

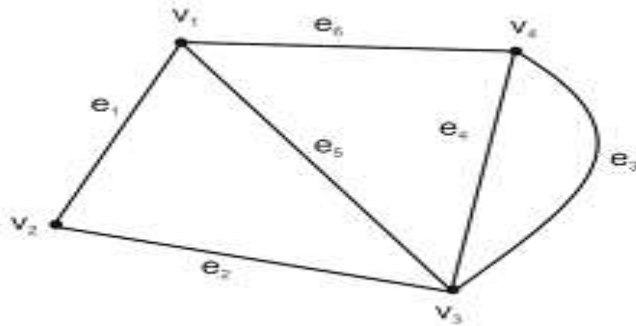
Γράφοι, BFS & DFS, Συνεκτικές Συνιστώσες

Τι είναι Γράφος

Ως γράφος $G=(V,E)$ ορίζεται ένα σύνολο κορυφών V που συνδέονται μεταξύ τους με ένα σύνολο ακμών E .

Τι είναι Γράφος

Ως γράφος $G=(V,E)$ ορίζεται ένα σύνολο κορυφών V που συνδέονται μεταξύ τους με ένα σύνολο ακμών E .



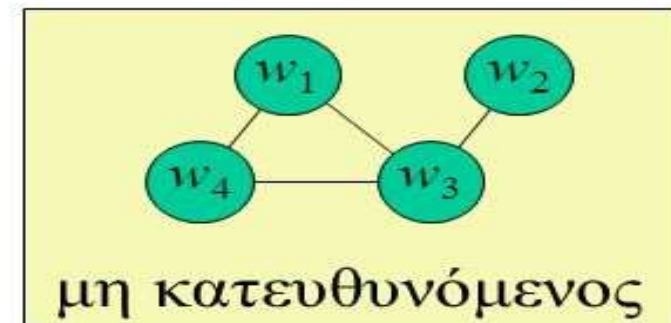
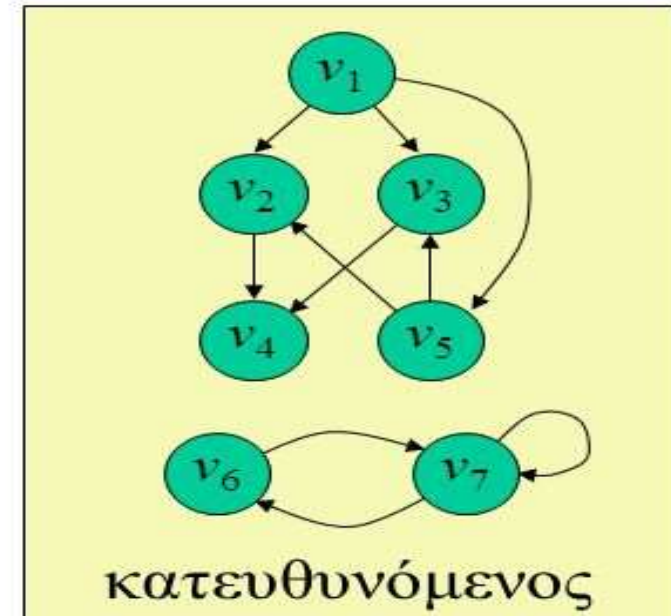
$$V=\{v_1,v_2,v_3,v_4\}$$

