Αλγόριθμοι και Πολυπλοκότητα

Αρχοντία Γιαννοπούλου
Όλγα Φουρτουνέλλη

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Αλγόριθμοι σε Γραφήματα ΙΙ

Διάσχιση Γραφημάτων και Εφαρμογές

Τί είναι ένα γράφημα;

Μη κατευθυνόμενο (ή απλό) γράφημα: $G = (V, E)$

- V = κορυφές (nodes) ή κόμβοι (nodes)

Τί είναι ένα γράφημα;

Μη κατευθυνόμενο (ή απλό) γράφημα: $G = (V, E)$

- V = κορυφές (nodes) ή κόμβοι (nodes)
- E = ακμές μεταξύ ζευγαριών κορυφών (edges)

Τί είναι ένα γράφημα;

Μη κατευθυνόμενο (ή απλό) γράφημα: $G = (V, E)$

- V = κορυφές (nodes) ή κόμβοι (nodes)
- E = ακμές μεταξύ ζευγαριών κορυφών (edges)
- Αναπαριστά διακριτή ανά ζεύγη σχέση μεταξύ αντικειμένων

Τί είναι ένα γράφημα;

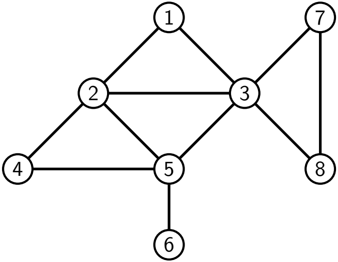
Μη κατευθυνόμενο (ή απλό) γράφημα: $G = (V, E)$

- V = κορυφές (nodes) ή κόμβοι (nodes)
- E = ακμές μεταξύ ζευγαριών κορυφών (edges)
- Αναπαριστά διακριτή ανά ζεύγη σχέση μεταξύ αντικειμένων
- Παράμετροι μεγέθους γραφήματος: $n(G) = |V|$ και $m(G) = |E|$

Τί είναι ένα γράφημα;

Μη κατευθυνόμενο (ή απλό) γράφημα: $G = (V, E)$

- V = κορυφές (nodes) ή κόμβοι (nodes)
- E = ακμές μεταξύ ζευγαριών κορυφών (edges)
- Αναπαριστά διακριτή ανά ζεύγη σχέση μεταξύ αντικειμένων
- Παράμετροι μεγέθους γραφήματος: $n(G) = |V|$ και $m(G) = |E|$
- Βαθμός κορυφής (degree) = πλήθος ακμών στις οποίες ανήκει



- ▷ $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$
- ▷ $E = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{4, 5\}, \{2, 5\}, \{3, 5\}, \{3, 7\}, \{7, 8\}, \{3, 8\}, \{5, 6\}\}$
- ▷ $n(G) = 8$
- ▷ $m(G) = 11$

Παραδείγματα Γραφημάτων

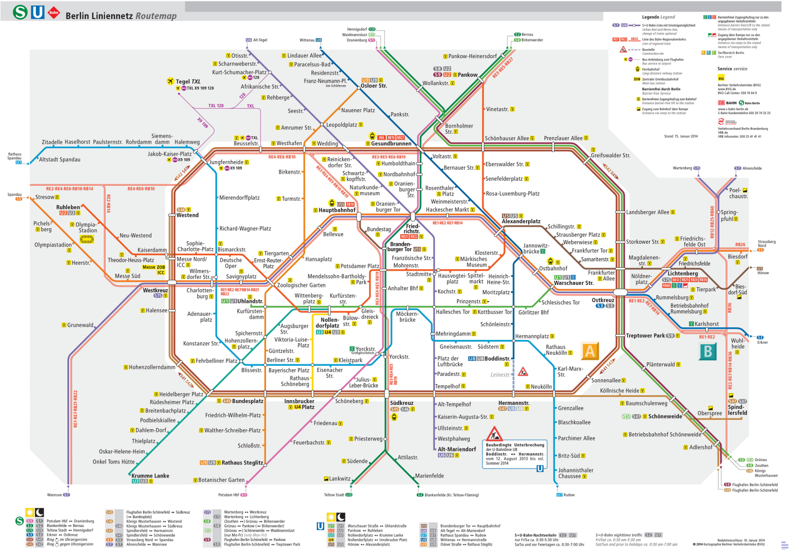
Γράφημα	Κορυφές	Ακμές
Δίκτυα μεταφορών	Διασταυρώσεις	Αυτοκινητόδρομοι
Δίκτυα επικοινωνιών	Υπολογιστές	Καλώδια οπτικών ινών
Παγκόσμιος Ιστός	Ιστοσελίδες	Υπερσύνδεσμοι
Κοινωνικά δίκτυα	Άνθρωποι	Σχέσεις
Τροφική αλυσίδα	Είδος οργανισμού	Θηρευτής-θήραμα
Συστήματα λογισμικού	Συναρτήσεις	Κλήσεις συναρτήσεων
Χρονοπρογραμματισμός	Εργασίες	Περιορισμοί προτεραιότητας
Κυκλώματα	Πύλες	Καλώδια

