ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΪΚΑ
ΜΑΘΗΜΑΤΑ

# Θεωρία και Αλγόριθμοι Γράφων

Ενότητα # **8**: Συνδεσμικότητα

Ιωάννης Μανωλόπουλος
Τμήμα Πληροφορικής

# Άδειες Χρήσης

# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

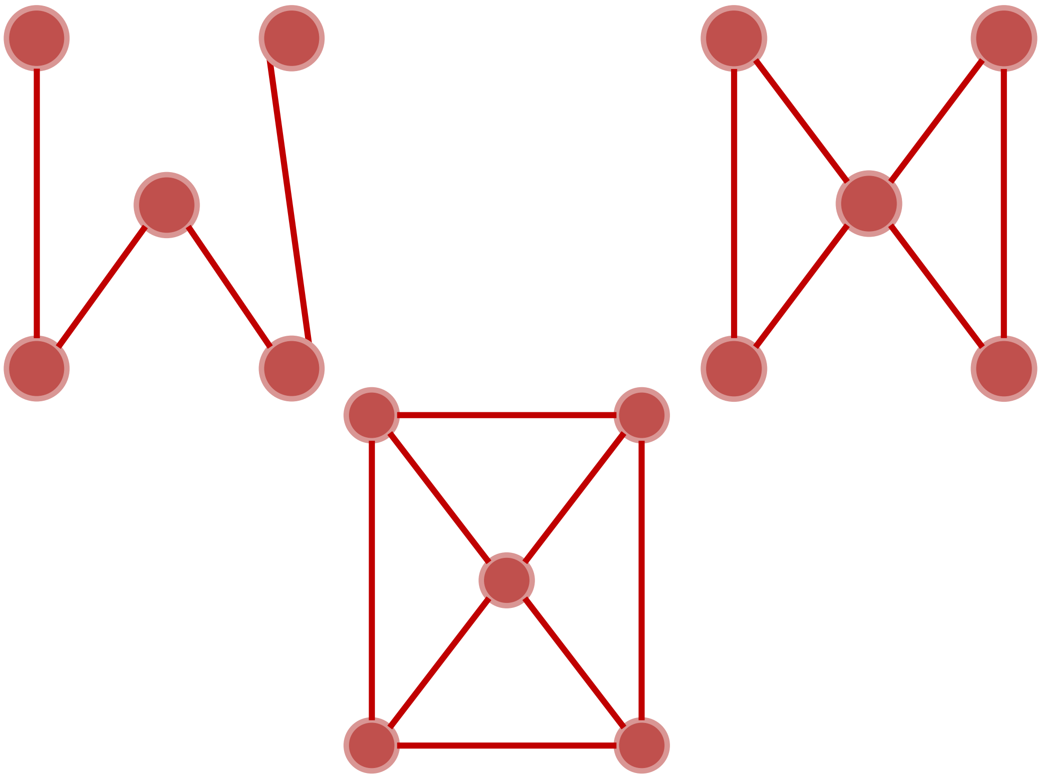
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

# Συνδεσμικότητα

# Εισαγωγή



- Μελέτη βαθμού συνδεσμικότητας ενός γράφου
- Εφαρμογές: σε οποιασδήποτε μορφής δίκτυα (τηλεπικοινωνιακά, συγκοινωνιακά κ.α.)
- Ποιο είναι το καλύτερο δίκτυο από άποψη αντοχής σε λάθη;

# Συνδεσμικότητα Κορυφών

- **Σύνολο αποκοπτουσών κορυφών** *V*΄ (vertex cut set, vertex separating set) ενός συνδεδεμένου γράφου *G* είναι το σύνολο των κορυφών ώστε ο γράφος *G*–*V*΄ να μην είναι συνδεδεμένος και να μην υπάρχει γνήσιο υποσύνολο του *V*΄ με την ίδια ιδιότητα
- **Συνδεσμικότητα κορυφών** (vertex connectivity), VC(*G*), είναι το ελάχιστο $k=|V'|$, ώστε ο γράφος *G* να είναι συνδεδεμένος αν διαγραφούν λιγότερες από *k* κορυφές.
- Ο G λέγεται ***k*-συνδεδεμένος** *k*-connected αν VC(*G*)≥*k*

# Μερικά Θεωρήματα Ι

- **<u>Θεώρημα</u>:** Μια κορυφή *v* ενός δένδρου είναι αποκόπτουσα αν και μόνον αν d(*v*)>1
- **<u>Πόρισμα</u>**: Κάθε μη ασήμαντος απλός συνδεδεμένος γράφος έχει τουλάχιστον 2 κορυφές που δεν είναι αποκόπτουσες
- **<u>Θεώρημα</u>:** Μια κορυφή v είναι αποκόπτουσα αν και μόνον αν υπάρχουν 2 κορυφές u και w (u,w≠v), ώστε η v να βρίσκεται σε κάθε μονοπάτι από την u προς την w

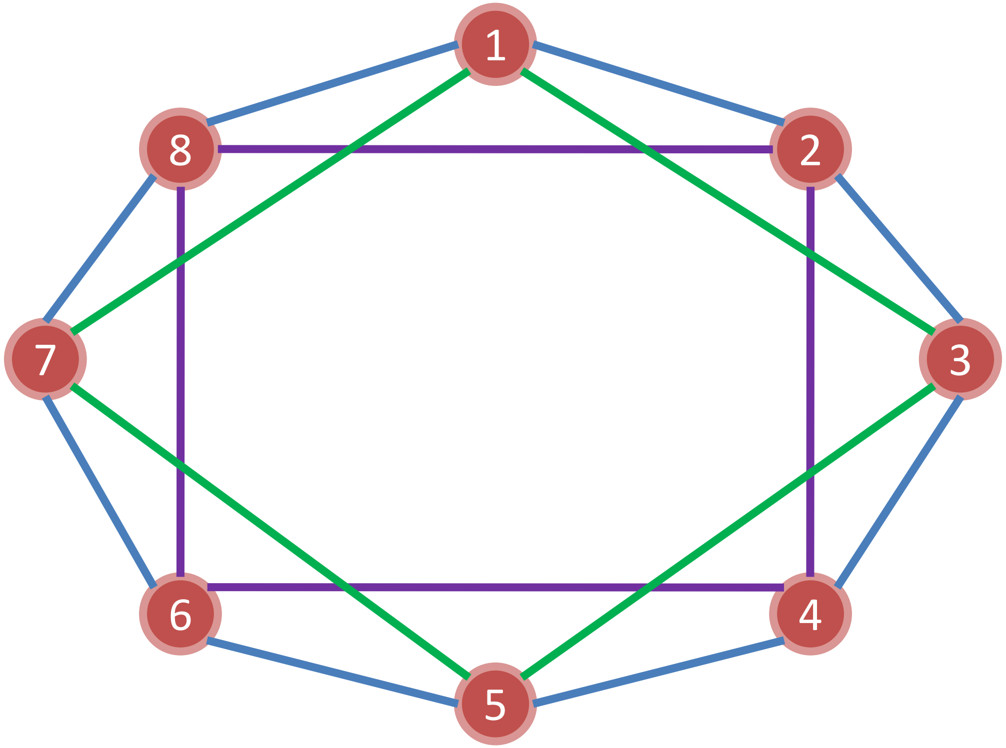
# Συνδεσμικότητα Ακμών

- **Σύνολο αποκοπτουσών ακμών *E΄*** – edge cut set, edge separating set – είναι το σύνολο των ακμών ώστε ο γράφος *G*-*E΄* να μην είναι συνδεδεμένος, χωρίς να υπάρχει γνήσιο υποσύνολο του *E΄* με την ίδια ιδιότητα
- **Συνδεσμικότητα ακμών** EC(*G*) – edge connectivity – ενός γράφου *G* είναι το ελάχιστο *k*=|*E΄*|, ώστε ο *G* να παραμένει συνδεδεμένος έπειτα από διαγραφή *k*-1 ακμών
- Ένας γράφος *G* λέγεται ***k*-συνδεδεμένος ως προς τις ακμές** – edge k-connected – αν EC(*G*)≥*k*

# Μερικά Θεωρήματα ΙΙ

- **<u>Θεώρημα</u>:** Μια ακμή *e* είναι αποκόπτουσα αν και μόνον αν υπάρχουν 2 κορυφές *u* και *w*, τέτοιες ώστε η *e* να βρίσκεται σε κάθε μονοπάτι από την *u* προς την *w*.
- **<u>Θεώρημα</u>**: Μια ακμή είναι αποκόπτουσα αν και μόνον αν δεν περιέχεται σε κύκλο
- **<u>Θεώρημα Whitney</u>**: VC(G) ≤ EC(G) ≤ d(G)
- **<u>Πόρισμα</u>**: EC(G) ≤ floor(2m/n)
- **<u>Θεώρημα</u>**: Έστω 1≤λ≤n-1. Αν d(G)≥(n+λ-2)/2, τότε ο G είναι λ-συνδεδεμένος

# Παράδειγμα




VC(*G*)=?
EC(*G*)=?