$$E=\{e_1,e_2,e_3,e_4,e_5,e_6\}$$

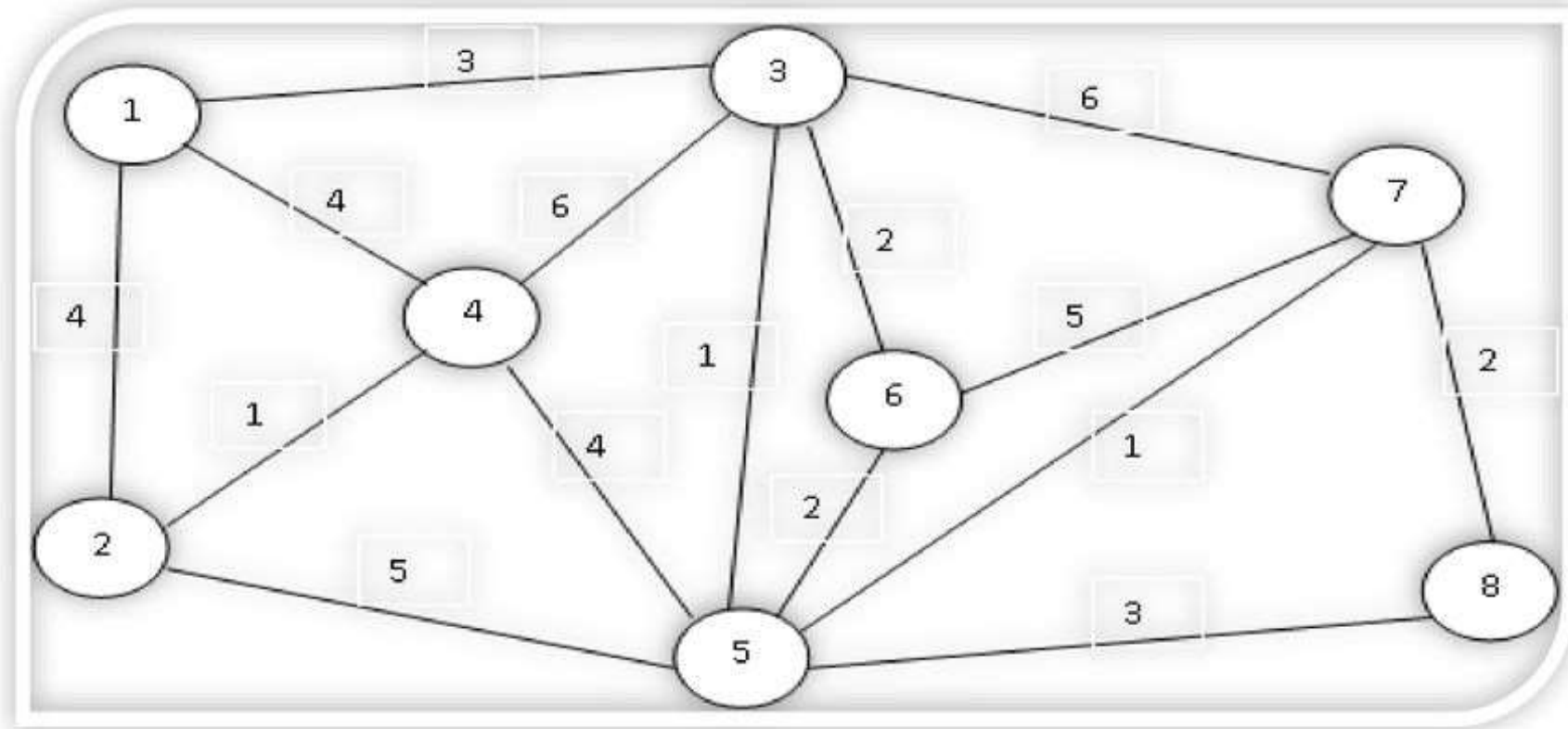
Γράφοι

- Είδη γράφων
 - Κατευθυνόμενος (directed) γράφος, αν $(v_1, v_2) \in E$ είναι διατεταγμένο ζεύγος
 - Μη κατευθυνόμενος (undirected) γράφος, αν (v_1, v_2) και (v_2, v_1) ταυτόσημα
- Ορισμοί
 - Δύο κορυφές v_1 και v_2 ονομάζονται γειτονικές (adjacent) αν $(v_1, v_2) \in E$

(ii)



Γράφος με βάρη



Αναζήτηση στους γράφους

- Βασικός αλγόριθμος

- Αναζήτηση κατά πλάτος

- Αναζήτηση κατά βάθος

- Συνεκτικές Συνιστώσες

- Αλγόριθμος εύρεσης συνεκτικών
συνιστωσών

Διάσχιση (αναζήτηση) στους γράφους

Φεύγοντας από μια δεδομένη αρχική κορυφή s , **διάσχιση** ενός γράφου G , είναι η αναζήτηση των διαδόχων του s .

Βασική ιδέα του αλγορίθμου

1. **Μαρκάρουμε** την κορυφή s
2. Προωθούμε τις μάρκες
Όσο βρίσκουμε τόξο (x, y) , με το κόμβο x μαρκαρισμένο και τον επόμενο του y **αμαρκάριστο** τότε μαρκάρουμε τον κόμβο y .

Αναζήτηση στους γράφους (**Δομές δεδομένων**)

**κωδικοποίηση του G / πίνακας $Mark$ /
σύνολο Z**

- Ο γράφος G είναι κωδικοποιημένος
με λίστες γειτνίασης

- Ο πίνακας $Mark[n]$ δείχνει τις
κορυφές που έχουμε ήδη επισκεφθεί

- Το σύνολο Z , περιέχει κορυφές των
οποίων απομένουν ακόμη διάδοχοι που δεν
έχουμε επισκευθεί.

ΓΕΝΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ ΑΝΑΖΗΤΗΣΗΣ

Αρχικοποίησε $\text{Mark}[n]$ σε false

$\text{Mark}[s] = \text{true}$

$Z = \{s\}$

Repeat

Πάρε ένα κόμβο x από το Z

Αν όλοι οι διάδοχοι του x έχουν εξεταστεί τότε
 $Z = Z - \{x\}$

Αλλιώς

Πάρε έναν κόμβο y διάδοχο του x

Αν y δεν είναι μαρκαρισμένος τότε

μάρκαρε y

Θέσε y μέσα στο Z

Until $Z = \emptyset$

ΑΝΑΖΗΤΗΣΗ ΚΑΤΑ ΠΛΑΤΟΣ (BFS: Breadth First Search)

