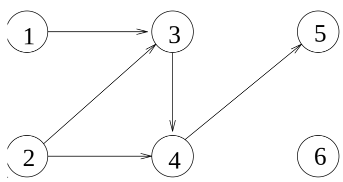
Γραφήματα(Graphs)

Ορισμός Ένα *γράφημα* G είναι ένα ζευγάρι (V, E) όπου V είναι το σύνολο των ***κόμβων**(vertices)* και E είναι το σύνολο των ***ακμών**(edges)*.

Παράδειγμα



V={1,2,3,4,5,6}

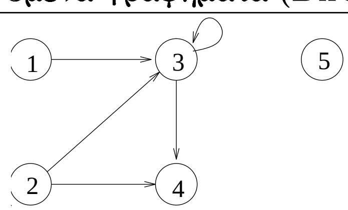
E={(1,3), (2,3), (2,4), (3,4), (4,5)}

Κάθε ακμή είναι ένα διατεταγμένο ζευγάρι από κόμβους.

Γιατί μελετάμε τα γραφήματα;

- Χρήσιμα στη μοντελοποίηση ενός τεραστίου αριθμου προβλημάτων.
- Σημαντικά εργσλεία στο σχεδιασμό αλγορίθμων.

Κατευθυνόμενα γραφήματα (Directed graphs – digraphs)

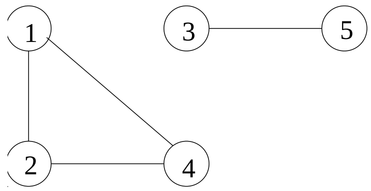


V={1,2,3,4,5}

E={(1,3), (2,3), (2,4), (3,3), (3,4)}

- Ο κόμβος 5 είναι ένας *απομονωμένος (isolated)* κόμβος.
- Η ακμή $(3, 3)$ είναι ένας *βρόγχος (self-loop)*.
- Ο *έσω-βαθμός (in-degree)* του κόμβου v είναι ο αριθμός των ακμών που εισέρχονται στον κόμβο v (συμβολίζεται με $d^-(v)$).
$$d^-(1) = d^-(2) = d^-(5) = 0, \quad d^-(3) = 3, \quad d^-(4) = 2$$
- Ο *έξω-βαθμός (out-degree)* του κόμβου v είναι ο αριθμός των ακμών που εξέρχονται από τον κόμβο v (συμβολίζεται με $d^+(v)$).
$$d^+(1) = 1, \quad d^+(2) = d^+(3) = 2, \quad d^+(4) = d^+(5) = 0$$
- $\sum_{v \in V} d^-(v) = \sum_{v \in V} d^+(v) = |E|$

Μη-κατευθυνόμενα γραφήματα (Undirected graphs)



V={1,2,3,4,5}

E={(1,2), (1,4), (2,4), (3,5)}

Κάθε ακμή $e \in E$ είναι ένα **μη-διατεταγμένο** ζευγάρι κόμβων.

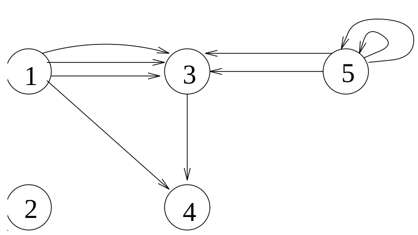
- Ο *βαθμός (degree)* ενός κόμβου v είναι ο αριθμός των προσκείμενων (incident) ακμών στον κόμβο v (συμβολίζεται με $d(v)$).

$$d(1) = d(2) = d(4) = 2, \quad d(3) = d(5) = 1$$

- $\displaystyle\sum_{v \in V} d(v) = 2 \cdot |E|$

;;(Multigraphs) (κατευθυνόμενα και μη-κατευθυνόμενα)

- Το E επιτρέπεται να περιέχει το ζεύγος κορυφών (u, v) περισσότερες απο μια φορές.

