Γράφοι: Προβλήματα και Αλγόριθμοι

Μάθημα: Εισαγωγή στην Επιστήμη των Υπολογιστών

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Σχολή Εφαρμοσμένων Μαθηματικών & Φυσικών Επιστημών
Εθνικό Μετσόβιο Πολυτεχνείο

Γράφοι (ή Γραφήματα)

Ορισμός. *Γράφος (ή γράφημα) G, ονομάζεται ένα διατεταγμένο ζεύγος συνόλων (V, E), όπου V είναι μη κενό σύνολο στοιχείων και E ένα σύνολο μη διατεταγμένων ζευγών του V, δηλαδή*

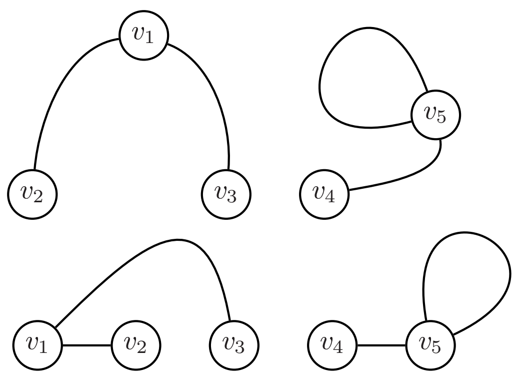
$$E \subseteq \binom{V}{2}$$

V: κορυφές (vertices) ή κόμβοι (nodes).

E: ακμές ή πλευρές (edges).

Παράδειγμα Γράφου

$$E = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_4, v_5\}, \{v_5, v_5\}\}$$



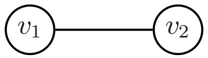
Ορολογία - Συμβολισμοί

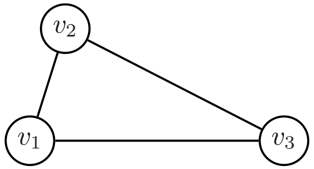
- Οι κορυφές συμβολίζονται με ένα γράμμα, συνήθως v, u, w, x, y, ή έναν αριθμό.
- Οι ακμές συμβολίζονται με ένα γράμμα, συνήθως e, αλλά και ως μη διατεταγμένο ζεύγος κορυφών $e = \{u, v\}$, μερικές φορές και απλά ως uv ή vu.
- Γειτονικές (adjacent) κορυφές, άκρα (endpoints) ακμής, προσπίπτουσα (incident) ακμή, γειτονικές ακμές.

Βαθμός κορυφής

Βαθμός (*degree, valence*) κορυφής v: ο αριθμός των ακμών που προσπίπτουν στην v. Συμβολίζεται με $d(v)$.

Ένας γράφος για τον οποίο ισχύει $d(v) = k$ για κάθε κορυφή του, λέγεται k-κανονικός γράφος.

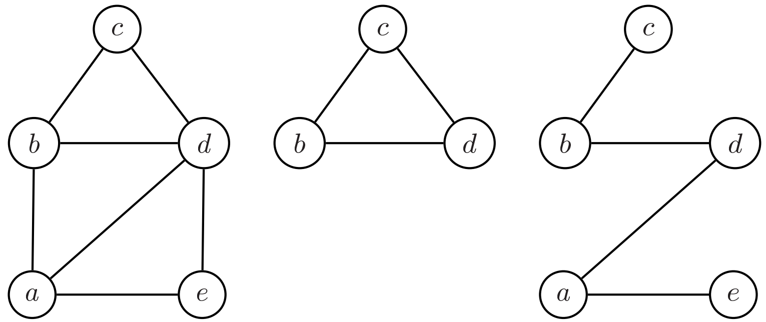
G_1: 

G_2: 

G_1: 1–κανονικός και G_2: 2–κανονικός γράφος

Σημαντική ιδιότητα: $\sum_{v\in V} d(v) = 2|E|$

Υπογράφοι

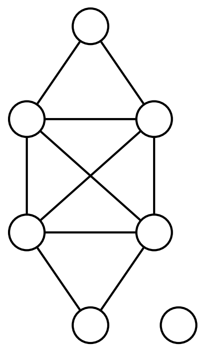


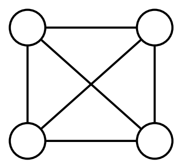
Ένας γράφος $G' = (V', E')$ είναι υπογράφος (*subgraph*) ενός άλλου γράφου $G = (V, E)$, αν ισχύει $V' \subseteq V$ και $E' \subseteq E$.