Δίκτυο Συγκοινωνιών

- Κορυφές: Σταθμοί
- Ακμές: Ανάμεσα σε δύο σταθμούς που απέχουν μία στάση

Δίκτυο Συγκοινωνιών

- Κορυφές: Σταθμοί
- Ακμές: Ανάμεσα σε δύο σταθμούς που απέχουν μία στάση



Οικολογική τροφική αλυσίδα

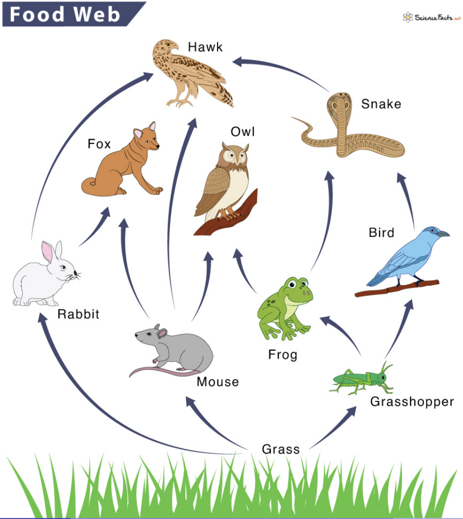
Γράφημα τροφικής αλυσίδας

- Κορυφές: είδη οργανισμών
- Ακμές: από θήραμα σε θηρευτή

Οικολογική τροφική αλυσίδα

Γράφημα τροφικής αλυσίδας

- Κορυφές: είδη οργανισμών
- Ακμές: από θήραμα σε θηρευτή



Αναπαράσταση Γραφημάτων: Πίνακας Γειτνίασης

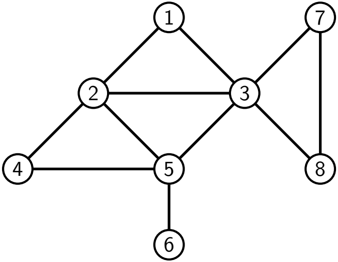
Πίνακας $n \times n$ στοιχείων όπου $a_{ij} = 1$ αν $\{i, j\} \in E$ και $a_{ij} = 0$ διαφορετικά.

- Δύο στοιχεία για κάθε ακμή
- Χώρος ανάλογος του n^2 (πυκνή αναπαράσταση)
- Ο έλεγχος αν υπάρχει η ακμή $\{i, j\}$ εκτελείται σε $\Theta(1)$ χρόνο
- Ο προσδιορισμός όλων των ακμών εκτελείται σε $\Theta(n^2)$ χρόνο

Αναπαράσταση Γραφημάτων: Πίνακας Γειτνίασης

Πίνακας $n \times n$ στοιχείων όπου $a_{ij} = 1$ αν $\{i, j\} \in E$ και $a_{ij} = 0$ διαφορετικά.

- Δύο στοιχεία για κάθε ακμή
- Χώρος ανάλογος του n^2 (πυκνή αναπαράσταση)
- Ο έλεγχος αν υπάρχει η ακμή $\{i, j\}$ εκτελείται σε $\Theta(1)$ χρόνο
- Ο προσδιορισμός όλων των ακμών εκτελείται σε $\Theta(n^2)$ χρόνο

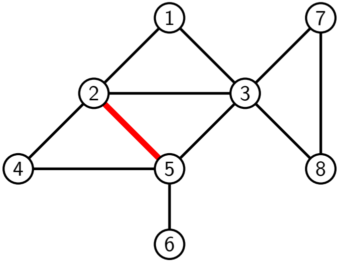


$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Αναπαράσταση Γραφημάτων: Πίνακας Γειτνίασης

Πίνακας $n \times n$ στοιχείων όπου $a_{ij} = 1$ αν $\{i, j\} \in E$ και $a_{ij} = 0$ διαφορετικά.

