Αλγόριθμοι Δικτύων και Πολυπλοκότητα

Μια εισαγωγή σε γραφοθεωρητικά προβλήματα

Άρης Παγουρτζής

Ε.Μ.Π. — Μ.Π.Λ.Α.

Ευχαριστίες: μέρος των διαφανειών αυτών προέρχεται από τις Σημειώσεις Ε. Ζάχου για το μάθημα "Αλγόριθμοι και Πολυπλοκότητα" της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών Ε.Μ.Π.

Επισκόπηση

- Υπολογιστική πολυπλοκότητα και προσεγγισιμότητα γραφοθεωρητικών προβλημάτων με εφαρμογές σε δίκτυα: Matching, Shortest Paths, Vertex Cover, Traveling Salesman Problem, Steiner tree, Maximum Flow, Maximum Edge-Disjoint Paths, Multicommodity Flow, Facility Location, Multicut, k-Center, Scheduling, Clustering.

- Κατανεμημένα πρωτόκολλα σε ασύρματα δίκτυα γνωστής ή άγνωστης τοπολογίας (ad hoc, wireless, sensor networks). Μετάδοση μηνύματος (broadcasting, k-selection, 'gossiping'), δρομολόγηση (compact routing, geometric routing), αρχικοποίηση (initialization), εκλογή αρχηγού, τοπικοί υπολογισμοί.

Επισκόπηση (συν.)

- Παράλληλοι / κατανεμημένοι υπολογισμοί με χρήση μηνυμάτων (message passsing). Μοντέλα BSP, LogP, Map-Reduce.
- Προβλήματα χρονοδρομολόγησης (scheduling) και υπολογισμού ροής. Οπτικά δίκτυα: δρομολόγηση και ανάθεση συχνοτήτων, δίκτυα πολλαπλών ινών. Εξερεύνηση γράφων, αλγόριθμοι πλοήγησης, προγραμματισμός δρομολογίων οχημάτων.
- Θεωρία παιγνίων: μη-συνεργατικά μοντέλα, εγωιστική δρομολόγηση, ισορροπία Nash, "κόστος της αναρχίας", ηλεκτρονικές δημοπρασίες, mechanism design, truthfulness.

Βιβλιογραφία: αλγόριθμοι

1. Vijay V. Vazirani. **Approximation Algorithms**. Springer, 2001.
2. David P. Williamson, David B. Shmoys. **The Design of Approximation Algorithms**. Cambridge University Press, 2010. (available online)
3. Dorit S. Hochbaum. **Approximation Algorithms for NP-Hard Problems**. PWS Publishing Company, 1997.
4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest: **Introduction to algorithms**. The MIT Press/ McGraw-Hill Book Company, 1989. (or 2nd edition.)
5. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani: **Algorithms**. MacGraw-Hill, 2006. (Διατίθεται ελεύθερα στο διαδίκτυο).

Βιβλιογραφία: δίκτυα και κατανεμημένοι υπολογισμοί

1. Roger Wattenhofer. **Principles of Distributed Computing**. ETH Zuerich course notes, 2011.
2. Nancy A. Lynch. **Distributed Algorithms**. Morgan Kaufmann Publishers, San Mateo, CA, 1996.
3. Robert E. Tarjan. **Data Structures and Network Algorithms**. SIAM, 1983.
4. Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin. **Network flows: Theory, Algorithms, and Applications**. Prentice-Hall, 1993.
5. Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. **Algorithmic Game Theory**. Cambridge University Press, New York, NY, USA.

Εισαγωγή

- Γράφοι
- Προβλήματα, Αλγόριθμοι, Πολυπλοκότητα
- Συμβολισμοί O, Ω, Θ
- Πολυωνυμικοί αλγόριθμοι: κύκλος Euler, διάσχιση γράφων, συντομότερα μονοπάτια, ελάχιστο συνδετικό δένδρο (minimum spanning tree), μέγιστη ροή, ταίριασμα (matching), χρωματισμός ακμών σε διμερείς γράφους.