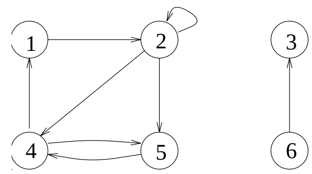


V={1,2,3,4,5}

E={(1,3), (1,3), (1,3), (1,4),
(3,4),
(5,3), (5,3), (5,5), (5,5)}

Μονοπάτια (paths)

- Ένα *μονοπάτι μήκους* k από ένα κόμβο u προς ένα κόμβο u' του γραφήματος $G = (V, E)$ είναι μια ακολουθία $< v_0,\ v_1,\ v_2, \cdots,\ v_k >$ από κόμβους έτσι ώστε $u = v_0,\ u' = v_k$ και $(v_{i-1}, v_i) \in E$ για $i = 1, 2, \cdots, k$.
- Το *μήκος (length)* του μονοπατιού είναι ο αριθμός των ακμών του.
- Ένα μονοπάτι είναι *απλό (simple)* εάν κανένας κόμβος δεν εμφανίζεται περισσότερες από μια φορά.



$< 2, 4, 5, 4, 1 >$ είναι ένα μονοπάτι μήκους 4 (μη-απλό).
$< 2, 4, 1 >$ είναι ένα απλό μονοπάτι μήκους 2.

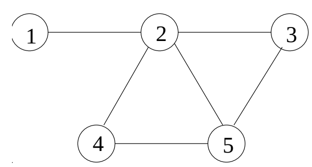
- Το μονοπάτι $< v_0,\ v_1, \cdots,\ v_k >$ σχηματίζει έναν ***κύκλο** (cycle)* εάν $v_0 = v_k$ και αποτελείται από τουλάχιστον μία ακμή.

$< 2, 4, 5, 4, 1, 2 >$ είναι ένας κύκλος.

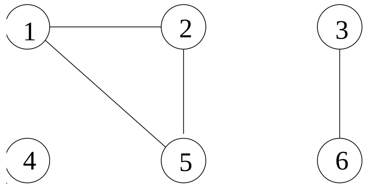
- Ένας κύκλος είναι *απλός (simple)* εάν κανένας κόμβος δεν εμφανίζεται περισσότερες από μια φορά.
- Ένα γράφημα χωρίς κύκλους ονομάζεται ***ακυκλικό** (acyclic)*.

Συνδεδεμένα συστατικά γραφήματα (Connected components)

- Ένα μη-κατευθυνόμενο γράφημα είναι *συνδεδεμένο* εάν κάθε ζευγάρι κόμβων ενώνεται με ένα μονοπάτι.



- Τα *συνδεδεμένα συστατικά* ενός μη-κατευθυνόμενου γραφήματος είναι οι κλάσεις ισοδυναμίας των κόμβων που ορίζονται από τη σχέση *'είναι προσπελάσιμος από'*.

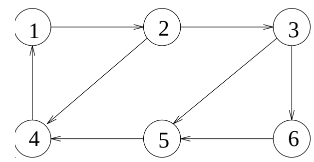


Connected components:

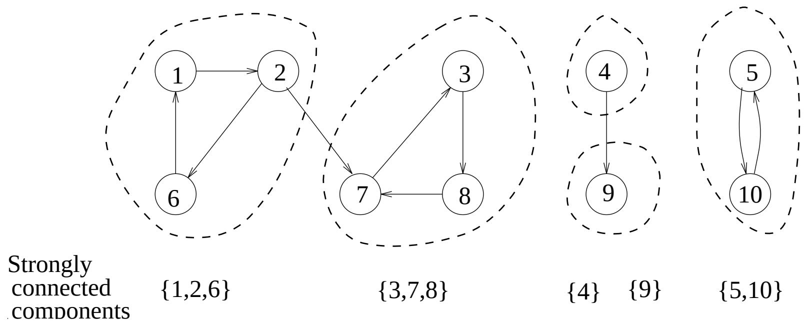
{1,2,5}
{3,6}
{4}

Ισχυρά συνδεδεμένα συστατικά γραφήματα (Strongly connected components)

- Ένα κατευθυνόμενο γράφημα είναι *ισχυρά συνδεδεμένο* εάν κάθε 2 κόμβοι του είναι προσπελάσιμοι ο ένας από τον άλλο.