# Τεμάχια γράφου Ι

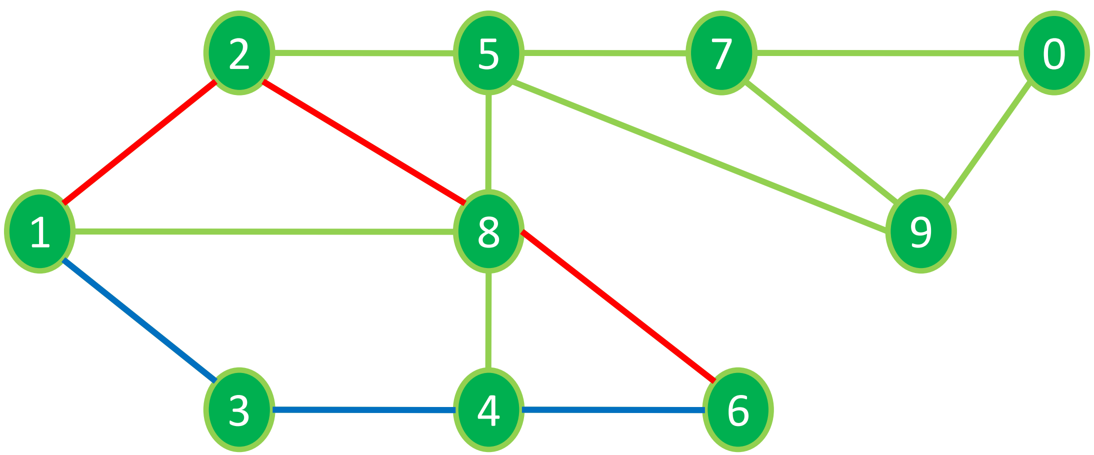
- Ένας δισυνδεδεμένος-biconnected γράφος δεν έχει αποκόπτουσες κορυφές.
- Ένας τέτοιος γράφος αποτελεί ένα τεμάχιο-block ή μια δισυνιστώσα-bicomponent (biconnected component)
- Τεμάχιο ενός γράφου λέγεται ένας υπογράφος που είναι δισυνδεδεμένος και έχει το μέγιστο αριθμό κορυφών.
- Δύο τεμάχια ενός γράφου έχουν το πολύ μία κοινή κορυφή.

# Τεμάχια γράφου ΙΙ

- Κάθε γράφος ταυτίζεται με την ένωση των τεμαχίων του.
- Τα τεμάχια ενός γράφου μπορούν να βρεθούν με DFS
- Δεν μπορεί μία κορυφή να είναι κοινή σε δύο τεμάχια ενός γράφου.
- Τα τεμάχια ενός γράφου χωρίζουν τις ακμές σε ανεξάρτητα σύνολα.

# Εσωτερικά ξένα μονοπάτια

- **Εσωτερικά ξένα μονοπάτια** (internally disjoint paths) είναι δύο μονοπάτια με κοινές τερματικές κορυφές, χωρίς άλλες κοινές κορυφές.



# Θεώρημα Whitney

- **<u>Θεώρημα</u>**: Ένας γράφος *G* με *n*≥3 είναι δι-συνδεδεμένος αν και μόνον αν δύο οποιεσδήποτε κορυφές του είναι συνδεδεμένες με τουλάχιστον δύο εσωτερικά ξένα μονοπάτια.
- **<u>Πόρισμα</u>**: Αν ένας γράφος *G* είναι δισυνδεδεμένος, τότε δύο οποιεσδήποτε κορυφές του ανήκουν σε έναν κύκλο.
- **<u>Πόρισμα</u>**: Αν ένας γράφος αποτελείται από ένα τεμάχιο με n≥3, τότε δύο οποιεσδήποτε ακμές του ανήκουν σε έναν κύκλο.

# Θεώρημα Menger

- **Θεώρημα:** Ο μέγιστος αριθμός εσωτερικά ξένων μονοπατιών από μία κορυφή u σε μια κορυφή v ενός συνδεδεμένου γράφου ισούται με τον ελάχιστο αριθμό κορυφών, που χωρίζουν τις κορυφές u και v.
- **Θεώρημα Whitney:** Ένας γράφος είναι k-συνδεδεμένος αν και μόνον αν όλα τα ζεύγη κορυφών ενώνονται με τουλάχιστον k εσωτερικά ξένα μονοπάτια.

# Πορίσματα (ως προς ακμές)

- **Πόρισμα**: Ο μέγιστος αριθμός εσωτερικά ξένων μονοπατιών από μία κορυφή u σε μια κορυφή v ενός συνδεδεμένου γράφου ισούται με τον ελάχιστο αριθμό ακμών, που χωρίζουν τις κορυφές u και v.
- **Πόρισμα**: Ένας γράφος G είναι k-συνδεδεμένος ως προς τις ακμές, αν και μόνον αν όλα τα ζεύγη κορυφών ενώνονται με τουλάχιστον k εσωτερικά ξένα μονοπάτια.

# Γενικευμένο πρόβλημα συνδέσμου Ι

- Το λεγόμενο πρόβλημα του συνδέσμου αναφέρεται στο πρόβλημα της εύρεσης ελάχιστων ζευγνυόντων δένδρων.
- Ένα ελάχιστο ζευγνύον δένδρο με $n \geq 3$ έχει EC=VC=1.
- Το γενικευμένο πρόβλημα του συνδέσμου είναι “να βρεθεί υπογράφος δοθέντος γράφου με ελάχιστο βάρος, ώστε η συνδεσμικότητα να ισούται με $l$”
- Αν $l$=1, τότε τα προβλήματα ταυτίζονται

# Γενικευμένο πρόβλημα συνδέσμου ΙΙ

- Θα θεωρηθεί η περίπτωση μη ζυγισμένου γράφου.
- Σκοπός είναι η εύρεση ενός γράφου με *n* κορυφές και επιγραφές [0..*n*-1]), *l*-συνεδεδεμένου και με τον ελάχιστο αριθμό ακμών.
- Ο γράφος αυτός συμβολίζεται με $H_{l,n}$
- Διακρίνονται τρεις περιπτώσεις:
  - *l* άρτιο (*l*=2r).
  - *l* περιττό (*l*=2r+1), n άρτιο.
  - *l* περιττό (*l*=2r+1), n περιττό.

# Γενικευμένο πρόβλημα συνδέσμου ΙΙΙ

- *l* άρτιο (*l*=2*r*). Δύο κόμβοι *i* και *j* είναι γειτονικοί, αν $i–r \le j \le i+r$
- *l* περιττό (*l*=2*r*+1), *n* άρτιο. Κατασκευάζεται ο γράφος $H_{2r,n}$ (όπως άνω). Επίσης δύο κόμβοι *i* και $i+n/2$ ενώνονται για $1 \le i \le n/2$.
- *l* περιττό (*l*=2*r*+1), n περιττό. Κατασκευάζεται ο γράφος $H_{2r,n}$. Επίσης ενώνεται ο κόμβος 0 με τους $(n–1)/2$ και $(n+1)/2$ και ο κόμβος *i* με τον κόμβο $i+(n+1)/2$ για $1 \le i \le (n–1)/2$.

# Γενικευμένο πρόβλημα συνδέσμου IV

- <u>**Θεώρημα**</u>: Ο γράφος $H_{l,n}$ είναι $l$-συνδεδεμένος.
- <u>**Θεώρημα**</u>: Ο ελάχιστος αριθμός ακμών του γράφου $H_{l,n}$ είναι ceil($ln$/2).

# Ισομορφισμός

- <u>Ορισμός</u>**:** Δύο γράφοι $G_1=(V_1,E_1)$ και $G_2=(V_2,E_2)$ λέγονται **ισομορφικοί-**isomorphic αν υπάρχει μία αμφιμονοσήμαντη αντιστοιχία f από το το σύνολο $V_1$ στο σύνολο $V_2$ με την ιδιότητα ότι οι κορυφές a, b είναι γειτονικές στο $G_1$ αν και μόνο αν οι κορυφές f(a), f(b) είναι γειτονικές στο $G_2$, για κάθε ζεύγος a,b του $V_1$.
- Η συνάρτηση f ονομάζεται **ισομορφική-** isomorphism.

# Διαπίστωση ισομορφισμού – 1η λύση

Αλγόριθμος ελέγχου ισομορφισμού:

**Είσοδος:** γράφος G και γράφος H

**Έξοδος:** ΝΑΙ ή ΟΧΙ

1. Αν $|V(G)| \neq |V(H)|$, τότε return ΟΧΙ
2. Θεωρούμε μία διάταξη των κορυφών του G
3. Καταγράφουμε τον πίνακα γειτονίας $A_G$ του G
4. Για κάθε διάταξη των κορυφών του H

   Καταγράφουμε τον πίνακα γειτονίας $A_H$

   Αν $A_G$=$A_H$ τότε return ΝΑΙ
5. return ΟΧΙ

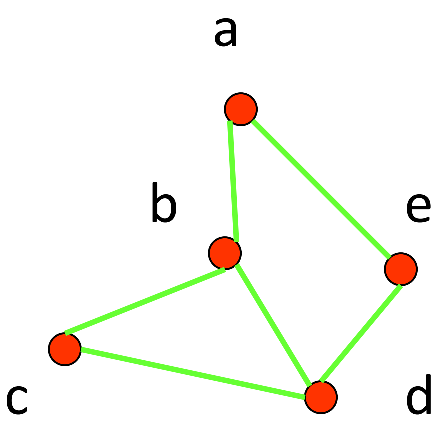
# Διαπίστωση ισομορφισμού – 2η λύση

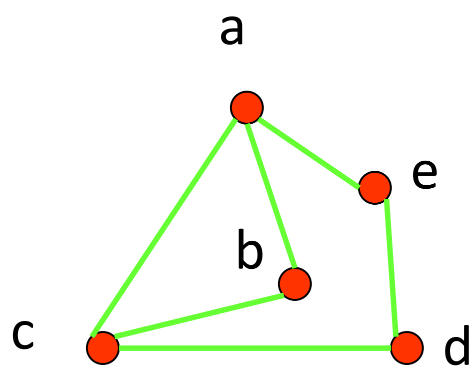
- Έστω ότι οι γράφοι αναπαριστώνται με τη μέθοδο των πινάκων πρόσπτωσης (incidence matrix).
- Μπορεί ο ένας πίνακας πρόσπτωσης να μετασχηματισθεί στον άλλο μέσω αντιμεταθέσεων γραμμών ή/και στηλών. Επίσης μη αποτελεσματική λύση.
- Υπάρχουν αποτελεσματικοί αλγόριθμοι μόνο για ειδικές περιπτώσεις γράφων.
- Επίσης, μπορεί να είναι εύκολη η απόδειξη ότι δύο γράφοι δεν είναι ισομορφικοί

# Αμετάβλητες – invariants

- Συνθήκες για την εύκολη διαπίστωση αν δύο γράφοι δεν είναι ισομορφικοί:
  - Ίδια τάξη;
  - Ίδιο μέγεθος;
  - Ίδια ακολουθία βαθμών;
  - Ίδιος αριθμός συνιστωσών;
  - Για κάθε συνιστώσα του (4) απαντώνται θετικά οι πρώτες τρεις ερωτήσεις;
  - Έχουν οι δύο γράφοι το ίδιο χρωματικό πολυώνυμο;
- Για n<8, αν όλες οι ερωτήσεις απαντηθούν θετικά, τότε οι γράφοι είναι ισομορφικοί (έχει αποδειχθεί).