Z κωδικοποιημένο με μια ουρά

Διάδοχοι του s



Διάδοχοι διαδόχων του s



Εξάπλωση στίγματος λαδιού

Διάδοχοι διαδόχων ... του s

Αρχικοποίησε $Mark[n]$ σε false

$Mark[s] = true$

Θέσε s στην ουρά

Repeat

Πάρε x από την κεφαλή της ουράς

Για κάθε διάδοχο y του x

Αν y μη μαρκαρισμένος

μάρκαρε y

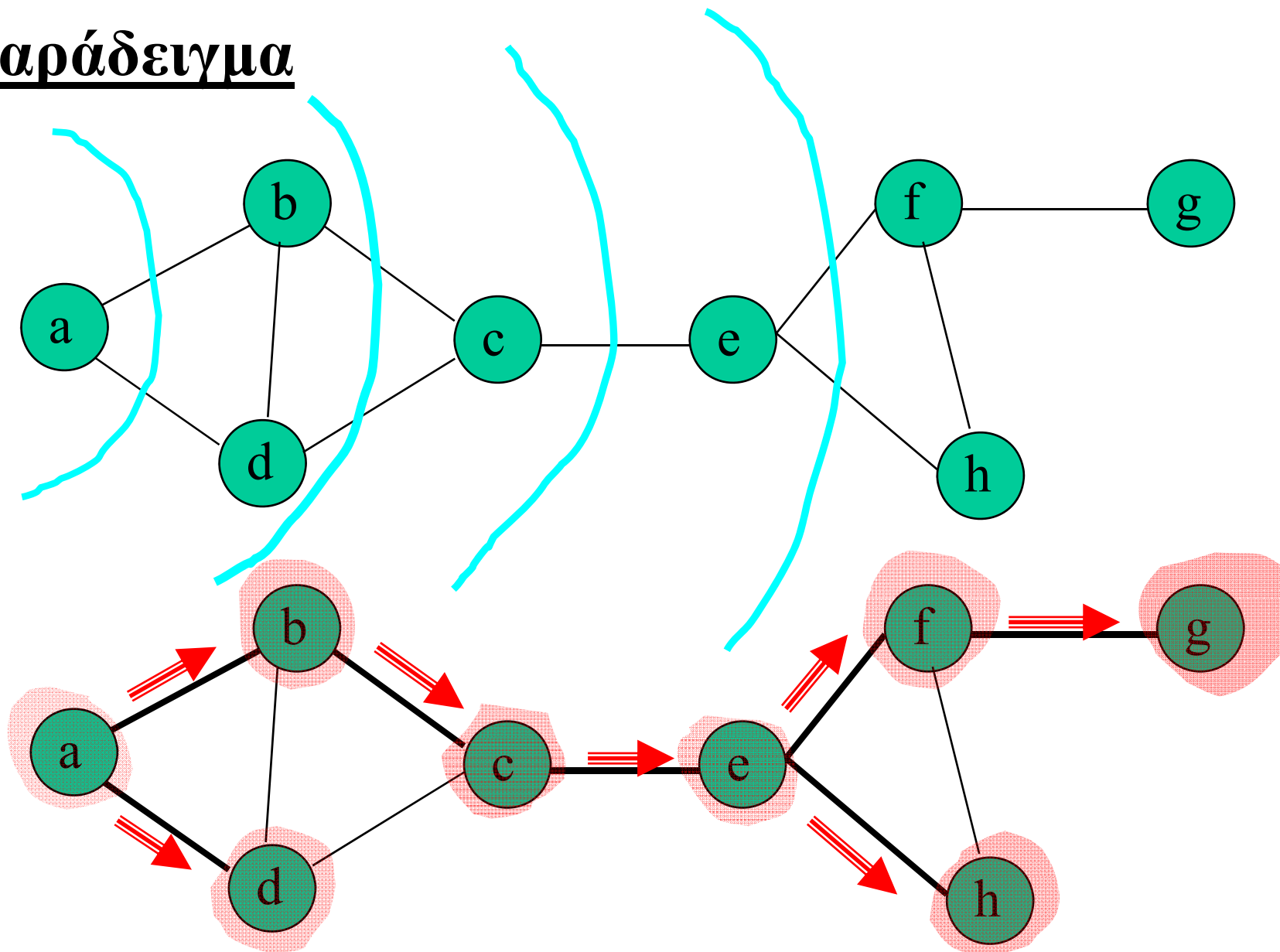
Θέσε y στο τέλος της ουράς

Until $Z = \emptyset$

(****)

ΑΝΑΖΗΤΗΣΗ ΚΑΤΑ ΠΛΑΤΟΣ: BFS

παράδειγμα



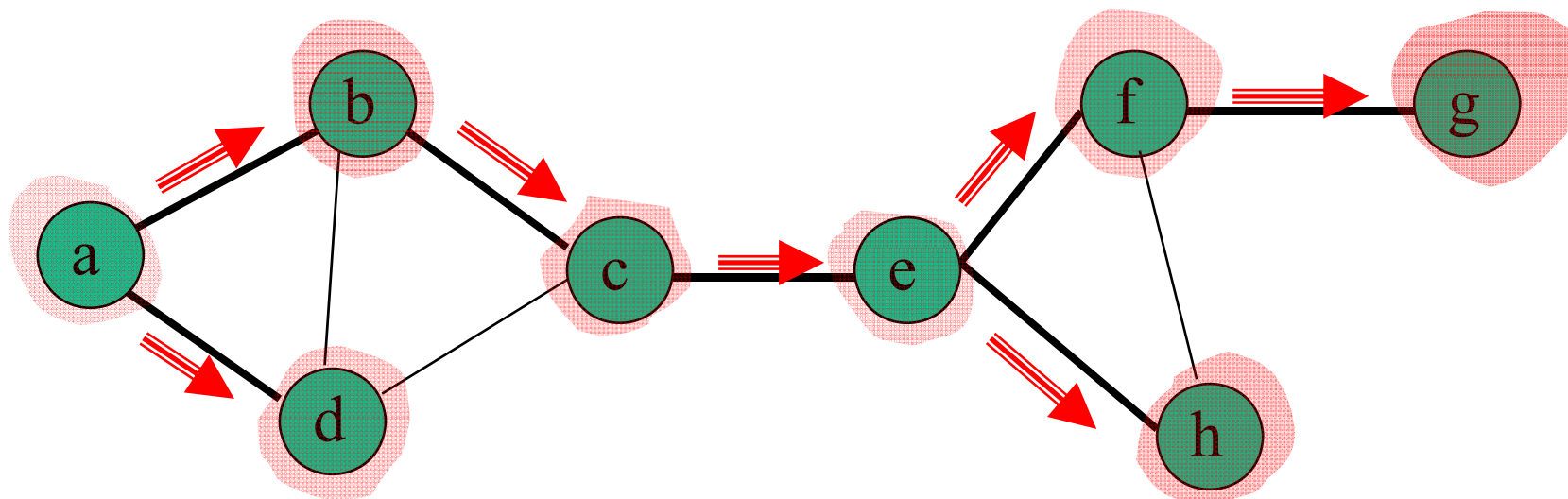
ΑΝΑΖΗΤΗΣΗ ΚΑΤΑ ΠΛΑΤΟΣ: BFS

Σειρά μαρκαρίσματος:

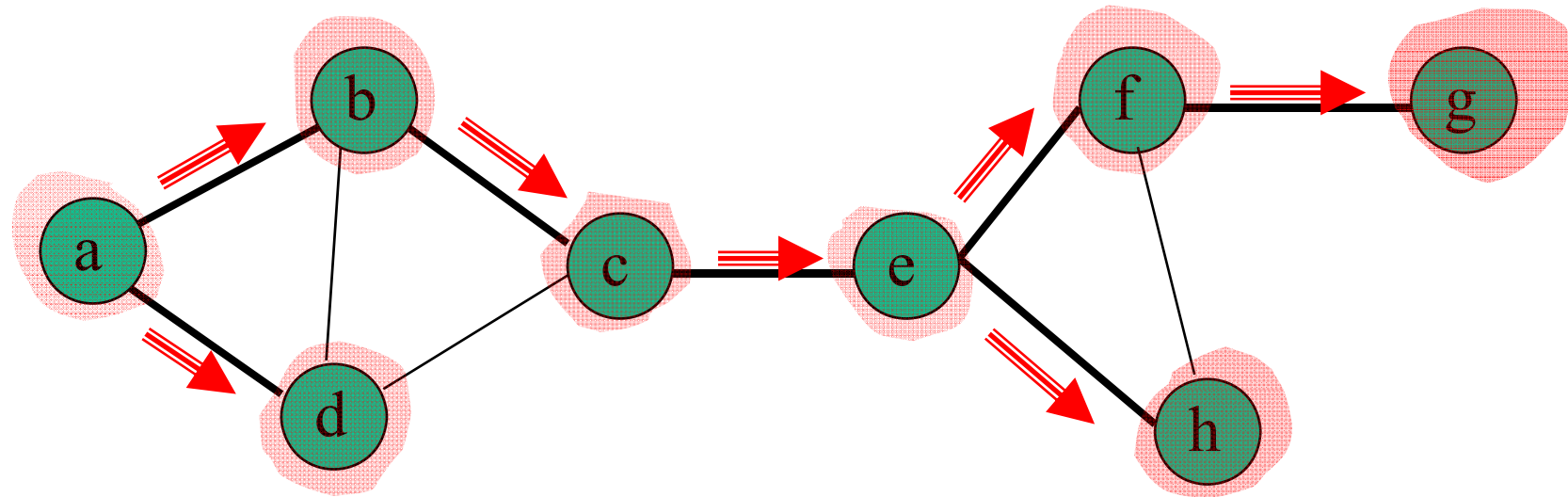
a:1, b:2, d:3, c:4, e:5, f:6, h:7, g:8

Εξέλιξη της ουράς Z:

a, db, cd, c, e, hf, gh, g, \emptyset



ΚΑΤΕΥΘΥΝΟΜΕΝΟ ΔΕΝΔΡΟ επίσκεψης με BFS



Κατευθυνόμενο δένδρο με την BFS ξεκινώντας απο τον κόμβο a.

- Κάθε κόμβος έχει ένα μόνο πατέρα.
- Πατέρας της ρίζας το \emptyset

- Ο κόμβος x δεν επιστρέφει στο σύνολο Z
- Η επανάληψη εξετάζει όλους τους διαδόχους ενός κόμβου

$$\sum_{x=1}^n (d^+(x) + 1) = n + \sum_{x=1}^n d^+(x) = n + m$$

Πλυπλοκότητα BFS: $\Theta(n+m)$. Γενικά $m > n$
και επομένως: **$O(m)$**

ΑΝΑΖΗΤΗΣΗ ΚΑΤΑ ΒΑΘΟΣ (DFS: Depth First Search)

*Το Z είναι κωδικοποιημένο με **στοίβα***

Η Αναζήτηση προχωρεί ως βολίδα, πηγαίνοντας όσο το δυνατόν **μακρύτερα** στο μονοπάτι με αρχή τον κόμβο s , πριν επιστρέψει σε άλλους κόμβους.

ΑΝΑΖΗΤΗΣΗ ΚΑΤΑ ΒΑΘΟΣ (DFS)

Αρχικοποίησε **Mark[n]** σε **false**

Mark[s] = true

Θέσε **s** στη στοίβα (**Z, s**)

Repeat

Πάρε **x** (απο την κορυφή της στοίβας **Z**)

Αν όλοι οι διάδοχοί του είναι **μαρκαρισμένοι**

Διέγραψε τον **x** απο την **Z**

Αλλιώς

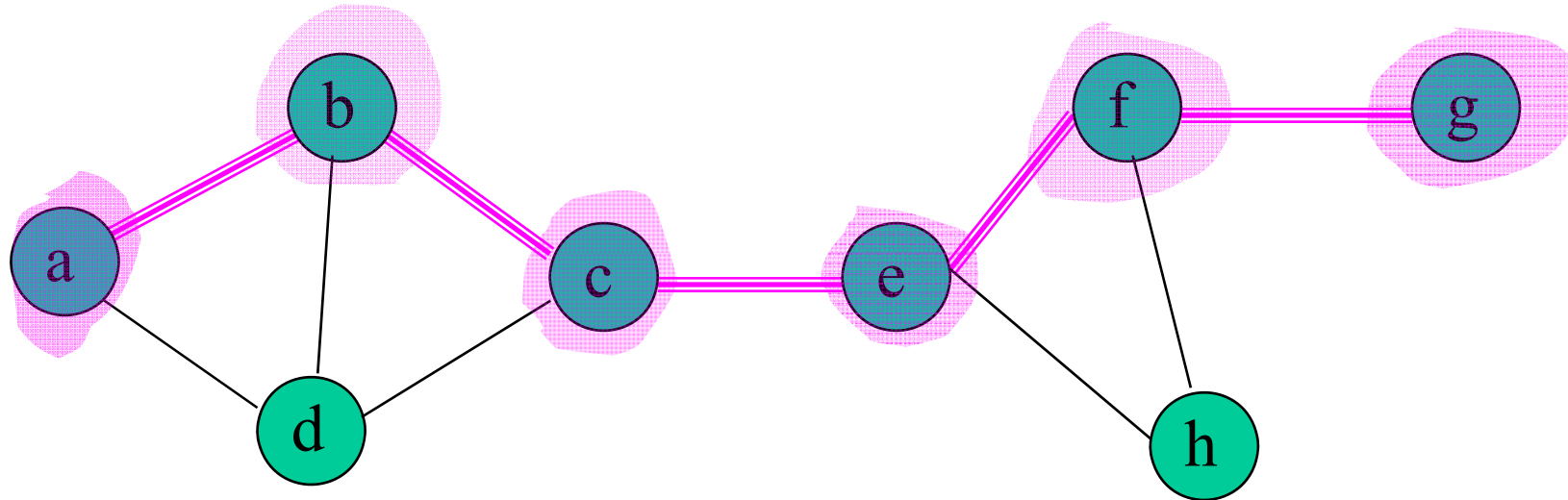
Πάρε ένα **διάδοχο** **y** του **x**, **μη μαρκαρισμένο**