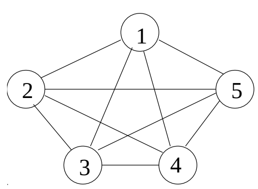


- Τα *ισχυρά συνδεδεμένα συστατικά* ενός κατευθυνόμενου γραφήματος είναι οι κλάσεις ισοδυναμίας των κόμβων που ορίζονται από τη σχέση *'είναι αμοιβαία προσπελάσιμοι'*.



Πλήρη γραφήματα (Complete graphs, cliques)

- Ένα μη-κατευθυνόμενο γράφημα στο οποίο κάθε ζευγάρι από κόμβους ενώνεται με μία ακμή.

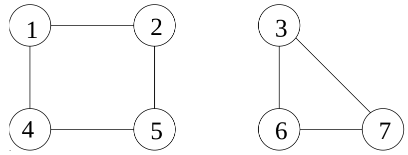


$$|E| = \frac{|V|\cdot(|V|-1)}{2}$$

$$d(v) = |V| - 1, \forall v \in V$$

Κανονικά γραφήματα βαθμού d (d-regular graphs)

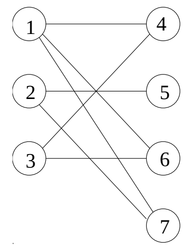
- Ένα μη κατευθυνόμενο γράφημα του οποίου όλοι οι κόμβοι είναι βαθμού d.



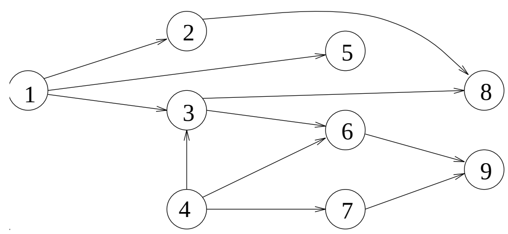
Ένα πλήρες γράφημα είναι ένα κανονικό γράφημα βαθμού $(|V| - 1)$.

Διμερή γραφήματα (Bipartite graphs)

- Ένα μη-κατευθυνόμενο γράφημα $G = (V, E)$ στο οποίο το V είναι διαμερισμένο σε δύο σύνολα, V_1 και V_2, έτσι ώστε:
  εάν $(u, v) \in E$ τότε είτε ($u \in V_1$ και $v \in V_2$) ή ($u \in V_2$ και $v \in V_1$).

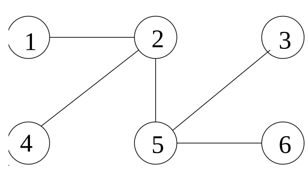


Κατευθυνόμενα ακυκλικά γραφήματα (Directed Acyclic Graphs, DAGs)



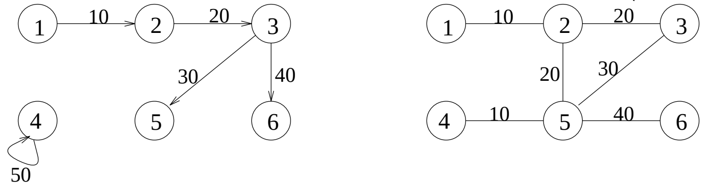
Δένδρα (Trees)

- Ένα μη-κατευθυνόμενο γράφημα που δεν περιέχει κύκλους.



Γραφήματα με βάρη (Weighted graphs)

- Ένα γράφημα του οποίου κάθε ακμή συνοδεύεται από ένα βάρος (ή ετικέτα).