# Παραδείγματα – Άσκηση Ι



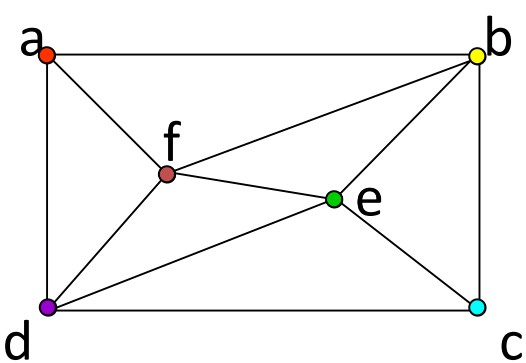

f(a) = e

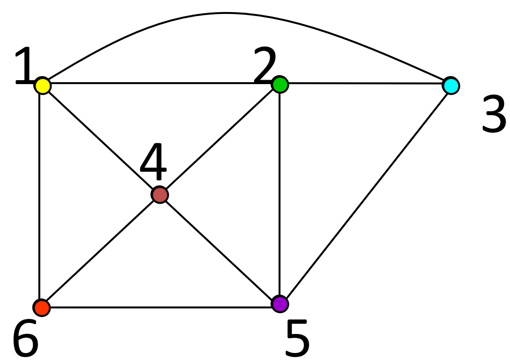
f(b) = a

f(c) = b

f(d) = c

f(e) = d




f(a) = 6

f(b) = 1

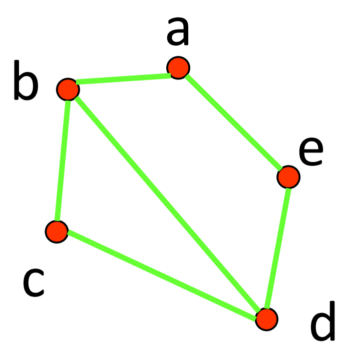
f(c) = 3

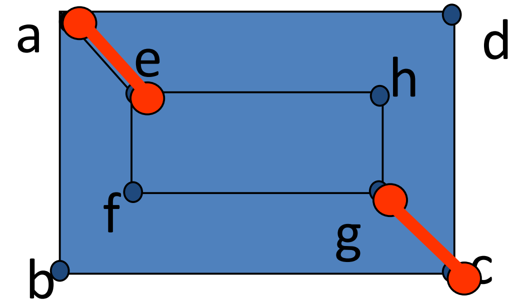
f(d) = 5

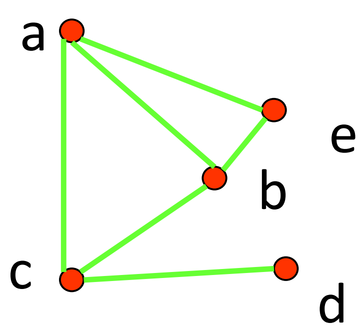
f(e) = 2

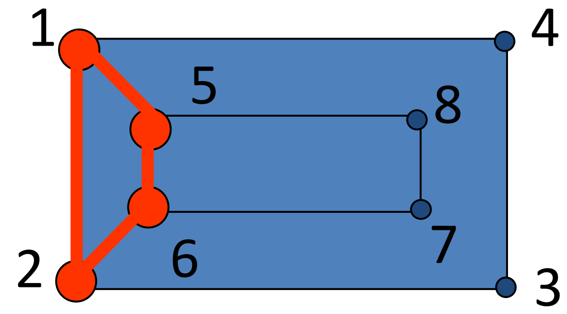
f(f) = 4

# Παραδείγματα – Άσκηση ΙΙ



# Γενίκευση ισομορφισμού Ι

- Δύο γράφοι είναι **1-ισομορφικοί** αν καθίστανται ισομορφικοί μετά την επανειλημμένη διάσπαση των αποκοπτουσών κορυφών.
- Τα τεμάχια ενός γράφου είναι ισομορφικά προς τις συνιστώσες ενός άλλου γράφου.

# Γενίκευση ισομορφισμού ΙΙ

- Αν ο γράφος αποτελείται από ένα τεμάχιο, τότε η έννοια της 1-ισομορφικότητας ταυτίζεται με την έννοια της ισομορφικότητας.
- **Θεώρημα:** Αν δύο γράφοι είναι 1-ισομορφικοί, τότε η σειρά και η μηδενικότητά τους είναι ίσες.

# Αντιστοιχία κύκλων

- Δύο γράφοι έχουν **αντιστοιχία κύκλων –** circuit correspondence, αν υπάρχει αμφιμονοσήμαντη αντιστοιχία ακμών και κύκλων, έτσι ώστε ένας κύκλος του πρώτου γράφου να έχει αντίστοιχο κύκλο στο δεύτερο γράφο, που αποτελείται από τις αντίστοιχες ακμές.

- <u>**Θεώρημα:**</u> Δύο 1-ισομορφικοί γράφοι έχουν αντιστοιχία κύκλων

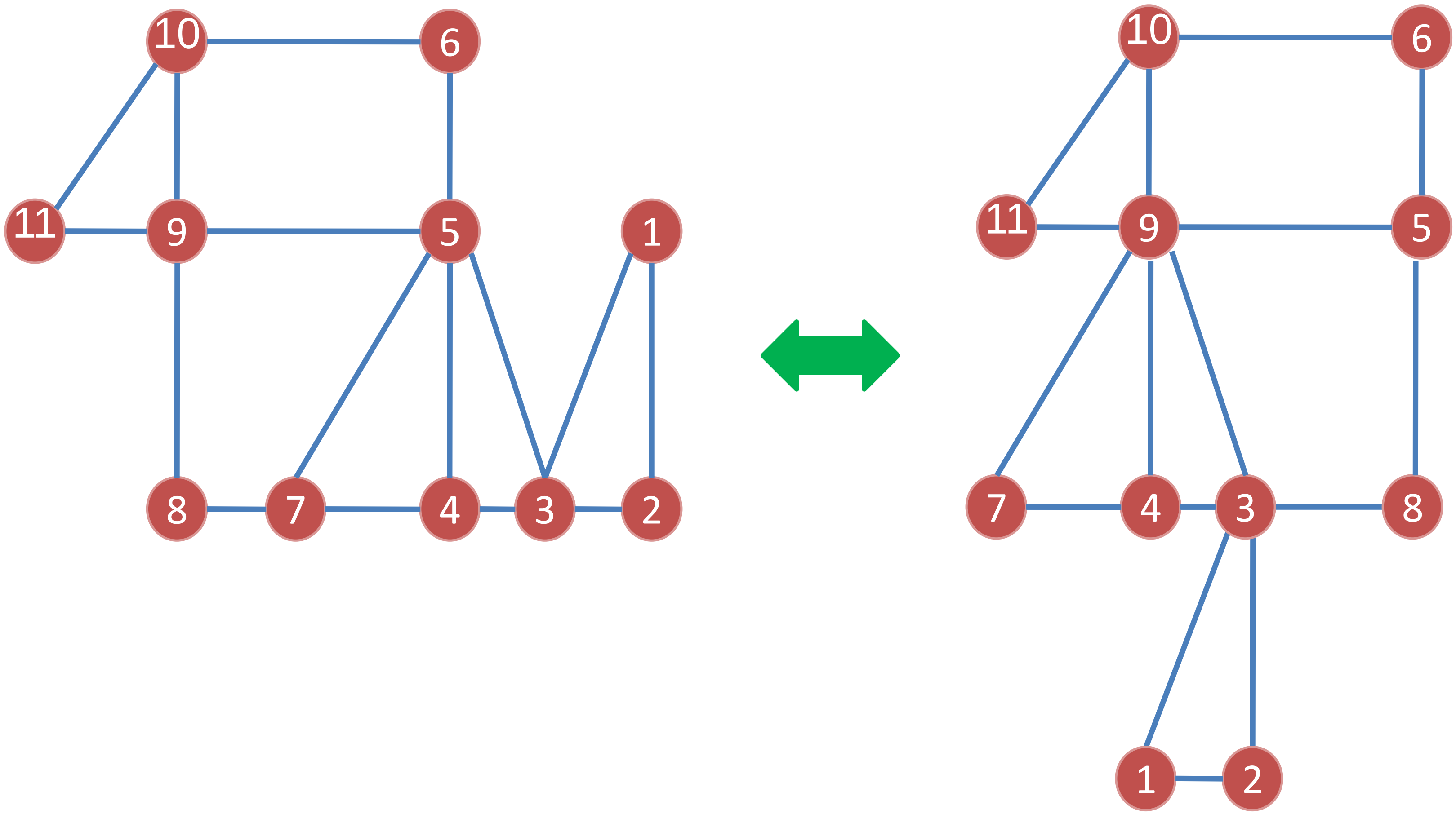
# 2-ισομορφισμός Ι

- Δύο γράφοι είναι 2-ισομορφικοί αν καθίστανται ισομορφικοί μετά την επανειλημμένη εφαρμογή μιας ή δύο από τις εξής πράξεις
  - διαδοχική διάσπαση αποκοπτουσών κορυφών
  - διαχωρισμός του ενός γράφου σε δύο ξένους υπογράφους, που έχουν 2 ζεύγη κοινών κορυφών και επανασύνδεση των υπογράφων ταυτοποιώντας τις κορυφές με διαφορετικό τρόπο