Δρόμοι, μονοπάτια, κύκλοι

- Δρόμος (walk): έγκυρη ακολουθία κορυφών-ακμών.
- Μονοπάτι (path): δρόμος χωρίς επαναλήψεις ακμών.
- Απλό μονοπάτι (simple path): μονοπάτι χωρίς επαναλήψεις κορυφών.
- Κύκλος (cycle): κλειστό μονοπάτι. Απλός κύκλος: κλειστό απλό μονοπάτι.
- Μήκος δρόμου: το πλήθος των ακμών του.

Γράφοι Euler, Hamilton





Γράφος Euler Γράφος Hamilton

Παράσταση Γράφου

- Πίνακας γειτνίασης (adjacency matrix)
- Πίνακας πρόσπτωσης (incidence matrix)
- Λίστες γειτνίασης (adjacency lists): αποδοτική παράσταση σε αραιούς γράφους.

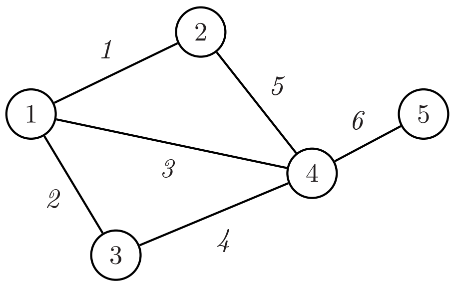
Παράσταση με πίνακα γειτνίασης

Έστω γράφος $G = (V, E)$ με $V = \{v_1, v_2, \cdots, v_n\}$. Τότε ο γράφος μπορεί να παρασταθεί με τη βοήθεια ενός $n \times n$ πίνακα $A(G)$, όπου

$$A(G) = [a_{ij}], \quad a_{ij} = \begin{cases} 1, & \text{αν } \{v_i, v_j\} \in E \\ 0, & \text{αλλιώς} \end{cases}$$

Ο πίνακας $A(G)$ λέγεται πίνακας γειτνίασης (*adjacency matrix*), και είναι συμμετρικός ($a_{i,j} = a_{j,i}$).

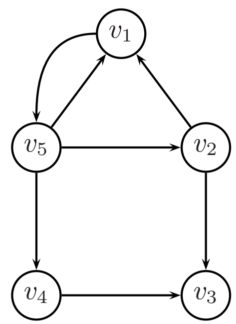
Παράσταση με λίστες γειτνίασης



[1]→ 2 3 4
[2]→ 1 4
[3]→ 1 4
[4]→ 1 2 3 5
[5]→ 4

Κατευθόμενος γράφος (directed graph)

$$E \subseteq V \times V$$



Συνεκτικότητα (connectivity)

- Ένας μη κατευθυνόμενος γράφος λέγεται συνεκτικός *(connected)* αν υπάρχει δρόμος μεταξύ οποιωνδήποτε δύο κορυφών του.
- Ένας κατευθυνόμενος γράφος λέγεται:
  - ισχυρά συνεκτικός *(strongly connected)* αν υπάρχει δρόμος από κάθε κορυφή σε κάθε κορυφή ακολουθώντας τις κατευθύνσεις των ακμών.
  - ασθενώς συνεκτικός *(weakly connected)* αν αν υπάρχει δρόμος μεταξύ οποιωνδήποτε δύο κορυφών του αγνοώντας τις κατευθύνσεις.
- Σε συνεκτικό γράφο ισχύει: $n-1 \leq e \leq \frac{n(n-1)}{2}$, $n = |V|, e = |E|$.
  - Αραιοί (sparse) γράφοι: $e = O(n)$.
  - Πυκνοί (dense) γράφοι: $e = \Omega(n^2)$.
  - Για αραιούς γράφους συμφέρει η αναπαράσταση με λίστες γειτνίασης.