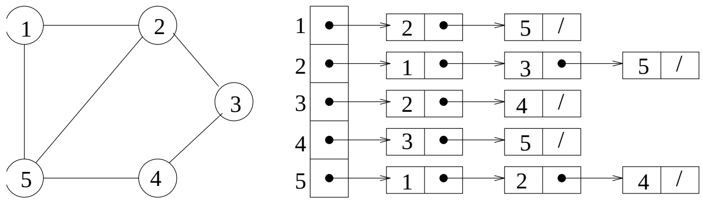
Directed Undirected

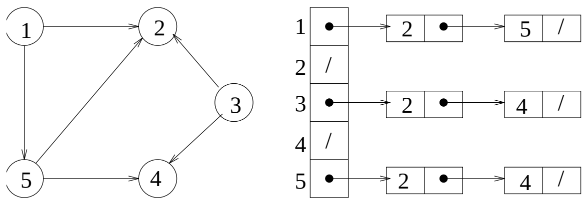
Συμβολισμός Όταν χρησιμοποιούμε τους ασυμπτωτικούς συμβολισμούς ($\Theta()$, $\Omega()$, $O()$) το σύμβολο V υποδηλώνει το $|V|$ και το E υποδηλώνει το $|E|$.

Αναπαράσταση των γραφημάτων

Λίστες γειτονικών κόμβων (Adjacency list representation)

- Ένα διάνυσμα Adj από $|V|$ λίστες, μία για κάθε κόμβο του V.
- Για κάθε $u \in V$, η λίστα γειτονικών κόμβων $Adj(u)$ περιέχει όλους τους κόμβους v για τους οποίους ισχύει ότι η ακμή $(u, v) \in E$.

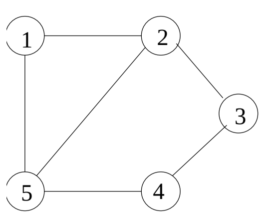




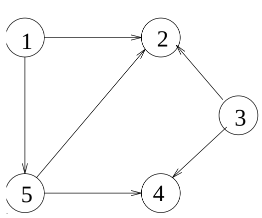
Απαιτούμενη μνήμη: $O(V + E)$.

Πίνακα γειτονικών κόμβων (Adjacency matrix representation)

- Ένας $|V| \times |V|$ πίνακας $A = (a_{ij})$ τέτοιος ώστε: $a_{ij} = \begin{cases} 1 & \text{εάν } (i,j) \in E \\ 0 & \text{σε άλλη περίπτωση} \end{cases}$



	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	0	1
3	0	1	0	1	0
4	0	0	1	0	1
5	1	1	0	1	0



	1	2	3	4	5
1	0	1	0	0	1
2	0	0	0	0	0
3	0	1	0	1	0
4	0	0	0	0	0
5	0	1	0	1	0

- Ο πίνακας γειτονικών κόμβων ενός μη-κατευθυνόμενου γραφήματος είναι συμμετρικός ως προς τη διαγώνιο του.

Απαιτούμενη μνήμη: $\Theta(V^2)$.

- Χρησιμοποιούμε λίστες γειτονικών κόμβων για *αραιά (sparse)* γραφήματα ($|E|$ είναι πολύ μικρότερο από το $|V|^2$).

- Χρησιμοποιούμε πίνακα γειτονικών κόμβων για *πυκνά (dense)* γραφήματα (το $|E|$ προσεγγίζει το $|V|^2$).

- Όταν χρησιμοποιούμε λίστες γειτονικών κόμβων, χρειαζόμαστε $O(|V|)$ για να διαπιστώσουμε εάν υπάρχει η ακμή (u, v).

- Όταν χρησιμοποιούμε πίνακα γειτονικών κόμβων, χρειαζόμαστε $O(1)$ για να διαπιστώσουμε εάν υπάρχει η ακμή (u, v).