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*

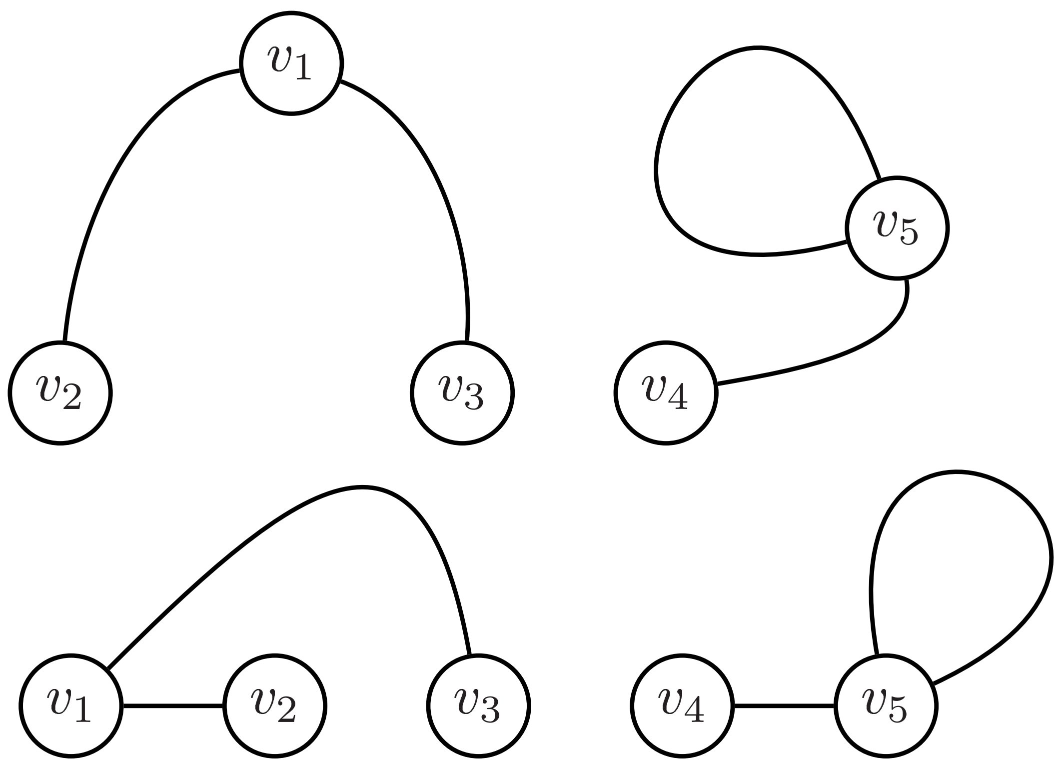
$$E \subseteq \binom{V}{2}$$

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

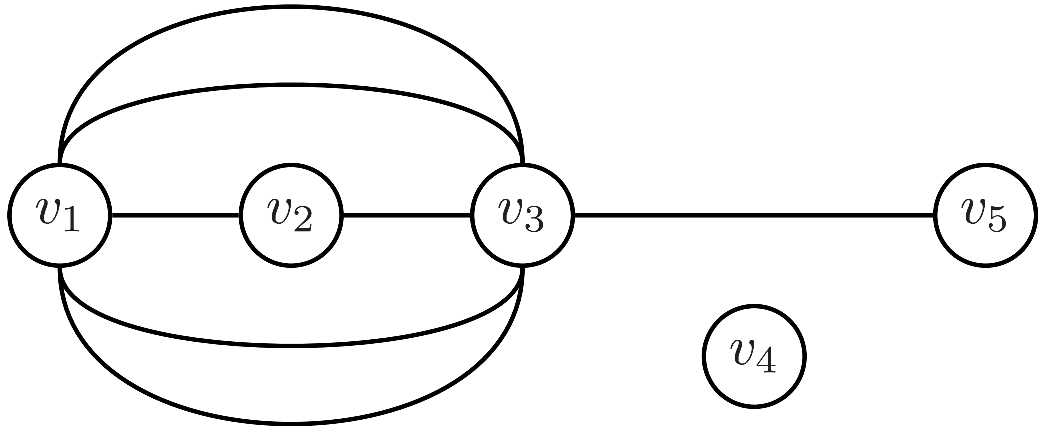
E: ακμές ή πλευρές (edges).

