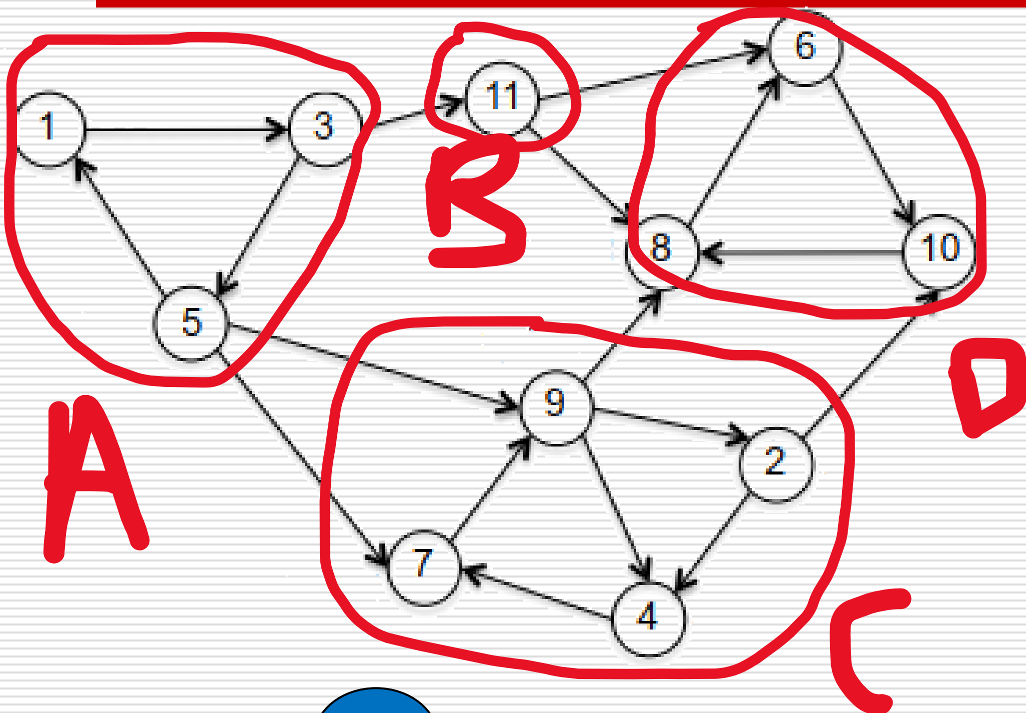


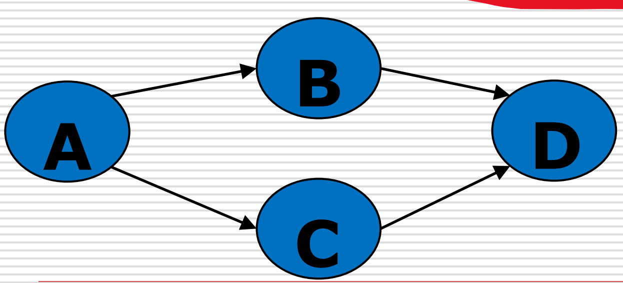
Υπολογισμός Ισχυρά Συνεκτικών Συνιστωσών

- **Διαίσθηση:**
  - Θέλουμε κάποια κορυφή που να ανήκει σε sink ΙΣΣ
- **Ιδέα:**
  - Υπολογισμός τοπολογικής διάταξης (χωρίς πίσω ακμές)
  - Εκκίνηση από τελευταία κορυφή;
- **Πρόβλημα:**
  - **Δεν** είναι υποχρεωτικό ότι τελευταία κορυφή τοπολογικής διάταξης ανήκει σε sink ΙΣΣ



Υπολογισμός Ισχυρά Συνεκτικών Συνιστωσών





(μία) Τοπολογική Ταξινόμηση:

- **1**
- 3
- **11**
- 5
- **7**
- 9
- 2
- 10
- 8
- **6**
- 4 *(ανήκει C και όχι D)*