Μάρκαρε τον κόμβο **y**

Θέσε τον κόμβο **y** στη στοίβα **Z**

Until **Z = ∅** (****)

παράδειγμα



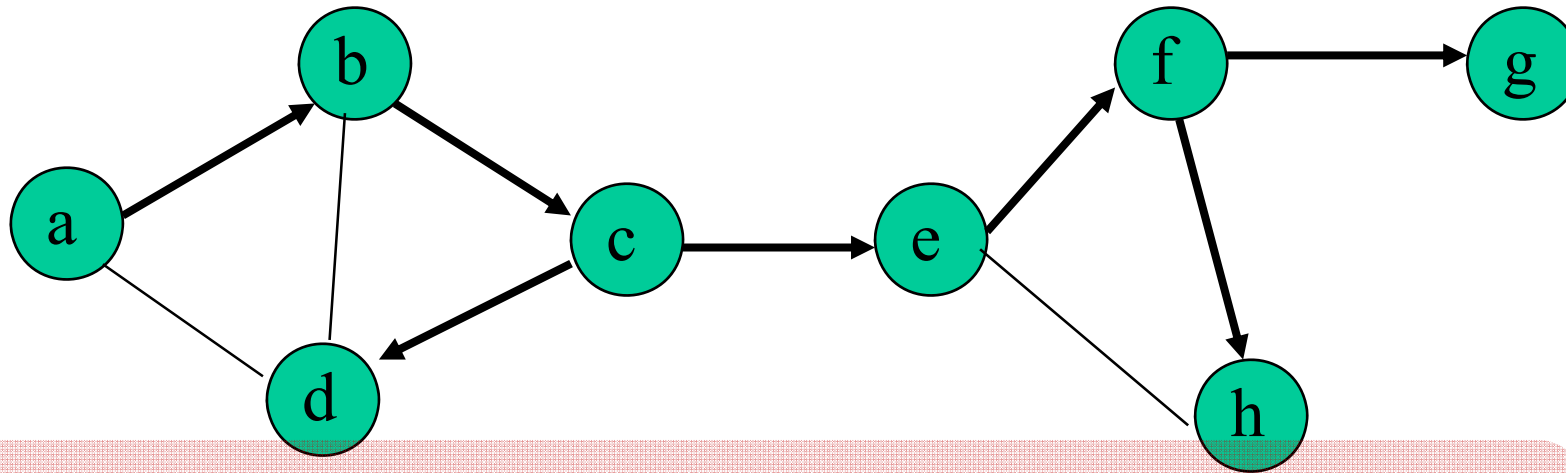
Σειρά μαρκαρίσματος:

$a:1, b:2, c:3, e:4, f:5, g:6, h:7, d:8$

Εξέλιξη της στοίβας Z:

a, ab, abc, abce, abcef, abcefg, abcef, abcefh, abcef,
abce, abc, abcd, abc, ab, a, \emptyset

ΚΑΤΕΥΘΥΝΟΜΕΝΟ ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ με την DFS



Κατευθυνόμενο δένδρο ξεκινώντας απο τον κόμβο a.

Πολυπλοκότητα $O(n+m)$

όπου m =αριθμός πλευρών του γράφου

1) Ελάχιστα μονοπάτια σε αριθμό τόξων

Χρησιμοποιώντας τη Αναζήτηση κατά πλάτος

- Χρήση ενός πίνακα: **απόσταση[]**
- Αρχικοποίηση: **απόσταση[s]=0**
- Θέσε **απόσταση[y]=απόσταση[x]+1**,
/* μετά το μαρκάρισμα του y διαδόχου του x */