- Δύο στοιχεία για κάθε ακμή
- Χώρος ανάλογος του n^2 (πυκνή αναπαράσταση)
- Ο έλεγχος αν υπάρχει η ακμή $\{i, j\}$ εκτελείται σε $\Theta(1)$ χρόνο
- Ο προσδιορισμός όλων των ακμών εκτελείται σε $\Theta(n^2)$ χρόνο



$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Αναπαράσταση Γραφημάτων: Λίστα Γειτνίασης

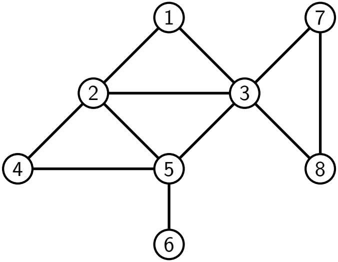
Λίστα γειτνίασης: Πίνακας λιστών δεικτοδοτημένος με κορυφές.

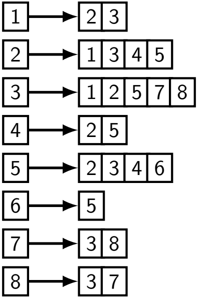
- Δύο αναπαραστάσεις για κάθε ακμή
- Χώρος ανάλογος του $m + n$ (αραιή αναπαράσταση)
- Ο έλεγχος αν το ζευγάρι i, j είναι ακμή εκτελείται σε χρόνο $\mathcal{O}(\deg(i))$, όπου $\deg(i) =$ βαθμός της κορυφής i
- Ο προσδιορισμός όλων των ακμών εκτελείται σε χρόνο $\Theta(m + n)$

Αναπαράσταση Γραφημάτων: Λίστα Γειτνίασης

Λίστα γειτνίασης: Πίνακας λιστών δεικτοδοτημένος με κορυφές.

- Δύο αναπαραστάσεις για κάθε ακμή
- Χώρος ανάλογος του $m + n$ (αραιή αναπαράσταση)
- Ο έλεγχος αν το ζευγάρι i, j είναι ακμή εκτελείται σε χρόνο $\mathcal{O}(\deg(i))$, όπου $\deg(i) =$ βαθμός της κορυφής i
- Ο προσδιορισμός όλων των ακμών εκτελείται σε χρόνο $\Theta(m + n)$

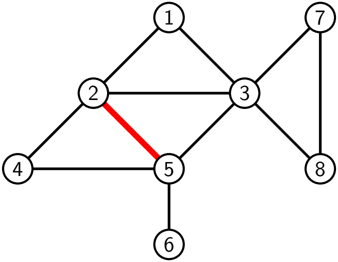




Αναπαράσταση Γραφημάτων: Λίστα Γειτνίασης

Λίστα γειτνίασης: Πίνακας λιστών δεικτοδοτημένος με κορυφές.

- Δύο αναπαραστάσεις για κάθε ακμή
- Χώρος ανάλογος του $m + n$ (αραιή αναπαράσταση)
- Ο έλεγχος αν το ζευγάρι i, j είναι ακμή εκτελείται σε χρόνο $\mathcal{O}(\deg(i))$, όπου $\deg(i) =$ βαθμός της κορυφής i
- Ο προσδιορισμός όλων των ακμών εκτελείται σε χρόνο $\Theta(m + n)$



1 → 2 3
2 → 1 3 4 5
3 → 1 2 5 7 8
4 → 2 5
5 → 2 3 4 6
6 → 5
7 → 3 8
8 → 3 7

Διαδρομές και συνεκτικότητα

- Μια διαδρομή σε ένα απλό γράφημα $G = (V, E)$ είναι μια ακολουθία P από κορυφές $v_1, v_2, \ldots, v_{k-1}, v_k$ με την ιδιότητα ότι κάθε διαδοχικό ζεύγος v_i, v_{i+1} συνδέεται με μια ακμή.