Παράδειγμα Γράφου

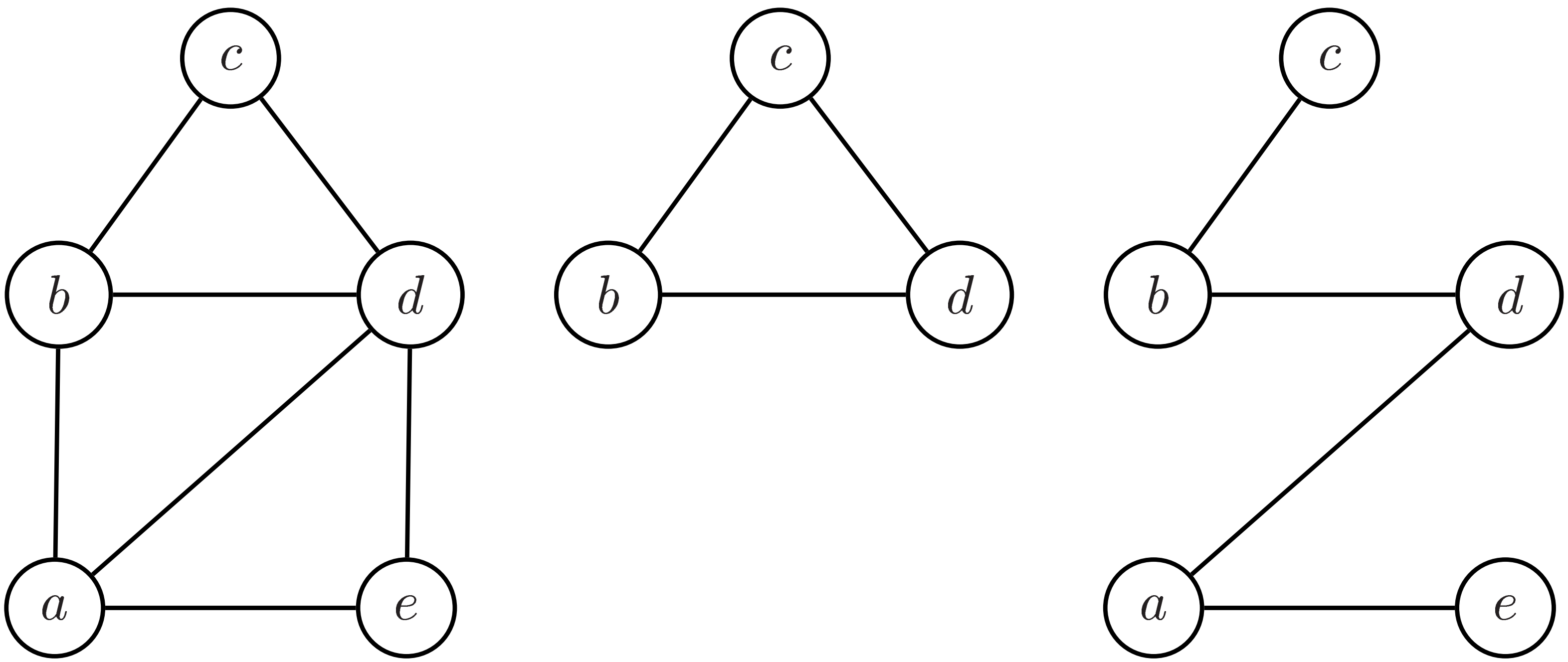
$$E = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_4, v_5\}, \{v_5, v_5\}\}$$



Πολυγράφημα



Υπογράφοι



Δρόμοι, μονοπάτια, κύκλοι

Δρόμος (walk): έγκυρη ακολουθία κορυφών-ακμών.

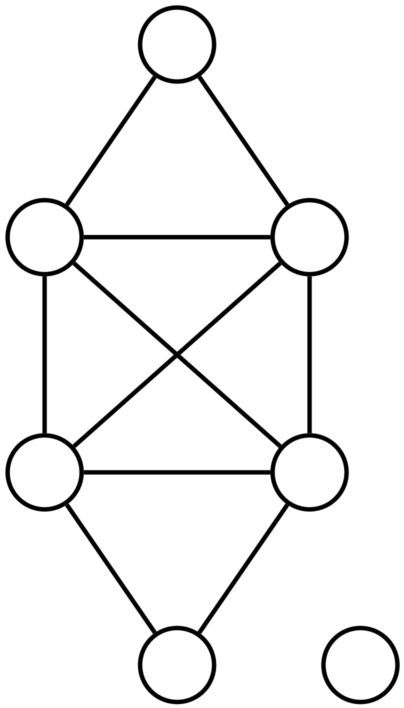
Μονοπάτι (path): δρόμος χωρίς επαναλήψεις ακμών.

Απλό μονοπάτι (simple path): μονοπάτι χωρίς επαναλήψεις κορυφών.

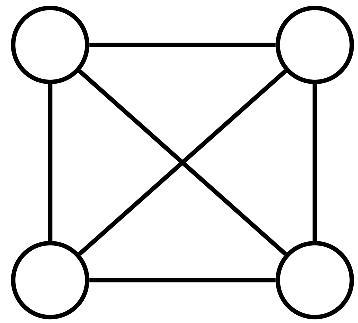
Κύκλος (cycle): κλειστό μονοπάτι. Απλός κύκλος: κλειστό απλό μονοπάτι.