Υπολογισμός Ισχυρά Συνεκτικών Συνιστωσών

- Όμως ισχύει το *ανάστροφο*:
  - Το πρώτο στοιχείο της τοπολογικής ταξινόμησης ανήκει στο source ΙΣΣ
- Και κάτι ισχυρότερο:
  - Αν ονομάσω κάθε ΙΣΣ με την μικρότερη κορυφή του στην τοπολογικής ταξινόμησης λαμβάνω τοπολογική ταξινόμηση του μεταγραφήματος των ΙΣΣ
  - Απόδειξη
    - Βλ. ορθότητα τοπολογικής ταξινόμησης
- Πώς θα λύσω το πρόβλημα με το sink?
  - Υπολογισμός ανάστροφου γραφήματος ώστε να βρεθεί στο source

Αλγόριθμος Kosaraju (– Sharir)

Εύρεση ΙΣΣ σε σε κατευθυνόμενο G(V, E):

- **Υπολόγισε** τοπολογική διάταξη με DFS (αγνοώντας τις πίσω ακμές).
- **Υπολόγισε** το ανάστροφο γράφημα G^T (αντιστροφή κατεύθυνσης ακμών, G και G^T έχουν ίδιες ΙΣΣ).
- **Εφάρμοσε** DFS σε G^T με σειρά της τοπολογικής διάταξης.
  - Κάθε φορά που φτάνουμε σε αδιέξοδο: νέα ΙΣΣ.

Δύο DFS: Θ(n+m)

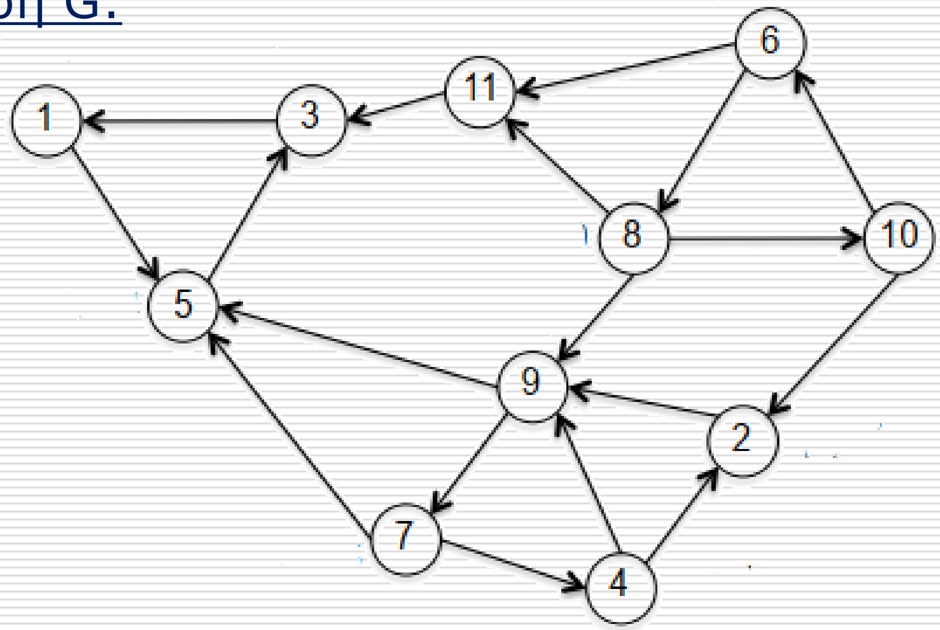
Βελτίωση: Tarjan με μία DFS.

Ολοκληρωμένο παράδειγμα

Τοπολογική Ταξινόμηση G:

- 1
- 3
- 11
- 5
- 7
- 9
- 2
- 10
- 8
- 6
- 4

Υπολογισμός G^T



DFS(G^T) με σειρά τοπολογικής

- 1 → {1,3,5}
- 3 → E
- 11 → {11}
- 5 → E
- 7 → {7,4,2,9}
- 9 → E
- 2 → E
- 10 → {10,8,6}
- 8 → E
- 6 → E
- 4 → E