Διαδρομές και συνεκτικότητα

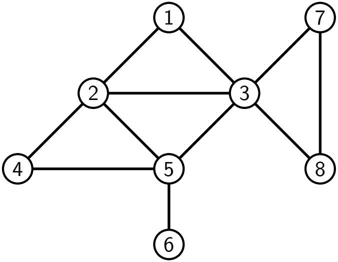
- Μια διαδρομή σε ένα απλό γράφημα $G = (V, E)$ είναι μια ακολουθία P από κορυφές v_1, v_2, ..., v_{k-1}, v_k με την ιδιότητα ότι κάθε διαδοχικό ζεύγος v_i, v_{i+1} συνδέεται με μια ακμή.
- Μια διαδρομή είναι απλή αν όλες οι κορυφές της είναι διακεκριμμένες (διαφορετικές).

Διαδρομές και συνεκτικότητα

- Μια διαδρομή σε ένα απλό γράφημα $G = (V, E)$ είναι μια ακολουθία P από κορυφές $v_1, v_2, \ldots, v_{k-1}, v_k$ με την ιδιότητα ότι κάθε διαδοχικό ζεύγος v_i, v_{i+1} συνδέεται με μια ακμή.
- Μια διαδρομή είναι απλή αν όλες οι κορυφές της είναι διακεκριμμένες (διαφορετικές).
- Ένα απλό γράφημα είναι συνεκτικό αν για κάθε ζευγάρι κορυφών u, v υπάρχει διαδρομή μεταξύ των u και v.

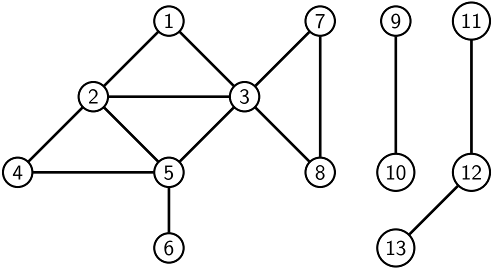
Διαδρομές και συνεκτικότητα

- Μια διαδρομή σε ένα απλό γράφημα $G = (V, E)$ είναι μια ακολουθία P από κορυφές $v_1, v_2, \ldots, v_{k-1}, v_k$ με την ιδιότητα ότι κάθε διαδοχικό ζεύγος v_i, v_{i+1} συνδέεται με μια ακμή.
- Μια διαδρομή είναι απλή αν όλες οι κορυφές της είναι διακεκριμμένες (διαφορετικές).
- Ένα απλό γράφημα είναι συνεκτικό αν για κάθε ζευγάρι κορυφών u, v υπάρχει διαδρομή μεταξύ των u και v.



Διαδρομές και συνεκτικότητα

- Μια διαδρομή σε ένα απλό γράφημα $G = (V, E)$ είναι μια ακολουθία P από κορυφές $v_1, v_2, \ldots, v_{k-1}, v_k$ με την ιδιότητα ότι κάθε διαδοχικό ζεύγος v_i, v_{i+1} συνδέεται με μια ακμή.
- Μια διαδρομή είναι απλή αν όλες οι κορυφές της είναι διακεκριμμένες (διαφορετικές).
- Ένα απλό γράφημα είναι συνεκτικό αν για κάθε ζευγάρι κορυφών u, v υπάρχει διαδρομή μεταξύ των u και v.



Κύκλοι

Ένας κύκλος είναι μια διαδρομή $v_1, v_2, \ldots, v_{k-1}, v_k$ στην οποία:

- $\{v_1, v_k\} \in E$,
- $k \geq 3$, και
- όλες οι k κορυφές είναι διακεκριμμένες (διαφορετικές) μεταξύ τους.

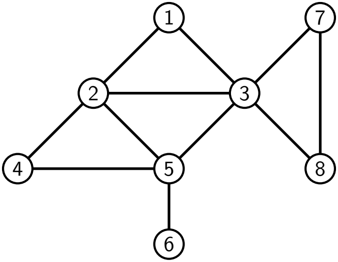
Με άλλα λόγια ένας κύκλος είναι μία απλή διαδρομή που "κλείνει".

Κύκλοι

Ένας κύκλος είναι μια διαδρομή $v_1, v_2, \ldots, v_{k-1}, v_k$ στην οποία:

- $\{v_1, v_k\} \in E$,
- $k \geq 3$, και
- όλες οι k κορυφές είναι διακεκριμμένες (διαφορετικές) μεταξύ τους.

Με άλλα λόγια ένας κύκλος είναι μία απλή διαδρομή που "κλείνει".