Αλλες έννοιες

- Παράγων υπογράφος (spanning subgraph), παραγόμενος υπογράφος (induced subgraph).
- Συνεκτικές συνιστώσες (connected components).
- Πλήρης γράφος (K_n), διμερής γράφος (πλήρης διμερής: $K_{n,m}$).
- Επίπεδος γράφος: ανν δεν περιέχει ως υπογράφους τα K_5, $K_{3,3}$ ούτε γράφους που προκύπτουν από αυτά με υποδιαιρέσεις των ακμών τους.
- Δένδρα (trees).

Βασικές Κλάσεις Πολυπλοκότητας

P: προβλήματα απόφασης επιλύσιμα σε πολυωνυμικό χρόνο από κάποιον ντετερμινιστικό αλγόριθμο.

NP: προβλήματα απόφασης επιλύσιμα σε πολυωνυμικό χρόνο από κάποιον μη ντετερμινιστικό αλγόριθμο. Πιθανές λύσεις (πιστοποιητικά, αποδείξεις, μάρτυρες) ελέγξιμες σε πολυωνυμικό χρόνο.

Το μεγάλο ανοιχτό ερώτημα: $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$

NP-completeness, αναγωγές.

NP-πλήρη προβλήματα γράφων

VERTEX COVER

CLIQUE

HAMILTON CIRCUIT (HC)

TRAVELING SALESMAN PROBLEM (TSP)

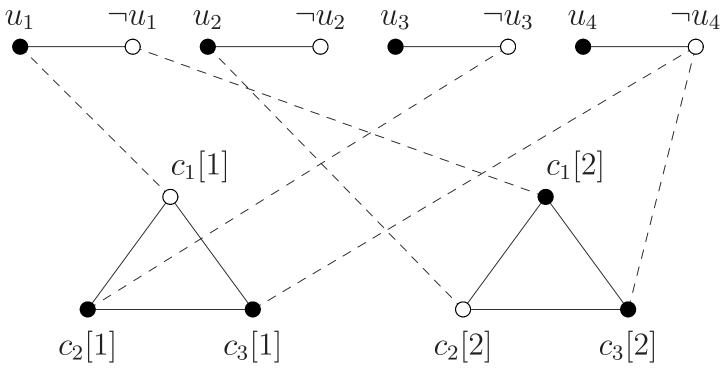
3-COLORABILITY

SUBGRAPH ISOMORPHISM

3-DIMENSIONAL MATCHING (3DM)

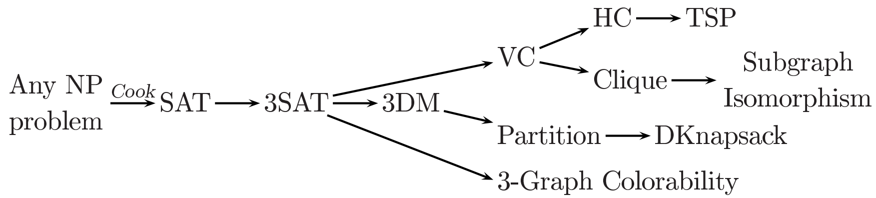
Αναγωγή 3SAT $\leq$ VERTEX COVER

Φ: $(u_1 \vee \neg u_3 \vee \neg u_4) \wedge (\neg u_1 \vee u_2 \vee \neg u_4)$



Η Φ είναι ικανοποιήσιμη ανν υπάρχει vertex cover μεγέθους $\leq k = n + 2m = 8$ στον γράφο που κατασκευάσαμε.