# 2-ισομορφισμός ΙΙ

- Δύο ισομορφικοί γράφοι είναι 1-ισομορφικοί, δύο 1-ισομορφικοί γράφοι είναι 2-ισομορφικοί. Το αντίθετο δεν ισχύει.
- **Θεώρημα (Whitney):** Δύο γράφοι είναι 2-ισομορφικοί αν και μόνον αν έχουν αντιστοιχία κύκλων

# Παράδειγμα Ι



# Αλγόριθμοι

- Αλγόριθμοι διάσχισης γράφων
  - BFS, αναζήτηση κατά πλάτος
  - DFS, αναζήτηση κατά βάθος
- Οι αλγόριθμοι αυτοί χρησιμοποιούνται
  - για την εύρεση αποστάσεων από κάποια κορυφή
  - για τη διαπίστωση αν γράφος είναι συνδεδεμένος
  - για την εύρεση (ισχυρά συνδεδεμένων) συνιστωσών
  - για την τοπολογική ταξινόμηση
  - για την εύρεση των σημείων άρθρωσης
  - για την εύρεση γεφυρών
  - για τη διαπίστωση κύκλων
  - για τη διαπίστωση επιπεδικότητας

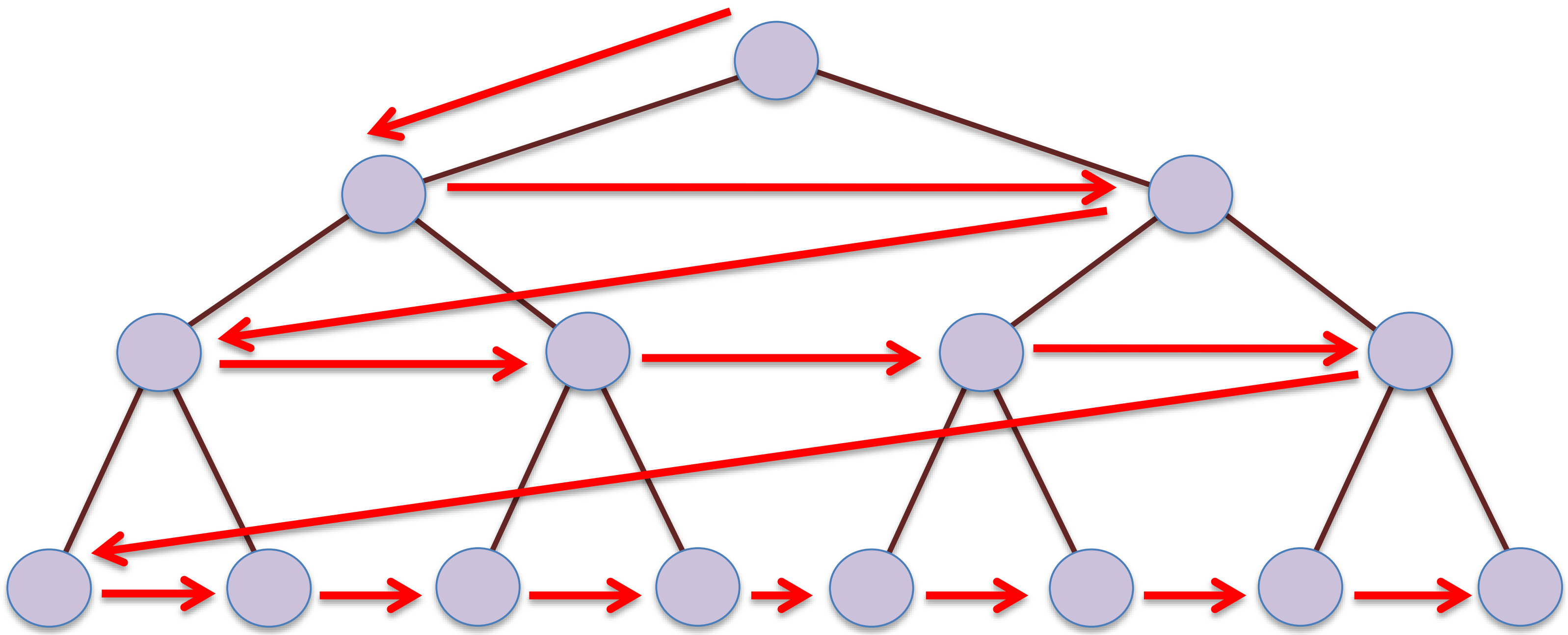
# BFS vs DFS

- BFS
  - 1959 Moore
  - Επισκεπτόμαστε τον πιο ρηχό κόμβο
  - Υλοποίηση με ουρά

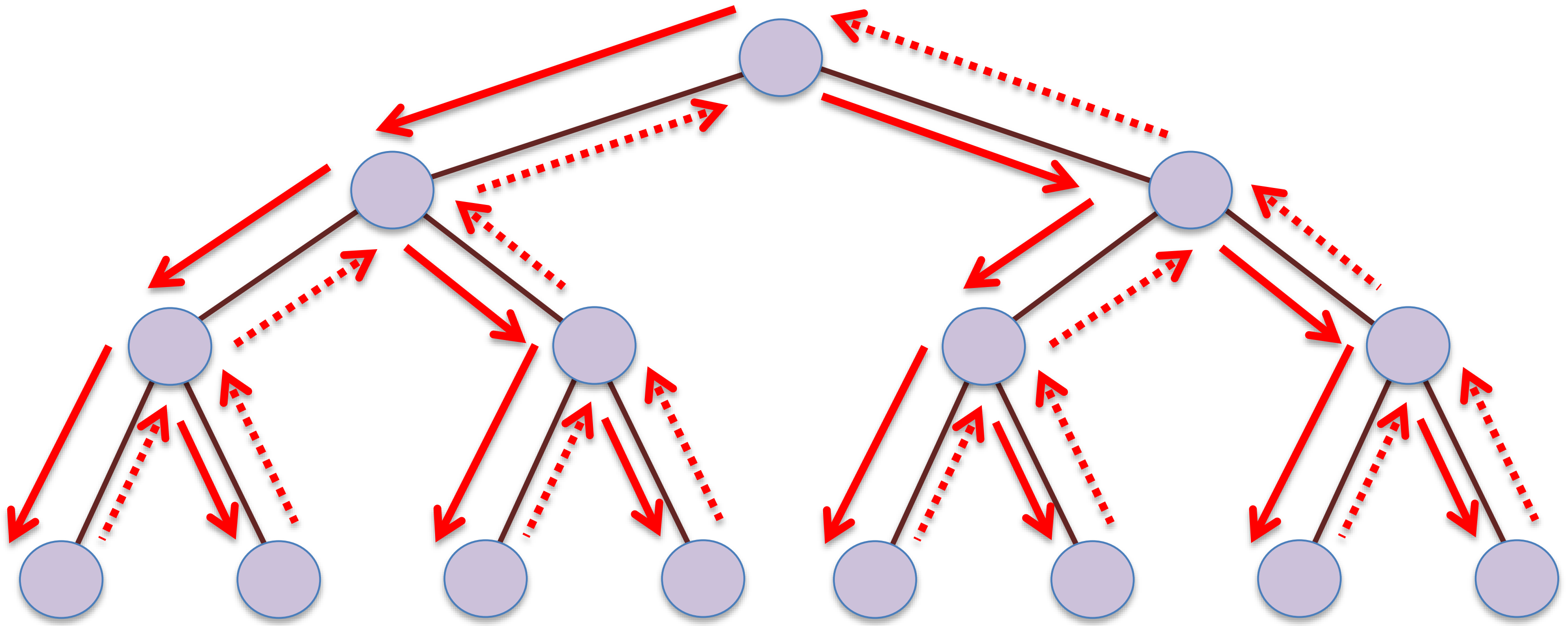
- DFS
  - 1973 Hopcroft-Tarjan
  - Επισκεπτόμαστε τον πιο βαθύ κόμβο
  - Υλοποίηση με στοίβα.

Η διαφορά έγκειται στη δομή δεδομένων!