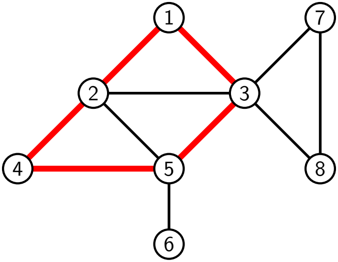


Κύκλοι

Ένας κύκλος είναι μια διαδρομή $v_1, v_2, \ldots, v_{k-1}, v_k$ στην οποία:

- $\{v_1, v_k\} \in E$,
- $k \geq 3$, και
- όλες οι k κορυφές είναι διακεκριμμένες (διαφορετικές) μεταξύ τους.

Με άλλα λόγια ένας κύκλος είναι μία απλή διαδρομή που "κλείνει".



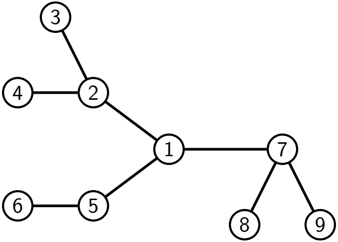
Ένας κύκλος C είναι η διαδρομή 1 - 2 - 4 - 5 -3 -1.

Δένδρα

Ένα απλό γράφημα είναι δένδρο αν είναι συνεκτικό και δεν περιέχει κύκλο.

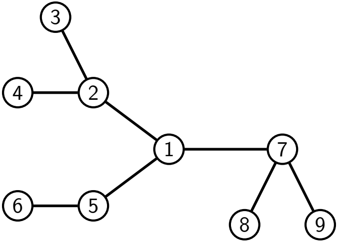
Δένδρα

Ένα απλό γράφημα είναι δένδρο αν είναι συνεκτικό και δεν περιέχει κύκλο.



Δένδρα

Ένα απλό γράφημα είναι δένδρο αν είναι συνεκτικό και δεν περιέχει κύκλο.



Θεώρημα. Έστω G ένα γράφημα με n κορυφές. Οποιεσδήποτε 2 από τις ακόλουθες προτάσεις συνεπάγονται την τρίτη:

1. Το G είναι συνεκτικό.
2. Το G δεν περιέχει κύκλο.
3. Το G έχει $n - 1$ ακμές.

Δένδρα με ρίζα

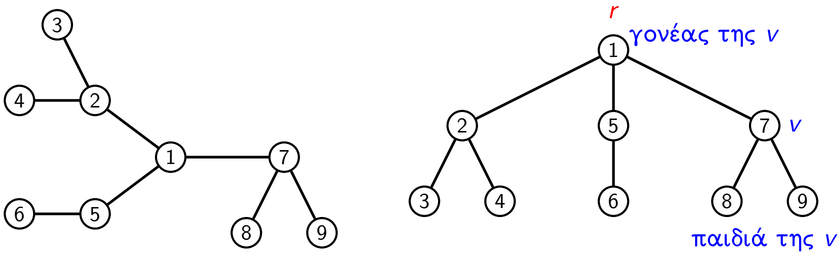
- Δένδρο με ρίζα. Δεδομένου ενός δένδρου T, διαλέξτε μία κορυφή r ως ρίζα (και προσανατολίστε κάθε ακμή έτσι ώστε να απομακρύνεται από την r).

- Σημασία. Μοντελοποιεί μια ιεραρχική δομή.

Δένδρα με ρίζα

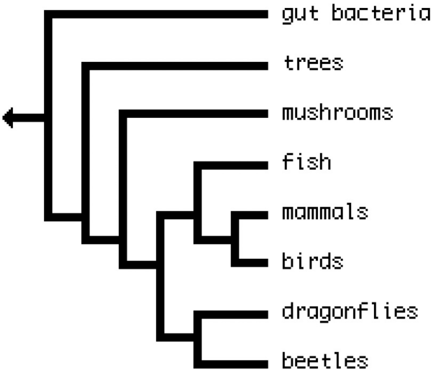
- Δένδρο με ρίζα. Δεδομένου ενός δένδρου T, διαλέξτε μία κορυφή r ως ρίζα (και προσανατολίστε κάθε ακμή έτσι ώστε να απομακρύνεται από την r).

- Σημασία. Μοντελοποιεί μια ιεραρχική δομή.



Φυλογενετικά δένδρα

Περιγραφή εξελικτικής ιστορίας των ειδών.