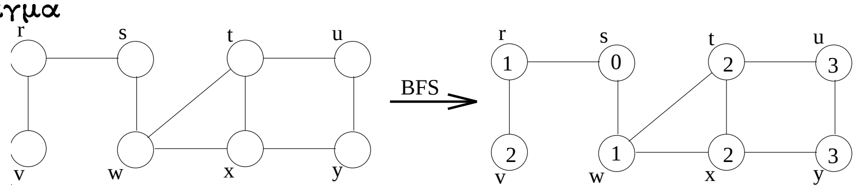
Αναζήτηση κατά πλάτος (Breadth-First Search - BFS)

Με δεδομένα ένα γράφημα $G = (V, E)$ και ένα κόμβο αφετηρία (source) s, θέλουμε να εξερευνήσουμε συστηματικά τις ακμές του G και να 'ανακαλύψουμε' κάθε κόμβο που είναι προσπελάσιμος από τον s.

- Η BFS υπολογίζει τις αποστάσεις (ελάχιστος αριθμός ακμών) προς όλους τους κόμβους που είναι προσπελάσιμοι από τον s.

- Η BFS δημιουργεί ένα δένδρο *(breadth-first tree)* που περιέχει όλους τους προσπελάσιμους από το s κόμβους.

Παράδειγμα

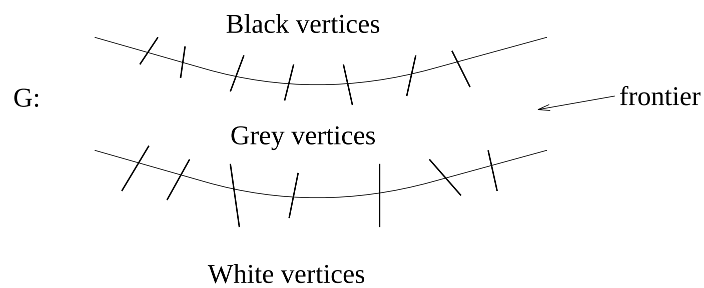


- Η BFS επεκτείνει τα όρια ανάμεσα στους κόμβους που έχουν 'ανακαλυφθεί' και σε αυτούς που δεν έχουν με ομοιόμορφο τρόπο (σε όλο το πλάτος των ορίων αυτών).

- Ανακαλύπτει όλους τους κόμβους που βρίσκονται σε απόσταση k από τον s πριν ανακαλύψει κάποιον κόμβο που βρίσκεται σε απόσταση $k+1$.

- Κατά την εκτέλεσή της, η BFS χρωματίζει τους κόμβους άσπρους, γκρι, ή μαύρους.

Άσπροι κόμβοι:	δεν έχουν ανακαλυφθεί ακόμη.
Γκρι κόμβοι:	έχουν ανακαλυφθεί· θα χρησιμοποιηθούν για την ανακάλυψη άλλων κόμβων.
Μαύροι κόμβοι:	έχουν ανακαλυφθεί· **δεν** θα χρησιμοποιηθούν για την ανακάλυψη άλλων κόμβων.



BFS(G, s)