# BFS σε δένδρο



# DFS σε δένδρο



Ταυτίζεται με την προδιατεταγμένη (preorder) διάσχιση

# BFS vs DFS

```
bfs(G)
  list L = empty
  tree T = empty
  choose a starting vertex x
  visit(x)
  while(L nonempty)
    remove edge (v,w) from
              beginning of L
    if w not visited
      add (v,w) to T
      visit(w)
```

```
dfs(G)
  list L = empty
  tree T = empty
  choose a starting vertex x
  visit(x)
  while(L nonempty)
    remove edge (v,w) from
              end of L
    if w not visited
      add (v,w) to T
      visit(w)
Visit ( vertex v)
    mark v as visited
    for each edge (v,w)
      add edge (v,w) to end of L
```

# BFS – Εύρεση αποστάσεων

**Είσοδος:** γράφος G με επιγραφές και κορυφή $x \in V$

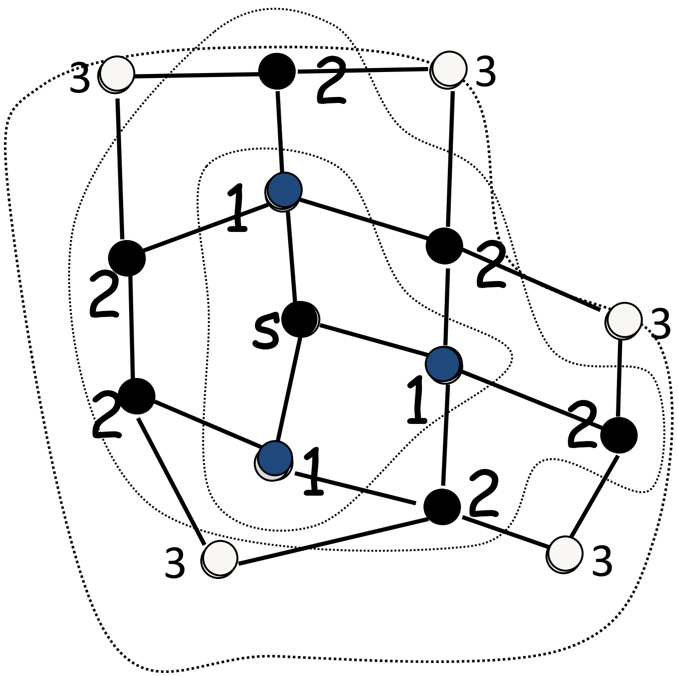
**Έξοδος:** οι αποστάσεις από την κορυφή x προς όλες τις κορυφές που είναι προσπελάσιμες από αυτήν

1. Θέτουμε $i \leftarrow 0$. Στην κορυφή x θέτουμε την επιγραφή i.
2. Βρίσκουμε όλες τις κορυφές χωρίς επιγραφές που είναι γειτονικές προς τουλάχιστον μία κορυφή με επιγραφή i. Αν δεν υπάρχει κάποια τέτοια κορυφή, τότε ο γράφος εξαντλήθηκε.
3. Θέτουμε την επιγραφή i+1 σε όλες τις κορυφές που βρέθηκαν στο Βήμα 2
4. Θέτουμε $i \leftarrow i+1$. Πηγαίνουμε στο Βήμα 2

# BFS – Εύρεση αποστάσεων, η ιδέα Ι

- Σε κάθε χρονική στιγμή υπάρχει ένα “μέτωπο” κορυφών που τις έχουμε ανακαλύψει, αλλά δεν τις έχουμε ακόμη επεξεργασθεί
- Λαμβάνουμε διαδοχικά τις κορυφές του “μετώπου” και ανακαλύπτουμε τους γείτονες δημιουργώντας ένα νέο “μέτωπο”

# BFS – Εύρεση αποστάσεων, η ιδέα ΙΙ




- Άσπρες κορυφές: μη μαρκαρισμένες και εκτός ουράς
- Γκρι κορυφές: μαρκαρισμένες και εντός ουράς
- Μαύρες κορυφές: μαρκαρισμένες και εκτός ουράς

# Αναλυτικά: BFS – αρχικοποίηση

```
procedure BFS(G:graph; s:node;
   var color:carray; dist:iarray; parent:parray);
  for each vertex u do
    color[u]:=white;
    dist[u]:=∞;
      parent[u]:=nil;
  color[s]:=gray; dist[s]:=0;
  init(Q); enqueue(Q, s);,
```

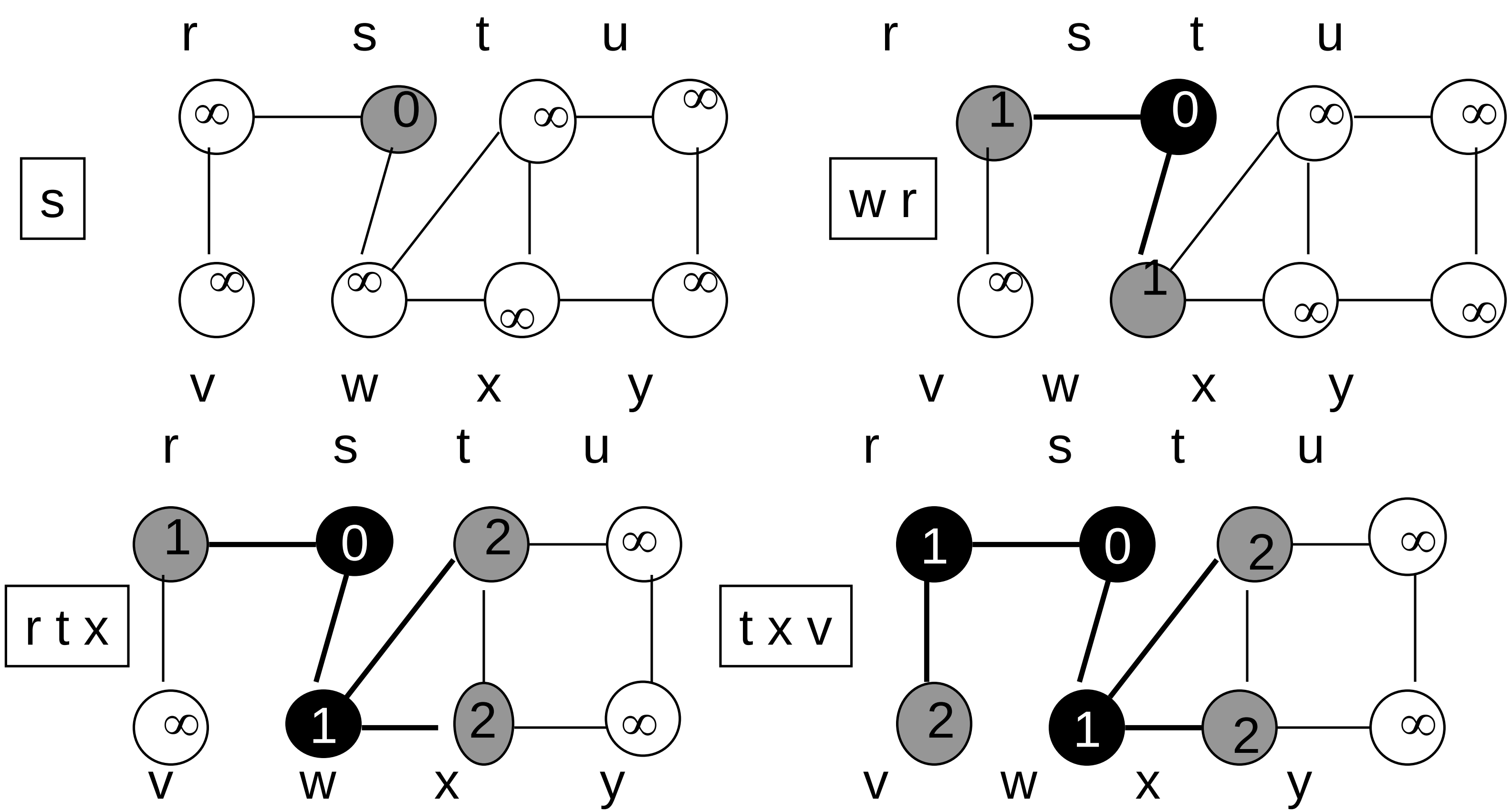
- **Θ(n)**

# Αναλυτικά: BFS – main

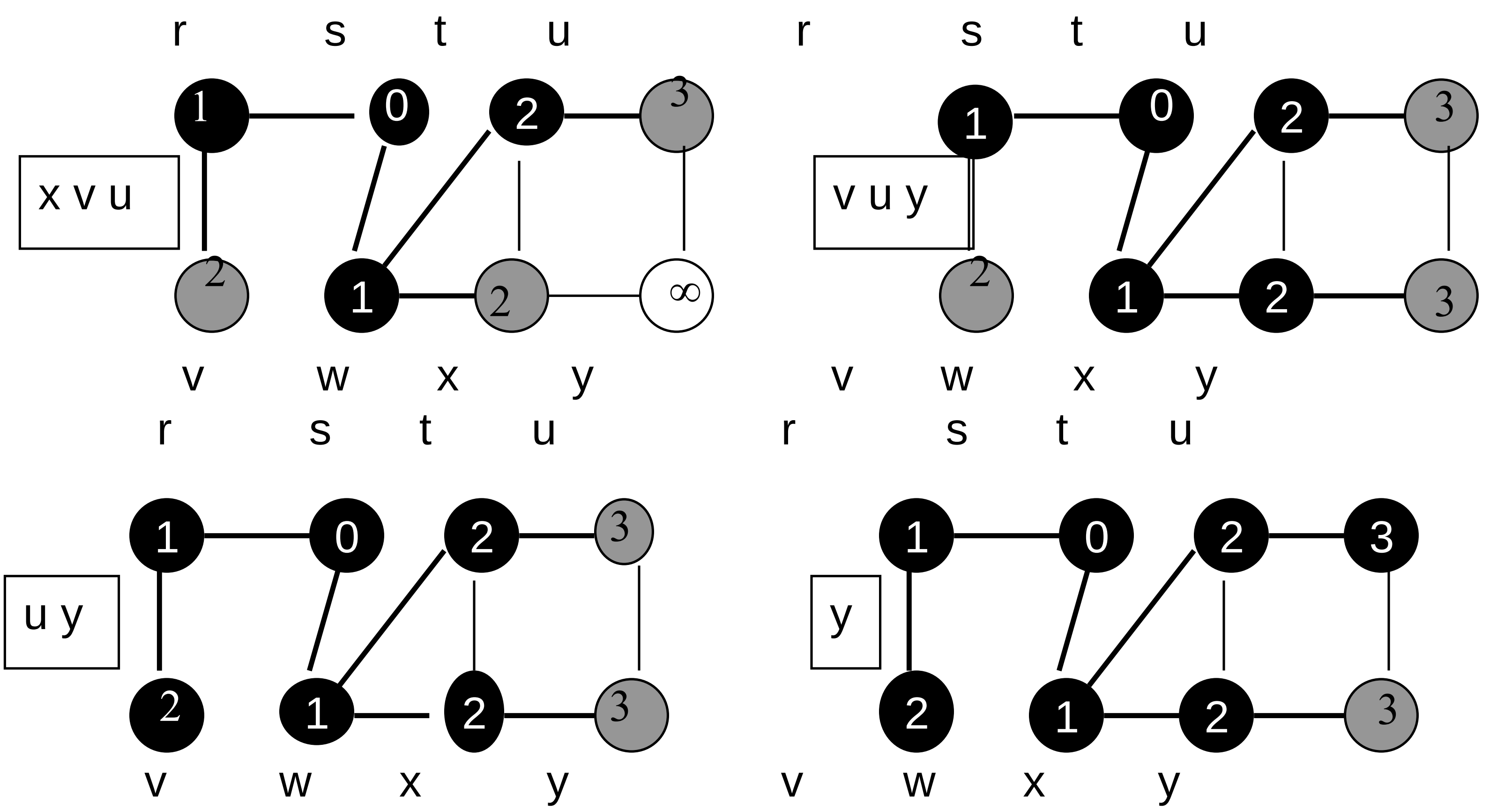
```
while not (empty(Q)) do
  u:=head(Q);
  for each v in adj[u] do
    if color[v]=white then
      color[v]:=gray;
          dist[v]:=dist[u]+1;
      parent[v]:=u;
      enqueue(Q,v);
  dequeue(Q); color[u]:=black;
end BFS
```