Συνεκτικότητα

- **Συνεκτικότητα** s-t. Δεδομένων κορυφών s και t, υπάρχει διαδρομή από την s στην t;
- **Πρόβλημα απόστασης** s-t. Δεδομένων κορυφών s και t, ποιο είναι το μήκος της **συντομότερης** διαδρομής από την s στην t;
- **Εφαρμογές**.
  - Κοινωνικά δίκτυα.
  - Επίλυση Λαβυρίνθου.
  - Six Degrees of Kevin Bacon (παιχνίδι γνώσεων).
  - Ελάχιστος αριθμός ενδιάμεσων κορυφών σε τηλεπικοινωνιακό δίκτυο.

Αναζήτηση κατά βάθος (Depth-First Search)

Βρείτε όλες τις κορυφές R που μπορούμε να επισκεπτούμε από την s.

Αναζήτηση κατά βάθος (Depth-First Search)

Βρείτε όλες τις κορυφές R που μπορούμε να επισκεπτούμε από την s.

DFS(G, s)
$R = \{s\}$
ενόσω υπάρχει ακμή $\{u, v\}$ όπου $u \in R$ και $v \notin R$
    $R \leftarrow R \cup \{v\}$

Αναζήτηση κατά βάθος (Depth-First Search)

Βρείτε όλες τις κορυφές R που μπορούμε να επισκεπτούμε από την s.

DFS(G, s)
$R = \{s\}$
ενόσω υπάρχει ακμή $\{u, v\}$ όπου $u \in R$ και $v \notin R$
$\quad R \leftarrow R \cup \{v\}$

Θεώρημα. Με τον τερματισμό, το R είναι η συνεκτική συνιστώσα που περιέχει την s.

Αναζήτηση κατά βάθος (Depth-First Search)

Βρείτε όλες τις κορυφές R που μπορούμε να επισκεπτούμε από την s.

DFS(G, s)
$R = \{s\}$
ενόσω υπάρχει ακμή $\{u, v\}$ όπου $u \in R$ και $v \notin R$
  $R \leftarrow R \cup \{v\}$

Θεώρημα. Με τον τερματισμό, το R είναι η συνεκτική συνιστώσα που περιέχει την s.

- DFS = αναζήτησε πρώτα κατά βάθος.
- BFS = εξερεύνησε με σειρά απόστασης από την s.

Αναζήτηση κατά βάθος (Depth-First Search)

DFS(G, s)
$R = \emptyset$
Για κάθε $u \in V$
    Εξερευνημένη(u) = false
Εξερευνημένη(s) = true
$R = \{s\}$
Για κάθε ακμή $\{s, u\}$ στο G
    Εάν Εξερευνημένη(u) = false
        DFS(G, u)

Αναζήτηση κατά πλάτος (Breadth-First Search)

BFS διαισθητικά. Εξερεύνηση από την s προς όλες τις πιθανές κατευθύνσεις, προσθέτοντας κορυφές ένα ένα ``επίπεδο'' τη φορά.

Αναζήτηση κατά πλάτος (Breadth-First Search)

BFS διαισθητικά. Εξερεύνηση από την s προς όλες τις πιθανές κατευθύνσεις, προσθέτοντας κορυφές άνα ένα ‘‘επίπεδο’’ τη φορά.

Ο αλγόριθμος BFS.

- $L_0 = \{s\}$.
- L_1 = όλοι οι γείτονες του L_0.
- L_2 = όλες οι κορυφές που δεν ανήκουν στο L_0 ή στο L_1, και έχουν μια ακμή προς μία κορυφή στο L_1.
- L_{i+1} = όλες οι κορυφές που δεν ανήκουν σε προηγούμενο επίπεδο, και έχουν μια ακμή προς μία κορυφή στο L_i.

Αναζήτηση κατά πλάτος (Breadth-First Search)

BFS διαισθητικά. Εξερεύνηση από την s προς όλες τις πιθανές κατευθύνσεις, προσθέτοντας κορυφές ένα ένα ''επίπεδο'' τη φορά.

Ο αλγόριθμος BFS.

- $L_0 = \{s\}$.
- $L_1 =$ όλοι οι γείτονες του L_0.
- $L_2 =$ όλες οι κορυφές που δεν ανήκουν στο L_0 ή στο L_1, και έχουν μια ακμή προς μία κορυφή στο L_1.
- $L_{i+1} =$ όλες οι κορυφές που δεν ανήκουν σε προηγούμενο επίπεδο, και έχουν μια ακμή προς μία κορυφή στο L_i.