Μήκος δρόμου: το πλήθος των ακμών του.

Γράφοι Euler, Hamilton



Γράφος Euler



Γράφος Hamilton

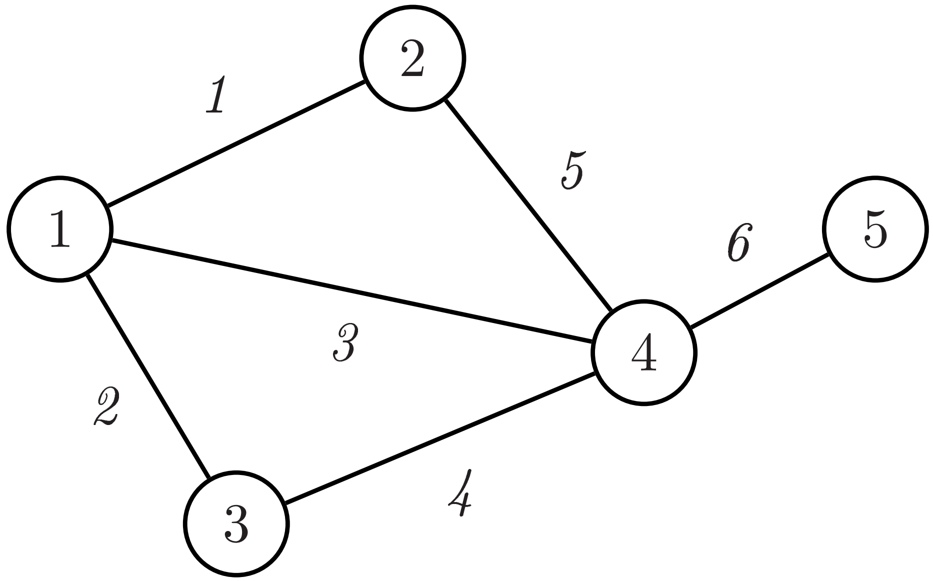
Παράσταση Γράφου

Πίνακας γειτνίασης (adjacency matrix)

Πίνακας πρόσπτωσης (incidence matrix)

Λίστες γειτνίασης (adjacency lists): αποδοτική παράσταση σε αραιούς γράφους.

Παράσταση με λίστες γειτνίασης



$$
\begin{array}{lllll}
[1]\rightarrow & 2 & 3 & 4 & \\
[2]\rightarrow & 1 & 4 & & \\
[3]\rightarrow & 1 & 4 & & \\
[4]\rightarrow & 1 & 2 & 3 & 5 \\
[5]\rightarrow & 4 & & &
\end{array}
$$

ʼΕνας ενδιαφέρων πίνακας

- Laplacian matrix: $Q(G) = D(G) - A(G)$
- $Q(G) = E'(G) {\cdot} E'^T(G)$ $\quad$ (E': προσανατολισμένος πίνακας πρόσπτωσης)
- Χρησιμεύει, μεταξύ άλλων, στον υπολογισμό του πλήθους των spanning trees (Kirchhoff's matrix tree theorem):

$$t(G) = \frac{1}{n}\lambda_1\lambda_2\ldots\lambda_{n-1} = \det(Q_{11}(G))$$

- Επίσης, το πλήθος των συνεκτικών συνιστωσών ενός γράφου ισούται με το πλήθος των μηδενικών ιδιοτιμών του $Q(G)$.

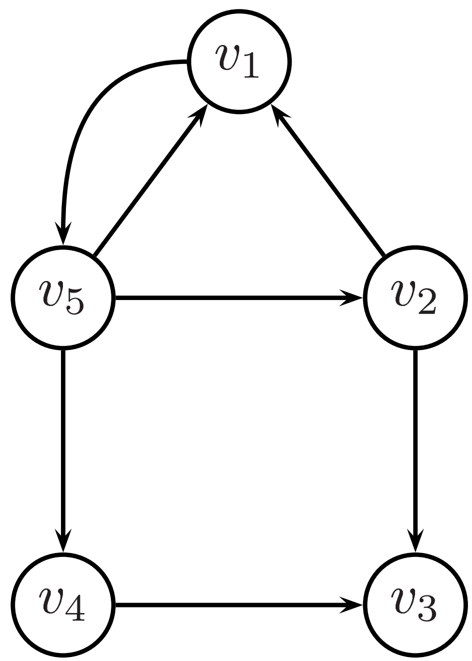
Περαιτέρω εφαρμογές: spectral graph theory (βλ. και Unique Games Conjecture.