- **O(m)**

# BFS – Παράδειγμα I



# BFS – Παράδειγμα II




Τέλος, η κορυφή y βγαίνει από την ουρά και μαυρίζεται

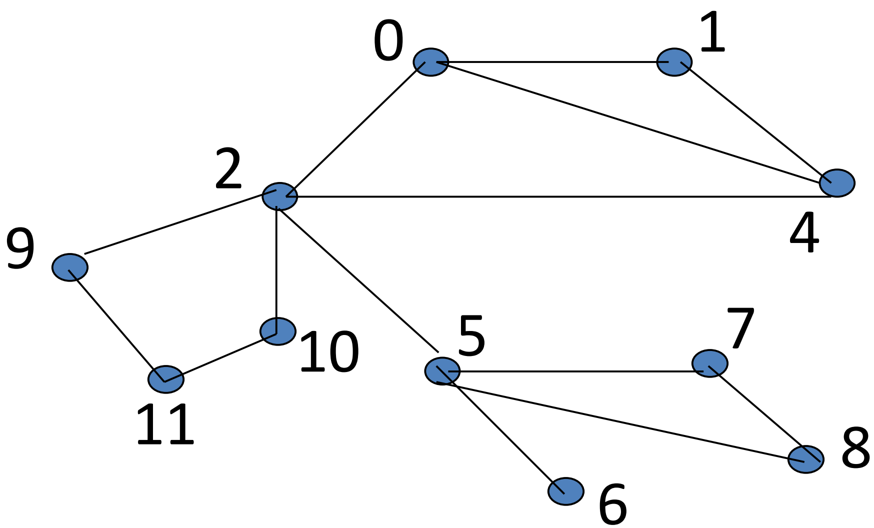
# Πολυπλοκότητα του BFS

- Αρχικοποίηση Θ(n).
- Κάθε κόμβος μπαίνει στην ουρά μία φορά (από άσπρος γίνεται γκρι) και η λίστα γειτνίασης διασχίζεται μία φορά.
- Το πολύ, όλες οι λίστες διασχίζονται.
- Κάθε ακμή λαμβάνεται δύο φορές, οπότε ο βρόχος επαναλαμβάνεται το πολύ 2|E| φορές.
- Χειρότερη περίπτωση O(n+m)
- Αν ο γράφος είναι υλοποιημένος με πίνακα γειτνίασης, τότε η πολυπλοκότητα είναι O(n2).

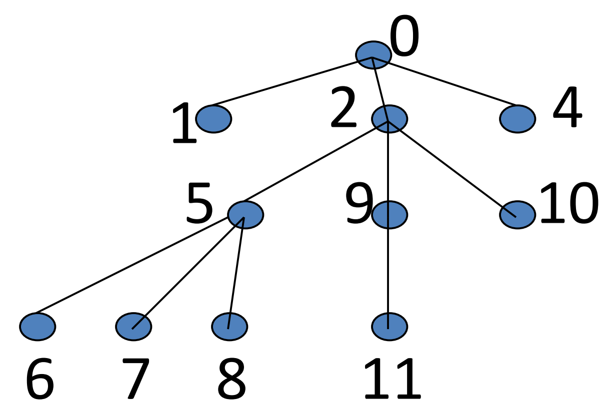
# BFS δένδρο I

- Όσες ακμές του γράφου G παρουσιάζονται στο BFS tree ονομάζονται **δενδρικές** –tree edges, οι υπόλοιπες ονομάζονται **διασταυρούμενες** –cross edges.

# BFS δένδρο II




Γράφος G




BFS δένδρο

# DFS – Αναζήτηση κατά βάθος

Αλγόριθμος DFS των Hopcroft-Tarjan (1973):

**Είσοδος:** γράφος G με επιγραφές και κορυφή $x \in V$

**Έξοδος:** σύνολο T δενδρικών κορυφών και αρίθμηση dfi(v)

1. Θέτουμε $T \leftarrow \emptyset$, $i \leftarrow 1$.
2. Για κάθε $v \in V$, θέτουμε $dfi(v) \leftarrow 0$
3. Για κάθε u με dfi(u) εκτελείται DFS(u).
4. Στην έξοδο δίνεται το σύνολο T.
5. **Διαδικασία** DFS(v)
6. Θέτουμε $dfi(v) \leftarrow i$, $i \leftarrow i+1$
7. Για κάθε $u \in N(v)$ εκτελούνται οι εντολές:

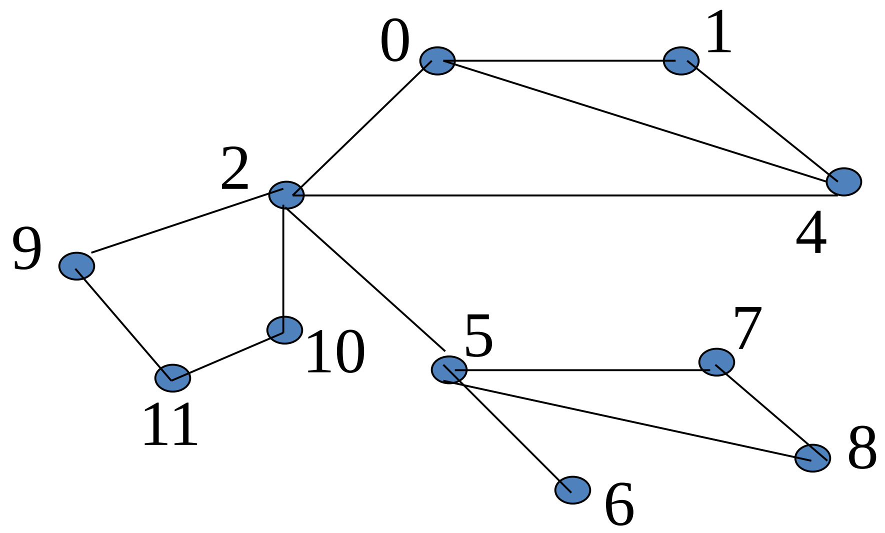
   Αν dfi(u)=0, τότε θέτουμε $T \leftarrow T \cup \{e\}$, όπου e=(u,v) μία μη χρησιμοποιημένη προσπίπτουσα ακμή, και $l(e) \leftarrow used$

   Καλούμε την DFS(u)

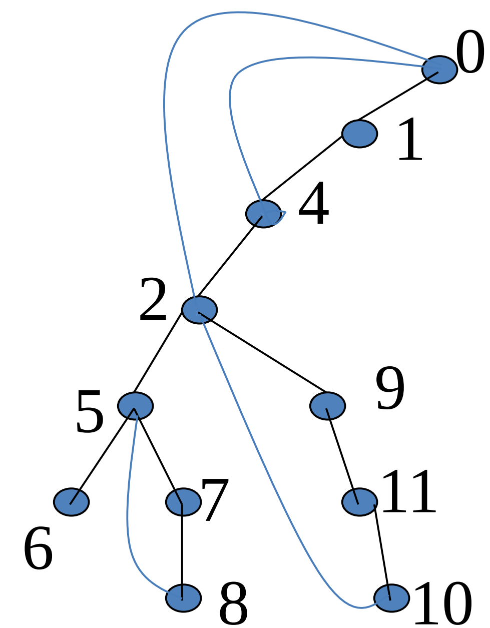
# Οπίσθιες ακμές Ι

- **Θεώρημα**: Κάθε οπίσθια ακμή (u,v) που προκύπτει κατά την αναζήτηση κατά βάθος (DFS) ενός μη κατευθυνόμενου γράφου ενώνει κορυφές που βρίσκονται σε σχέση απογόνου/προγόνου.