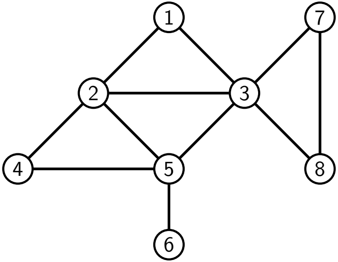
Θεώρημα. Για κάθε i, το L_i αποτελείται από όλες τις κορυφές σε απόσταση ακριβώς i από την s. Υπάρχει διαδρομή από την s στην t αν και μόνο αν η t εμφανίζεται σε κάποιο επίπεδο.

Αναζήτηση κατά πλάτος (Breadth-First Search)

Ιδιότητα. Έστω T ένα δένδρο με ρίζα s που παράγεται από τον BFS στο $G = (V, E)$ και έστω $\{x, y\}$ ακμή του G. Τότε τα επίπεδα των x και y διαφέρουν το πολύ κατά 1.

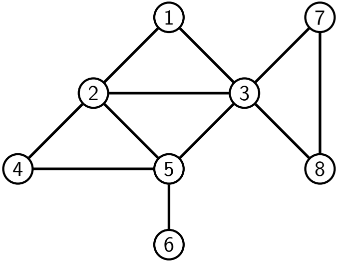
Αναζήτηση κατά πλάτος (Breadth-First Search)

Ιδιότητα. Έστω T ένα δένδρο με ρίζα s που παράγεται από τον BFS στο $G = (V, E)$ και έστω $\{x, y\}$ ακμή του G. Τότε τα επίπεδα των x και y διαφέρουν το πολύ κατά 1.



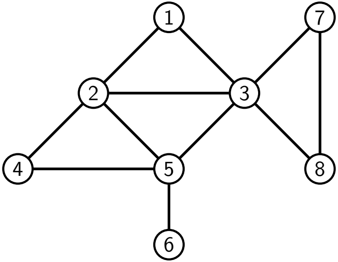
Αναζήτηση κατά πλάτος (Breadth-First Search)

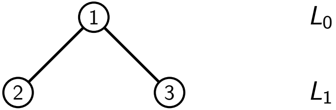
Ιδιότητα. Έστω T ένα δένδρο με ρίζα s που παράγεται από τον BFS στο $G = (V, E)$ και έστω $\{x, y\}$ ακμή του G. Τότε τα επίπεδα των x και y διαφέρουν το πολύ κατά 1.



Αναζήτηση κατά πλάτος (Breadth-First Search)

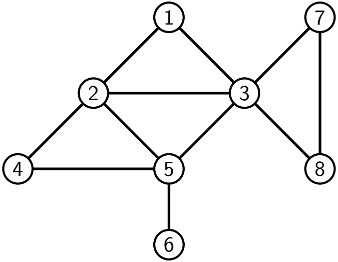
Ιδιότητα. Έστω T ένα δένδρο με ρίζα s που παράγεται από τον BFS στο $G = (V, E)$ και έστω $\{x, y\}$ ακμή του G. Τότε τα επίπεδα των x και y διαφέρουν το πολύ κατά 1.

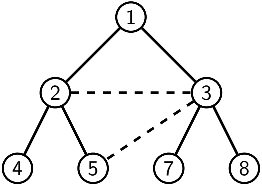




Αναζήτηση κατά πλάτος (Breadth-First Search)

Ιδιότητα. Έστω T ένα δένδρο με ρίζα s που παράγεται από τον BFS στο $G = (V, E)$ και έστω $\{x, y\}$ ακμή του G. Τότε τα επίπεδα των x και y διαφέρουν το πολύ κατά 1.