Αλλες Αναγωγές



Προβλήματα γράφων στην κλάση P

Κύκλος Euler.

Reachability - Διάσχιση Γράφων: DFS, BFS, D-Search.

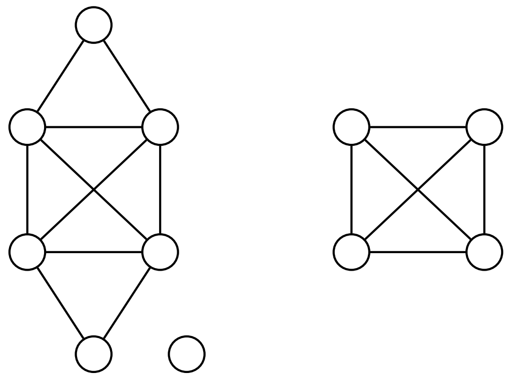
Συντομότερα μονοπάτια. Συνεκτικές συνιστώσες.

Ελάχιστο συνδετικό δένδρο (minimum spanning tree).

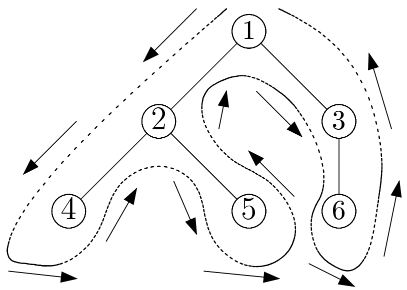
Μέγιστη ροή. Perfect matching.

Χρωματισμός ακμών σε διμερή γράφο (bipartite edge coloring).

Κύκλος Euler - Μονοπάτι Euler



Διάσχιση δένδρων



- προδιατεταγμένη: 1 2 4 5 3 6
- μεταδιατεταγμένη: 4 5 2 6 3 1
- ενδοδιατεταγμένη: 4 2 5 1 6 3

Accessibility problems - Διάσχιση γράφων

Αναζήτηση κατά βάθος (Depth First Search - DFS).

Αναζήτηση κατά πλάτος (Breadth First Search - BFS).

D-search: όμοιο με BFS, αλλά με στοίβα αντί για ουρά.

Διάσχιση γράφων: DFS

procedure dfs(v:vertex);
begin
  visited[v]:=true;
  for all vertices u adjacent to v do
   if not visited[u] then dfs(u)
end

Πολυπλοκότητα: $O(|V| + |E|)$.

Διάσχιση γράφων: BFS

procedure bfs(v:vertex);
begin
  initialize queue with v; visited[v]:=true;
  repeat
    dequeue(u);
    for all vertices w adjacent to u do
      if not visited[w] then
        begin visited[w] := true; enqueue(w) end
  until queue is empty
end

Πολυπλοκότητα: $O(|V| + |E|)$.

Συντομότερα μονοπάτια: Dijkstra

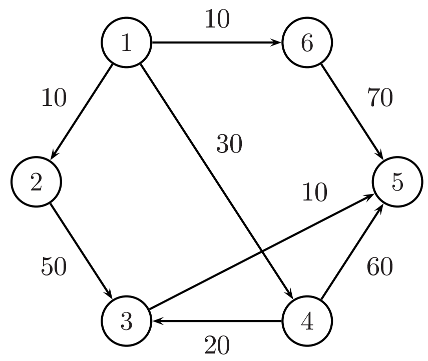
procedure Dijkstra;

begin (* Αρχικοποίηση *)
  $S := \{1\}$; for i:=2 to n do begin D[i]:=cost[1,i]; P[i]:=1 end;
  for i:=2 to n-1 do
  begin
    select w from $V - S$ such that D[w] is minimum;
    $S := S + \{w\}$;
    for all v in $V - S$ do
      if D[v] > D[w] + C[w,v] then
        P[v] := w;
        D[v] := D[w]+C[w,v]
  end
end