# Οπίσθιες ακμές ΙΙ




Γράφος G




DFS δένδρο

# Αναλυτικά: DFS

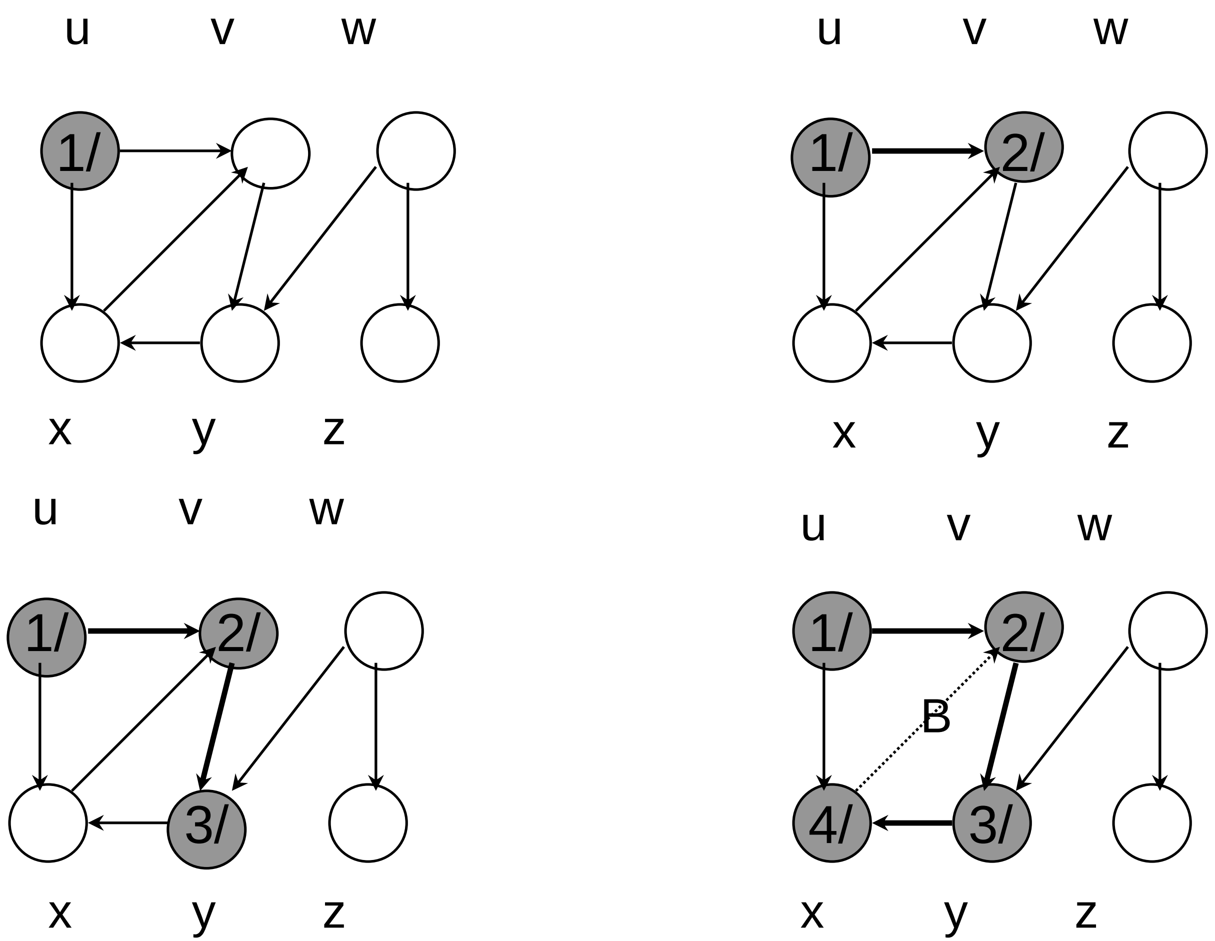
```
procedure DFS(G:graph; var color: carray; d,f:iarray;
    parent:parray);
    for each vertex u do
    color[u]:=white; parent[u]:=nil;
    time:=0;
    for each vertex u do
          if color[u]=white then
         DFS-Visit(u);
end DFS
```

- **Θ(n)**

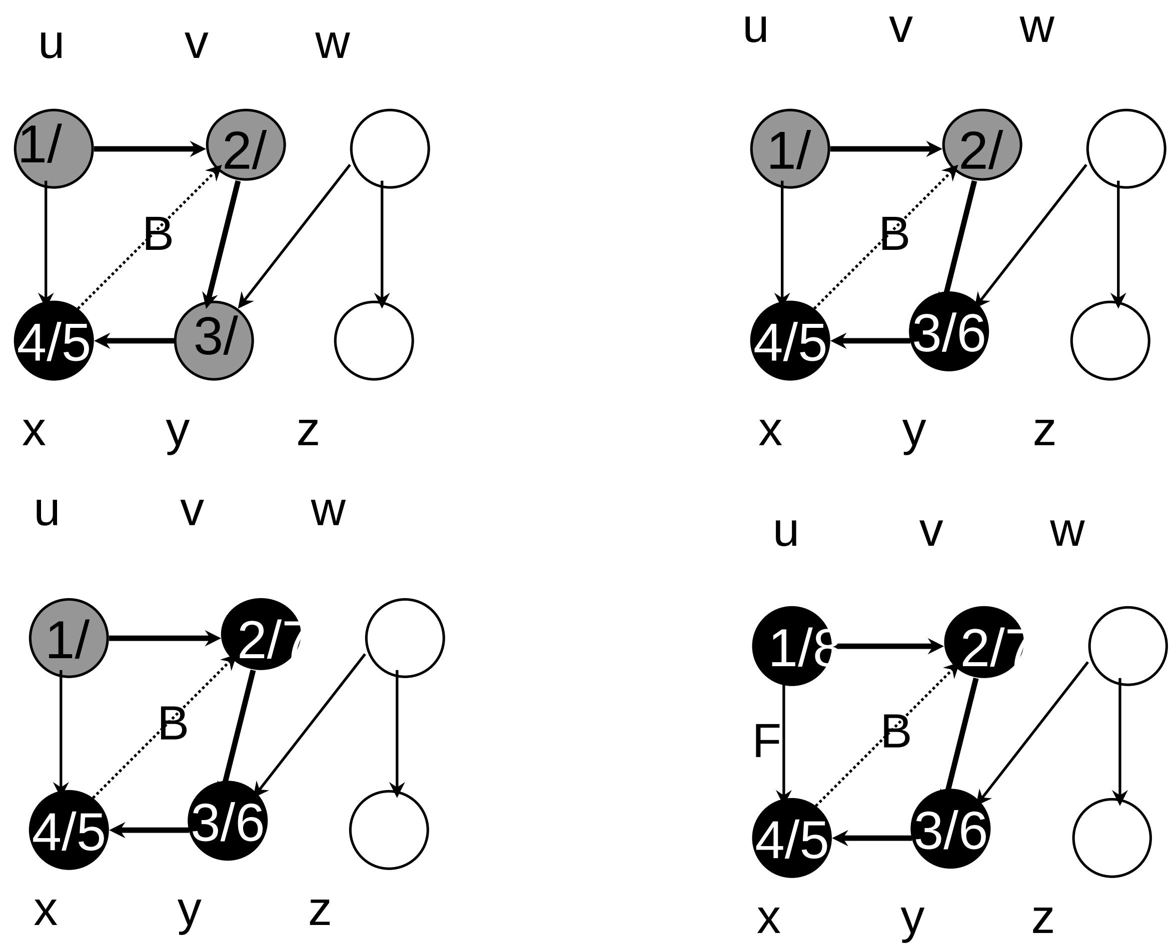
# Αναλυτικά: DFS-visit(u)

```
color[u]:=gray; time:=time+1; d[u]:=time
for each v in adj[u] do
   if color[v]=white then
      parent[v]:=u;  DFS-Visit(v);
      color[u]:=black; time:=time+1;
   f[u]:=time;
end DFS-Visit
```