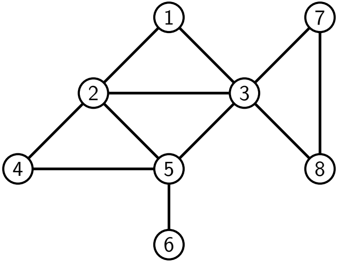
L_0

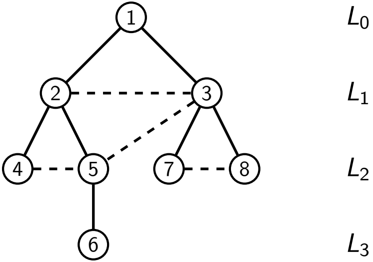
L_1

L_2

Αναζήτηση κατά πλάτος (Breadth-First Search)

Ιδιότητα. Έστω T ένα δένδρο με ρίζα s που παράγεται από τον BFS στο $G = (V, E)$ και έστω $\{x, y\}$ ακμή του G. Τότε τα επίπεδα των x και y διαφέρουν το πολύ κατά 1.





Ο Αλγόριθμος

BFS(s)
Για κάθε $v \in V$
  Αναγνωσμένη(v) $=$ false
  $\pi[v] \leftarrow \emptyset$
Αναγνωσμένη(s) $=$ true
$L[0] = \{s\}$
$i \leftarrow 0$ // δένδρο $T = \{s\}$.
ενόσω $L[i] \neq \emptyset$
  αρχικοποίηση κενής λίστας $L[i+1]$ // νέο επίπεδο
  για κάθε $u \in L[i]$
    για κάθε $\{u, v\} \in E$
      εάν Αναγνωσμένη(v) $=$ false **τότε**
        Αναγνωσμένη(v) $=$ true
        $\pi[v] \leftarrow u$ // $T = T \cup \{u, v\}$
        $L[i+1] \leftarrow L[i+1] \cup \{v\}$ // νέα κορυφή
  $i \leftarrow i + 1$

Ανάλυση Υλοποίησης

Η προηγούμενη υλοποίηση του αλγορίθμου BFS τρέχει σε χρόνο $\mathcal{O}(m+n)$ αν το γράφημα δίνεται με την αναπαράσταση λιστών γειτνίασης.

Ανάλυση Υλοποίησης

Η προηγούμενη υλοποίηση του αλγορίθμου BFS τρέχει σε χρόνο $\mathcal{O}(m+n)$ αν το γράφημα δίνεται με την αναπαράσταση λιστών γειτνίασης.

Απόδειξη Το πολύ n λίστες $L[i]$ στο (ενόσω).

- Εύκολο να αποδειχτεί χρόνος εκτέλεσης $\mathcal{O}(n^2)$:
  - Κάθε κορυφή βρίσκεται το πολύ σε μια λίστα $L[i]$,
  - ο βρόχος ''για κάθε u'' εκτελείται το πολύ n φορές σε όλη την διάρκεια του αλγόριθμου.
  - Όταν εξετάζουμε μία κορυφή u, υπάρχουν το πολύ n προσκείμενες ακμές $\{u, v\}$, και ξοδεύουμε $\mathcal{O}(1)$ χρόνο για την εξέταση κάθε ακμής.

  Σύνολο: $\sum_{v \in V}(1+n) = \mathcal{O}(n^2)$.

Ανάλυση Υλοποίησης

Η προηγούμενη υλοποίηση του αλγορίθμου BFS τρέχει σε χρόνο $\mathcal{O}(m+n)$ αν το γράφημα δίνεται με την αναπαράσταση λιστών γειτνίασης.

Απόδειξη Το πολύ n λίστες $L[i]$ στο (ενόσω).

- Εύκολο να αποδειχτεί χρόνος εκτέλεσης $\mathcal{O}(n^2)$:
  - Κάθε κορυφή βρίσκεται το πολύ σε μια λίστα $L[i]$,
  - ο βρόχος ‘‘για κάθε u’’ εκτελείται το πολύ n φορές σε όλη την διάρκεια του αλγόριθμου.
  - Όταν εξετάζουμε μία κορυφή u, υπάρχουν το πολύ n προσκείμενες ακμές $\{u, v\}$, και ξοδεύουμε $\mathcal{O}(1)$ χρόνο για την εξέταση κάθε ακμής.

  Σύνολο: $\sum_{v \in V}(1+n) = \mathcal{O}(n^2)$.

- Στην πραγματικότητα μας αρκεί $\mathcal{O}(m+n)$ χρόνος.
  - Όταν έξετάζουμε μία κορυφή u αυτή έχει το πολύ $\deg(u)$ προσκείμενες ακμές.
  - Άρα σύνολο: $\sum_{v \in V}(1+\deg(u)) = n + \sum_{v \in V}\deg(u) = n + 2m$.