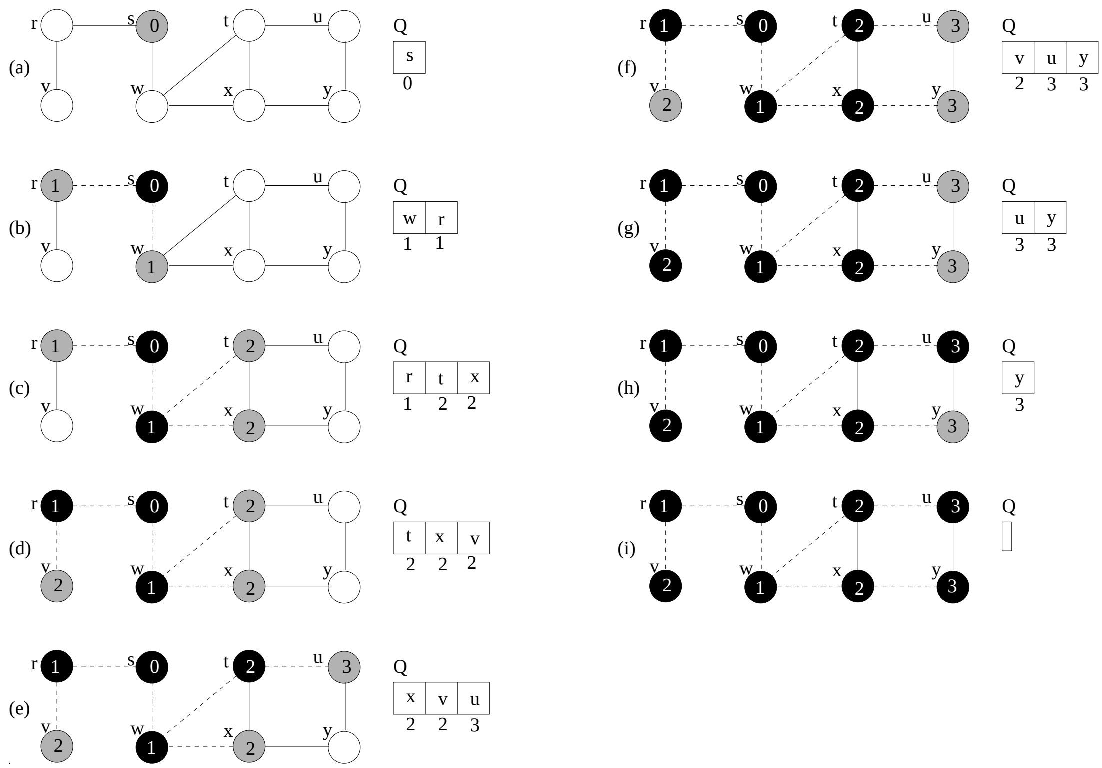
for each vertex $u \in V[G] - \{s\}$
    do $color[u] = white$
       $d[u] = +\infty$
       $\pi[u] = \text{NIL}$

$color[s] = grey$
$d[s] = 0$
$\pi[s] = \text{NIL}$

$Q \leftarrow \{s\}$ /* Q is a FIFO queue */
while $Q \neq \emptyset$
    do $u = head(Q)$
       for each vertex $v \in Adj[u]$
          do if $color[v] = white$
              then $color[v] = grey$
                  $d[v] = d[u] + 1$
                  $\pi[v] = u$
                  $enqueue(Q, v)$
          $dequeue(Q)$
          $color[u] = black$

Ανάλυση $O(V + E)$

(Υποθέτουμε ότι έχει χρησιμοποιηθεί η αναπαράσταση με λίστες γειτονικών κόμβων)



<u>Ιδιότητες της BFS</u>

- $d[u] = \delta(s,u)$
  ($\delta(s,u) \equiv$ Το μήκος του ελάχιστου μονοπατιού από τον s στον u, είναι ο ελάχιστος αριθμός ακμών από όλα τα μονοπάτια από τον s στον u.

- Η BFS παράγει ένα δένδρο (*breadth-first search tree*) το οποίο αποτελείται από όλους τους προσπελάσιμους από τον s κόμβους, έτσι ώστε, για κάθε κόμβο u του δένδρου ισχύει ότι το απλό μονοπάτι από τον s στον u είναι επίσης ένα ελάχιστο μονοπάτι.

<u>Με ποίο τρόπο ανακτούμε τα ελάχιστα μονοπάτια;</u>

Print_Path(G, s, v)

```
if v = s
   then print s
   else if π[v] = NIL
            then print " no path from s to v exists"
            else Print_Path(G, s, π[v])
                 print v
```

Ανάλυση: Χρόνος γραμμικός ως προς τον αριθμό των κόμβων του μονοπατιού.

- Η BFS μπορεί να εφαρμοστεί σε κατευθυνόμενα και μη-κατευθυνόμενα γραφήματα.