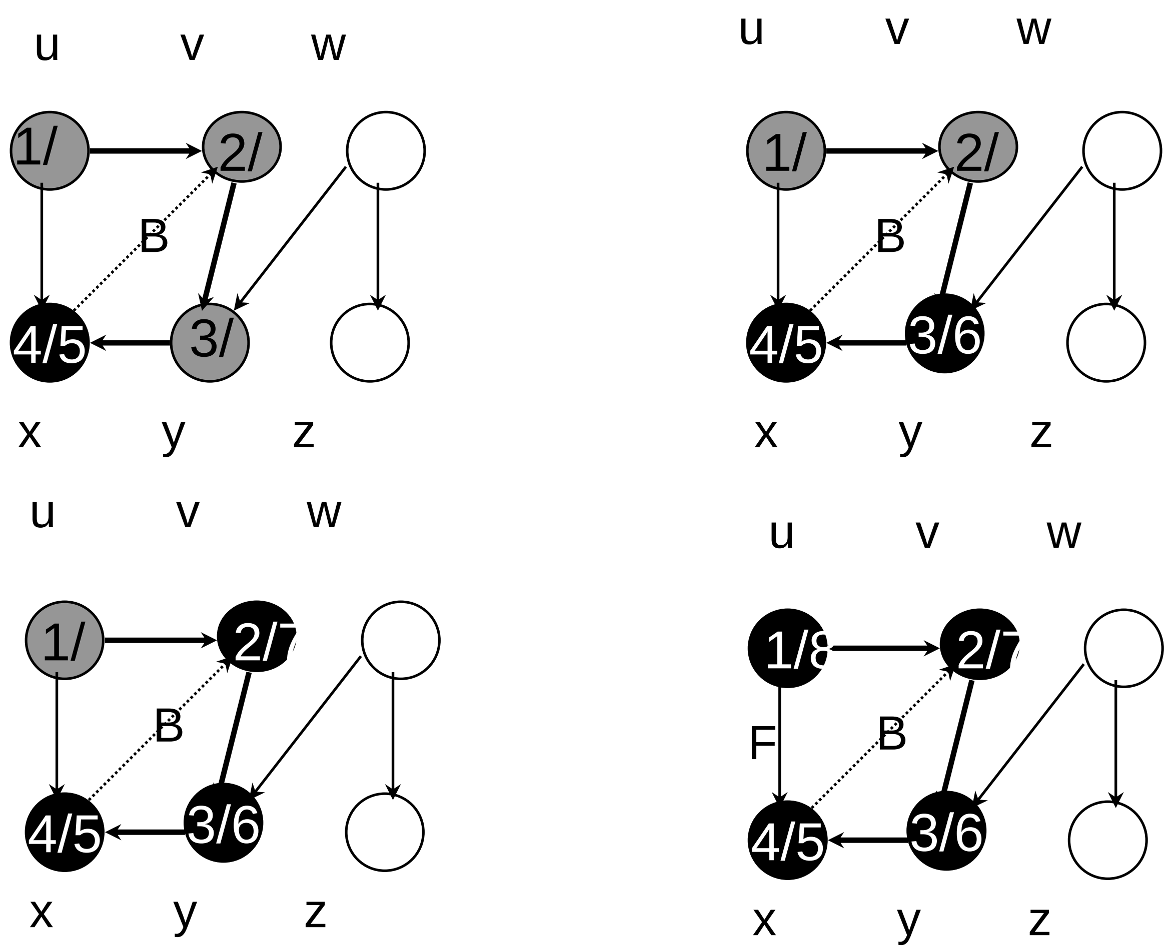
# DFS – Παράδειγμα I



# DFS – Παράδειγμα II



# DFS – Παράδειγμα III



# Πολυπλοκότητα του DFS

- Αρχικοποίηση Θ(n).
- Η DFS-visit καλείται μία φορά για κάθε κόμβο v.
- Ο βρόχος for εκτελείται dv φορές.
- Συνολικά η DFS-visit καλείται Θ(m) φορές.
- Χειρότερη περίπτωση Θ(n+m)

# Εφαρμογές του DFS I

- Ο γράφος G είναι συνδεδεμένος?
  Εκτελούμε DFS-Visit(v). Αν προσεγγίζουμε όλες τις κορυφές, τότε ΝΑΙ, αλλιώς ΟΧΙ. O(n+m)
- Ο γράφος G είναι δένδρο?
  Εκτελούμε DFS-Visit(v). Αν προσεγγίζουμε όλες τις κορυφές και δεν υπάρχουν οπίσθιες ακμές, τότε ΝΑΙ, αλλιώς ΟΧΙ. O(n)

# Εφαρμογές του DFS II

- Εύρεση τεμαχίων.
  Εκτελούμε DFS. Αναθέτουμε στις κορυφές ενός τεμαχίου ένα id. $\Theta(n+m)$
- Εύρεση κύκλου.
  Εκτελούμε DFS. Αν υπάρχουν οπίσθιες ακμές, τότε ΝΑΙ, αλλιώς ΟΧΙ. $O(n)$ διότι προσεγγίζονται το πολύ $n$ ακμές. Προσοχή: οι ακμές $(u,v)$ και $(v, u)$ δεν αποτελούν κύκλο.

# Αλγόριθμος εύρεσης τεμαχίων με DFS I

- Αρκεί να εντοπιστούν οι αποκόπτουσες κορυφές ως εξής:

1. Αν μία αποκόπτουσα κορυφή v είναι η ρίζα του δένδρου DFS, τότε η v πρέπει να έχει περισσότερο από ένα γιο.
2. Αν μία αποκόπτουσα κορυφή v δεν είναι ρίζα, τότε πρέπει η v να έχει ένα γιο s, του οποίου κάποιος απόγονος (συμπεριλαμβανομένου του s) να συνδέεται με έναν πρόγονο της v μέσω 1 οπίσθιας ακμής το πολύ.

# Αλγόριθμος εύρεσης τεμαχίων με DFS II

Για κάθε κορυφή v ορίζεται εκτός από τη d(v) και μία επιπλέον μεταβλητή, η l(v), που δηλώνει τη μικρότερη από τις επιγραφές d(v) και d(s), όπου s είναι είτε απόγονος της v, είτε πρόγονος μέσω μιας το πολύ οπίσθιας ακμής, που ενώνει τον πρόγονο αυτό με έναν απόγονο της v.

Άρα η μέγιστη τιμή που μπορεί να πάρει η l(v) είναι d(v).

Άρα για να ισχύει το 2, πρέπει $l(s) \geq d(v)$

# Υπολογισμός παραμέτρου l(v)

Αρχικά: l(v)=d(v).

Ο υπολογισμός της l(v) γίνεται θέτοντας την τιμή της ίση με το ελάχιστο στοιχείο του συνόλου:

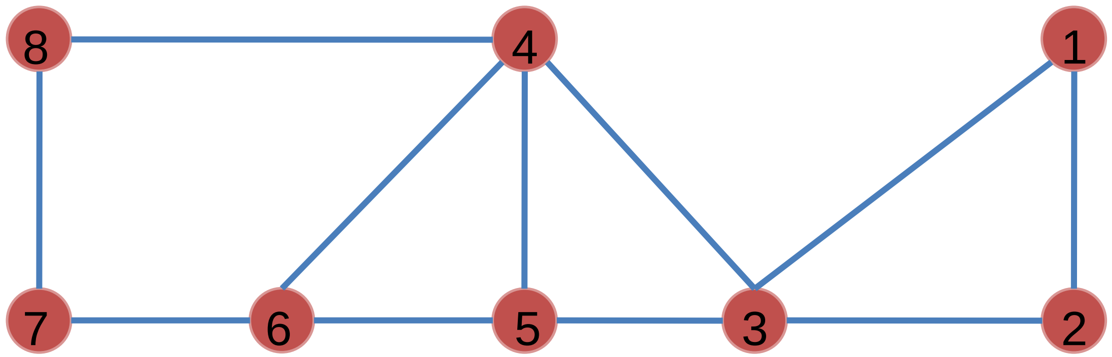
$\{d(v)\} \cup \{l(s) \mid s$ γιος της $v\} \cup \{d(w) \mid (s,w)$ μία οπίσθια ακμή$\}$

Η l(v) ενημερώνεται όποτε προσπελάζεται ένας γιος s, τέτοιος ώστε $l(v) \geq l(s)$ ή όποτε βρίσκεται μία οπίσθια ακμή. Αυτό μπορεί να επιτευχθεί με τη βοήθεια μιας στοίβας.

Αποκόπτουσες κορυφές είναι όσες $l(u) \geq d(v)$

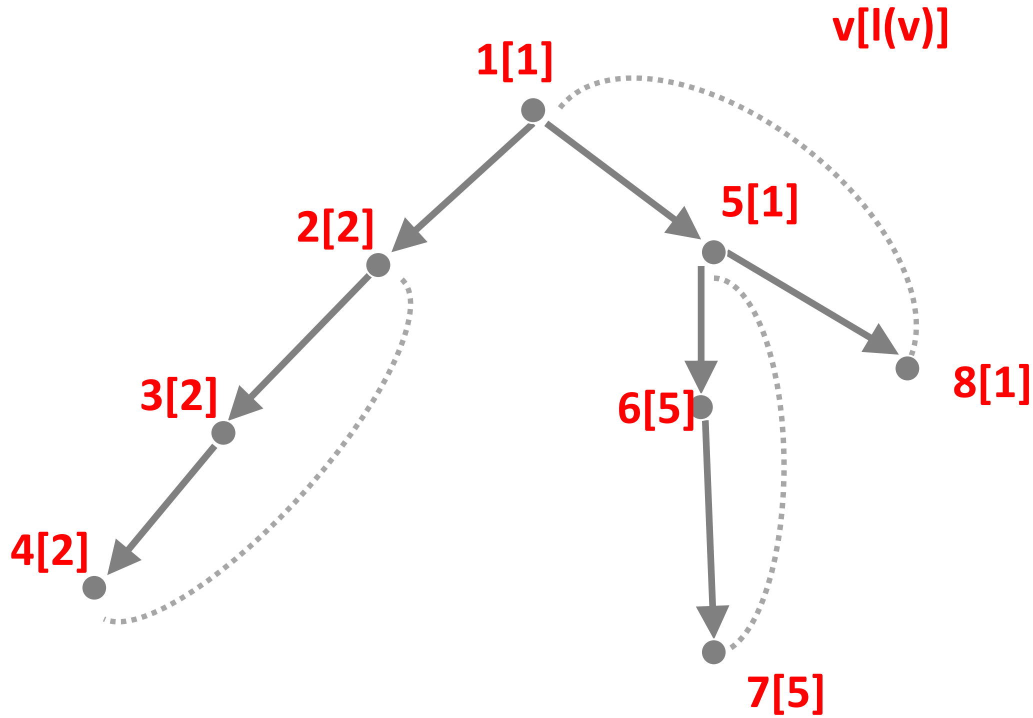
# Παράδειγμα Ι

Ξεκινώντας από την κορυφή 1,
να δοθούν τα dfi και l για όλες τις κορυφές



# Παράδειγμα ΙΙ

- Να εξηγηθούν οι τιμές dfi και l.



# Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Ιωάννης Μανωλόπουλος. «Αλγοριθμική Θεωρία Γράφων. Συνδεσμικότητα». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: http://eclass.auth.gr/courses/OCRS264/.

# Σημείωμα Αδειοδότησης

# Τέλος ενότητας

Επεξεργασία: Ανδρέας Κοσματόπουλος
Θεσσαλονίκη, Μάρτιος 2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

# Σημειώματα

# Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

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

μαζί με τους συνοδευόμενους υπερσυνδέσμους.