Πολυπλοκότητα: $O(|V|^2)$

All-pairs shortest paths: $O(|V|^3)$

Αλγόριθμος Dijkstra: παράδειγμα



Βήμα	S	w	D					P				
			2	3	4	5	6	2	3	4	5	6
-	{1}	-	10	∞	30	∞	10	1	1	1	1	1
2	{1,2}	2		60	30	∞	10		2			
3	{1,2,6}	6		60	30	80					6	
4	{1,2,6,4}	4		50		80			4			
5	{1,2,6,4,3}	3				60					3	
6	{1,2,6,4,3,5}	5										

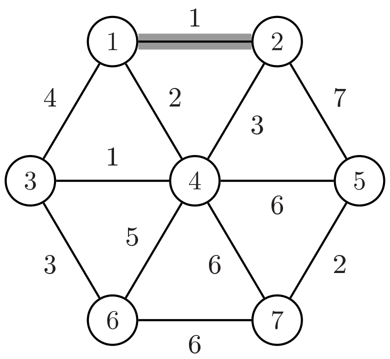
Μειονέκτημα Dijkstra: δεν δουλεύει όταν υπάρχουν ακμές με αρνητικά βάρη (γιατί;)

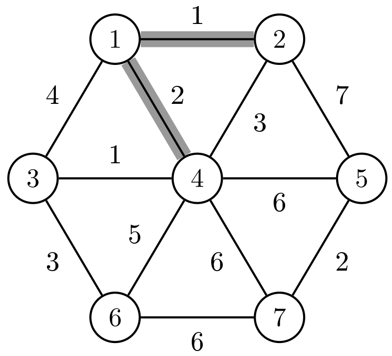
Ελάχιστο Συνδετικό Δένδρο (Minimum Spanning Tree - MST)

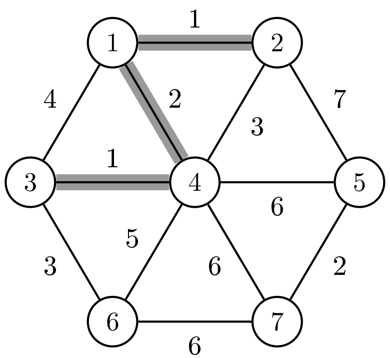
Αλγόριθμος Prim: Διαλέγουμε κάθε φορά την πλευρά ελαχίστου κόστους έτσι ώστε ο νέος υπογράφος να παραμένει δέντρο.

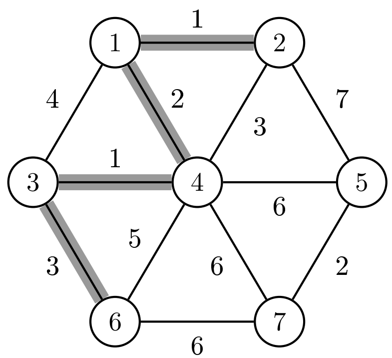
Αλγόριθμος Kruskal: Διαλέγουμε κάθε φορά την πλευρά ελαχίστου κόστους έτσι ώστε ο νέος υπογράφος να μην έχει κύκλους.