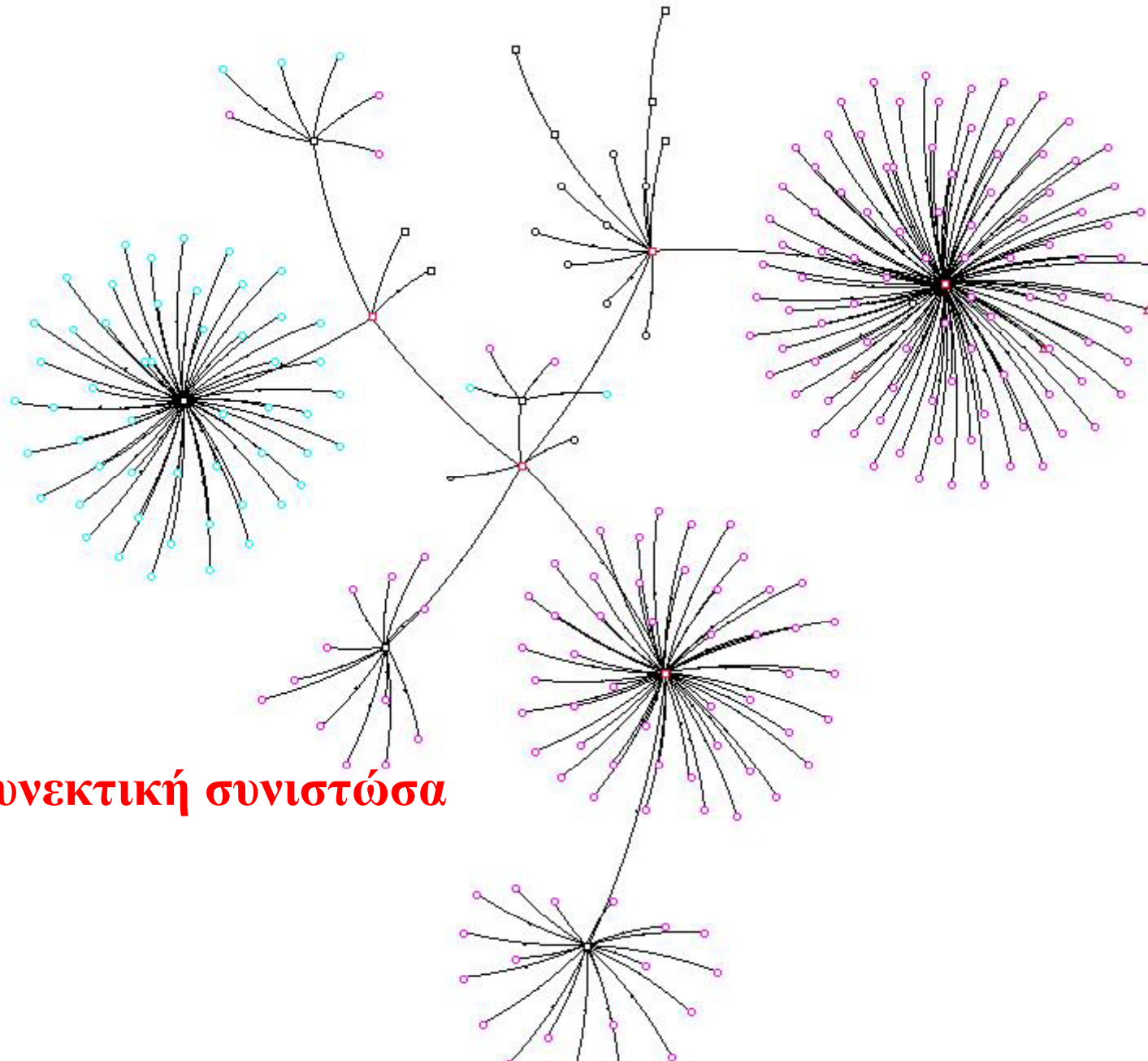
2) Ύπαρξη κλειστών μονοπατιών (κύκλων)

Χρησιμοποιώντας την Αναζήτηση κατά βάθος

Ένας γράφος είναι **συνεκτικός** αν υπάρχει ένα μονοπάτι που συνδέει δύο οποιουσδήποτε κόμβους. *Ο γράφος τότε αποτελείται από ένα μόνο bloc.*

Διαφορετικά:

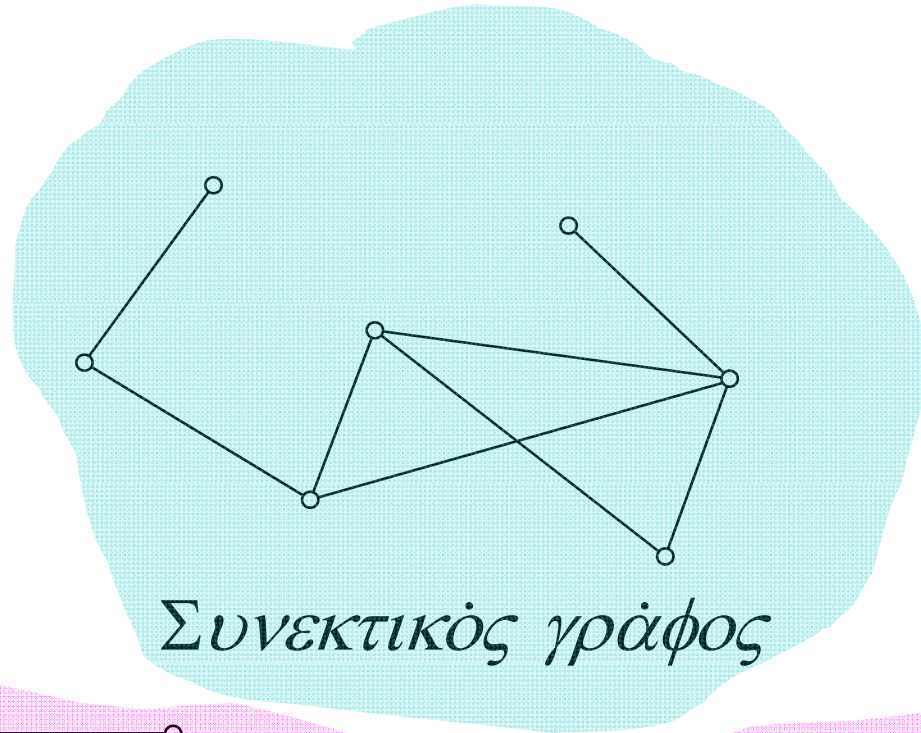
έχουμε 2 ή περισσότερες συνεκτικές συνιστώσες



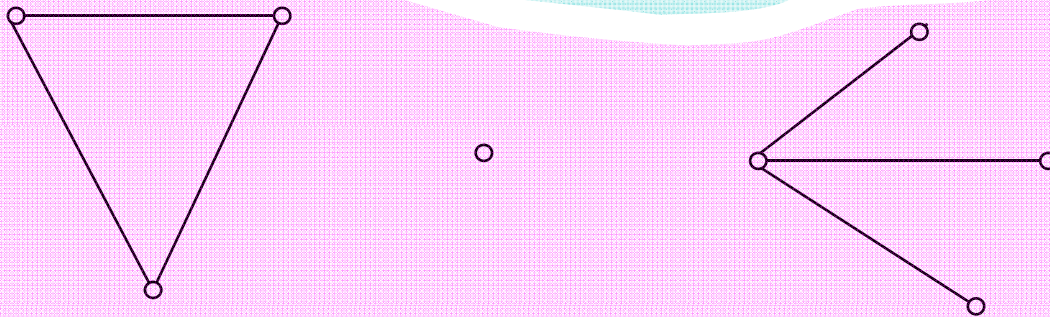
Μία συνεκτική συνιστώσα

Παραδείγματα ΣΥΝΕΚΤΙΚΟΤΗΤΑΣ

(****)