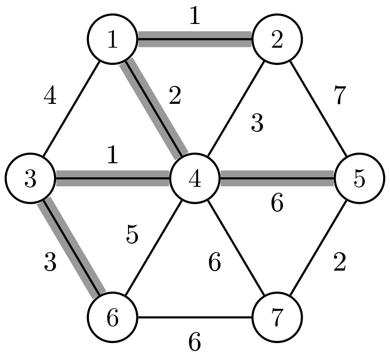
Αλγόριθμος Prim

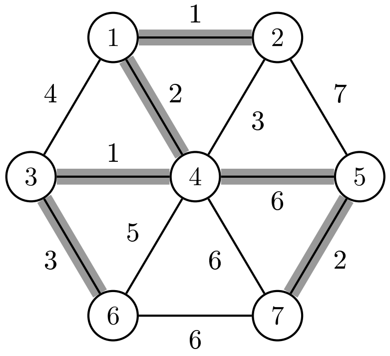










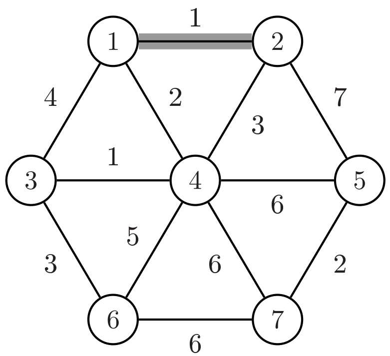


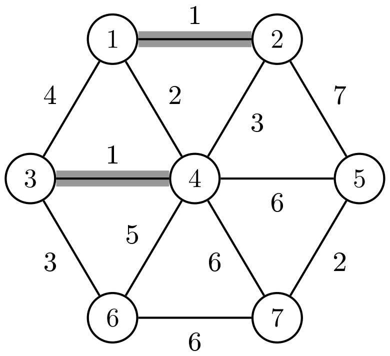
Αλγόριθμος Prim: υλοποίηση

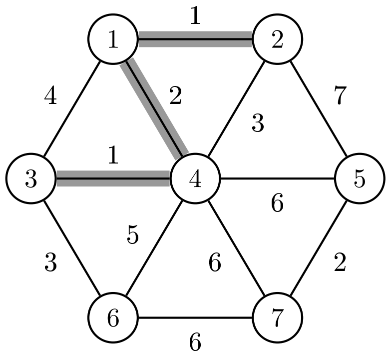
Κάθε φορά επιλέγεται ο κόμβος με την ελάχιστη απόσταση από το μέχρι στιγμής κατασκευασμένο δένδρο και προστίθεται στο δένδρο.

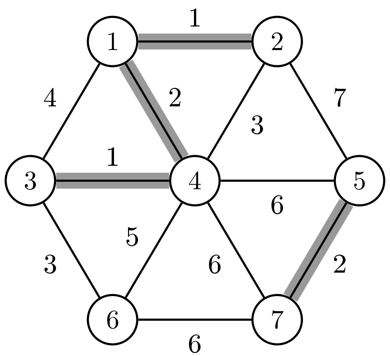
Πολυπλοκότητα: $O(|V|^2)$

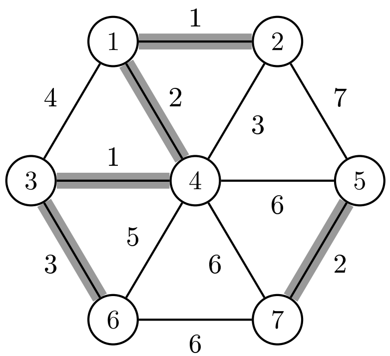
Αλγόριθμος Kruskal

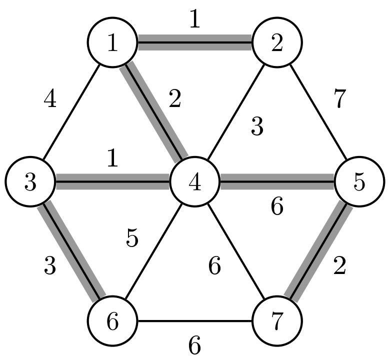



1
1 2
4 2 7
3
1
3 4 5
6
5
3 6 2
6 7
6


1
1 2
4 2 7
3
1
3 4 5
6
5
3 6 2
6 7
6


1
1 2
4 2 7
3
1
3 4 5
6
5
3 6 2
6 7
6


1
1 2
4 2 7
3
1
3 4 5
6
5
3 6 2
6 7
6



Αλγόριθμος Kruskal: υλοποίηση

Κάθε φορά επιλέγεται ακμή ελαχίστου κόστους και εάν δεν δημιουργεί κύκλο στο μέχρι στιγμής δάσος προστίθεται σε αυτό, αλλιώς απορρίπτεται.

Πολυπλοκότητα: $O(|E| \log |E|)$