Συνεκτικός γράφος



Μη συνεκτικός γράφος με 3 συνεκτικές συνιστώσες

$G = (V, E) \rightarrow \text{Είναι Συνεκτικός?} \rightarrow (BFS)$

Procedure BFT(G)

MARK $[v_i] = \text{FALSE} \forall i$

For $i := 1$ to n do

if MARK $[v_i] = 0$ then BFS(v_i)

Breadth First graph Traversal (BFT)

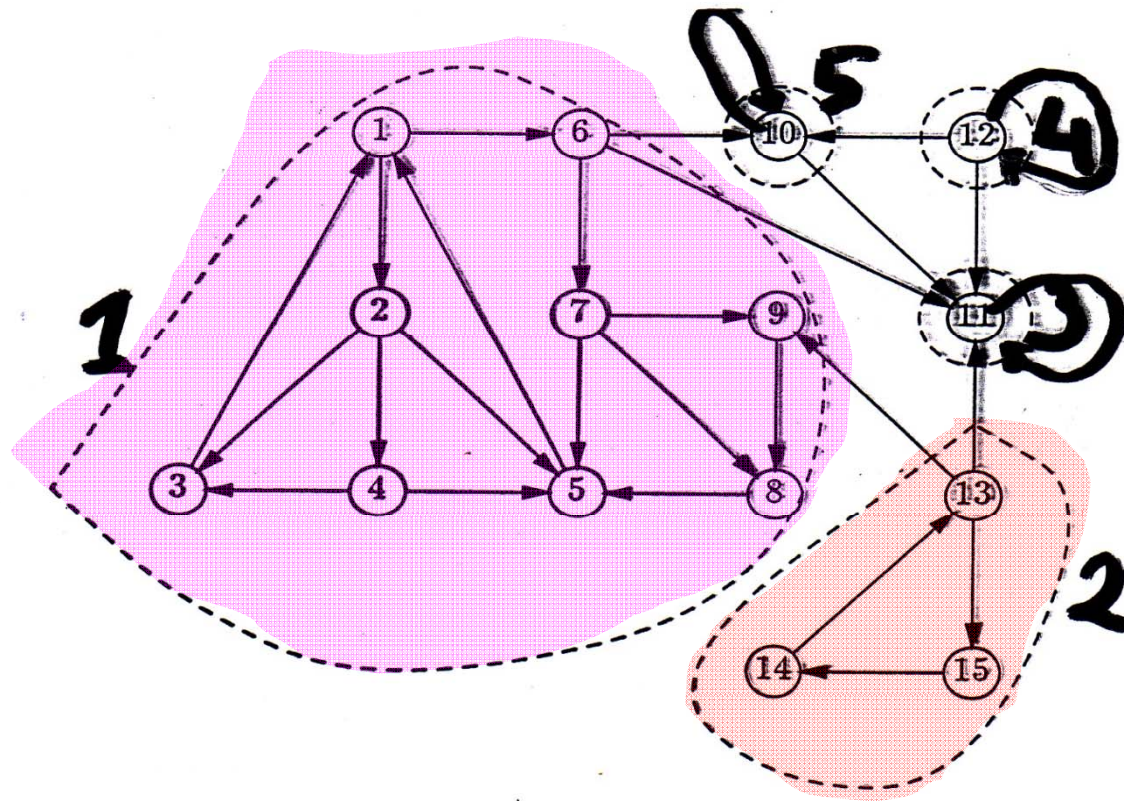
Πολ/τα = ? (****)

Ένας κατευθυνόμενος γράφος είναι **ισχυρά συνεκτικός** αν για κάθε ζευγάρι $\{x,y\}$ διακεκριμένων κόμβων υπάρχει ένα μονοπάτι από το x στο y και το αντίθετο από τον y στον x .

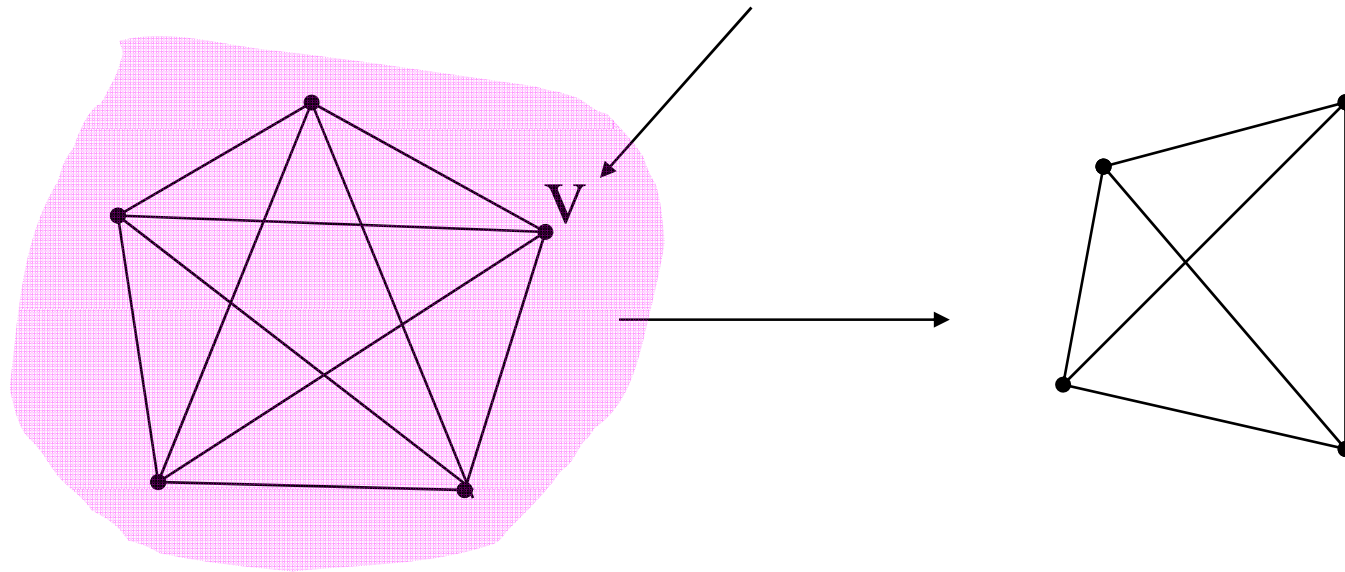
Δηλαδή, τα x και y βρίσκονται πάνω σε κλειστό μονοπάτι.

Διαφορετικά: δύο ή περισσότερες ισχυρά συνεκτικές συνιστώσες

ΙΣΧΥΡΗ ΣΥΝΕΚΤΙΚΟΤΗΤΑ

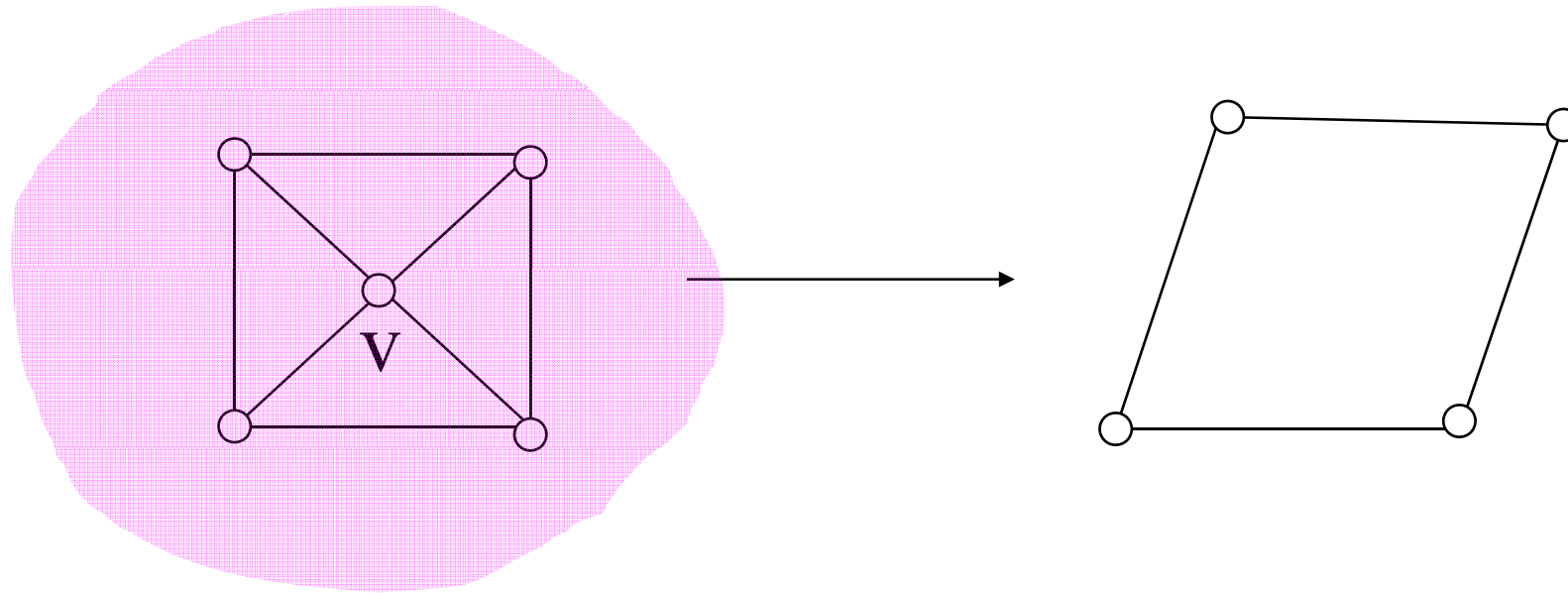


Συνεκτικότητα και διαγραφή ενός κόμβου



διαγραφή του κόμβου v

Συνεκτικότητα και διαγραφή ενός κόμβου



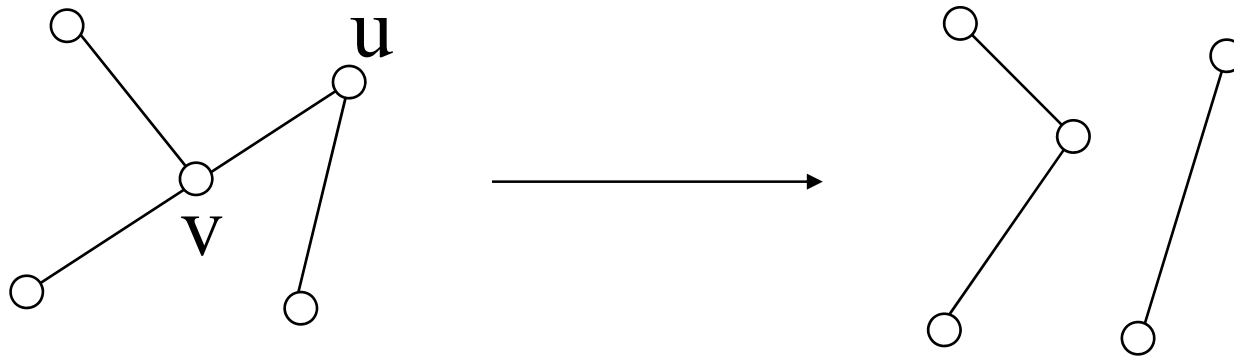
διαγραφή του κόμβου v

Συνεκτικότητα και διαγραφή ενός κόμβου/πλευράς



διαγραφή του κόμβου v

Συνεκτικότητα και διαγραφή ενός κόμβου



διαγραφή της πλευράς $[u,v]$