Unique Games Conjecture [Khot'02]

- Έστω γράφος με περιορισμούς στις ακμές: π.χ. χρωματικούς.
- Κάθε ανάθεση τιμής σε κόμβο καθορίζει μοναδικά τιμές γειτόνων.
- Το πρόβλημα ικανοποίησης όλων των περιορισμών είναι στο **P**.
- Το πρόβλημα μεγιστοποίησης δεν είναι!
- UCG: Για κάθε $\delta > 0$ και $\varepsilon > 0$ υπάρχει Unique Game όπου είναι **NP**-hard να διακρίνει κανείς αν $OPT > 1 - \varepsilon$ ή αν $OPT < \delta$.
- Ακόμη και αν δίνεται στιγμιότυπο με 99% των περιορισμών να μπορούν να ικανοποιηθούν, είναι **NP**-hard να ικανοποιηθεί έστω και το 1%.
- Πλούσια έρευνα: expander graphs, spectral graph theory.

Κατευθυνόμενος γράφος (directed graph)

$$E \subseteq V \times V$$



Αλλες έννοιες

Συνδεδεμένες κορυφές, παραγόμενος (induced) υπογράφος,

Συνεκτικότητα (connectivity), συνεκτικές συνιστώσες (connected components).

Κατευθυνόμενος γράφος: ισχυρή και ασθενής συνεκτικότητα.

Δένδρο (tree): συνεκτικός ακυκλικός γράφος.

Πλήρης γράφος (K_n), διμερής γράφος (πλήρης διμερής: $K_{n,m}$).

Επίπεδος γράφος (ανν δεν περιέχει K_5, $K_{3,3}$ ως ελάσσονα γραφήματα)

Χαρακτηρισμοί με αποκλεισμό ελασσόνων (minors)

Έλασσον υπογράφημα (minor): κάθε γράφημα που προκύπτει από το αρχικό με διαγραφή κόμβου, διαγραφή ακμής, ή σύνθλιψη ακμής.

Σύνολο παρεμπόδισης ή αποκλεισμού: ένα σύνολο ελασσόνων υπογραφημάτων που ο αποκλεισμός τους χαρακτηρίζει ακριβώς μια οικογένεια γραφημάτων.

$\{K_5, K_{3,3}\}$: σύνολο αποκλεισμού για τους επίπεδους γράφους.

Θεώρημα Robertson-Seymour: κάθε οικογένεια γραφημάτων που είναι κλειστή ως προς minors χαρακτηρίζεται από πεπερασμένο σύνολο αποκλεισμού.

Πόρισμα: κάθε ιδιότητα που είναι κλειστή ως προς minors μπορεί να ελεγχθεί σε πολυωνυμικό χρόνο (αλλά συνήθως δεν ξέρουμε πώς)!

Υπολογιστικά Προβλήματα

Υπολογιστικό πρόβλημα: καθορισμός αντιστοίχισης έγκυρων *δεδομένων εισόδου (στιγμιοτύπου)* σε *δεδομένα εξόδου (απαντήσεις / λύσεις)*.

Μαθηματική περιγραφή: σχέση (relation) μεταξύ συμβολοσειρών.

Παράδειγμα. Το πρόβλημα Satisfiability (SAT)

Προβλήματα απόφασης, προβλήματα βελτιστοποίησης.

Αλγόριθμος επίλυσης προβλήματος

Μηχανιστική διαδικασία παραγωγής απάντησης για κάθε έγκυρο στιγμιότυπο.

- Κάθε εκτέλεση είναι *πεπερασμένη*, δηλαδή τελειώνει ύστερα από έναν πεπερασμένο αριθμό διεργασιών ή βημάτων (finiteness).
- Κάθε κανόνας του ορίζεται επακριβώς και η αντίστοιχη διεργασία είναι συγκεκριμένη (definiteness).
- Έχει μηδέν ή περισσότερα μεγέθη *εισόδου* που δίδονται εξαρχής, πριν αρχίσει να εκτελείται ο αλγόριθμος (input).
- Δίδει τουλάχιστον ένα μέγεθος σαν αποτέλεσμα (*έξοδο*-output) που εξαρτάται κατά κάποιο τρόπο απ'τις αρχικές εισόδους.
- Είναι *μηχανιστικά* αποτελεσματικός, δηλαδή όλες οι διαδικασίες που περιλαμβάνει μπορούν να πραγματοποιηθούν με ακρίβεια και σε πεπερασμένο χρόνο èμε μολύβι και χαρτίη (effectiveness).

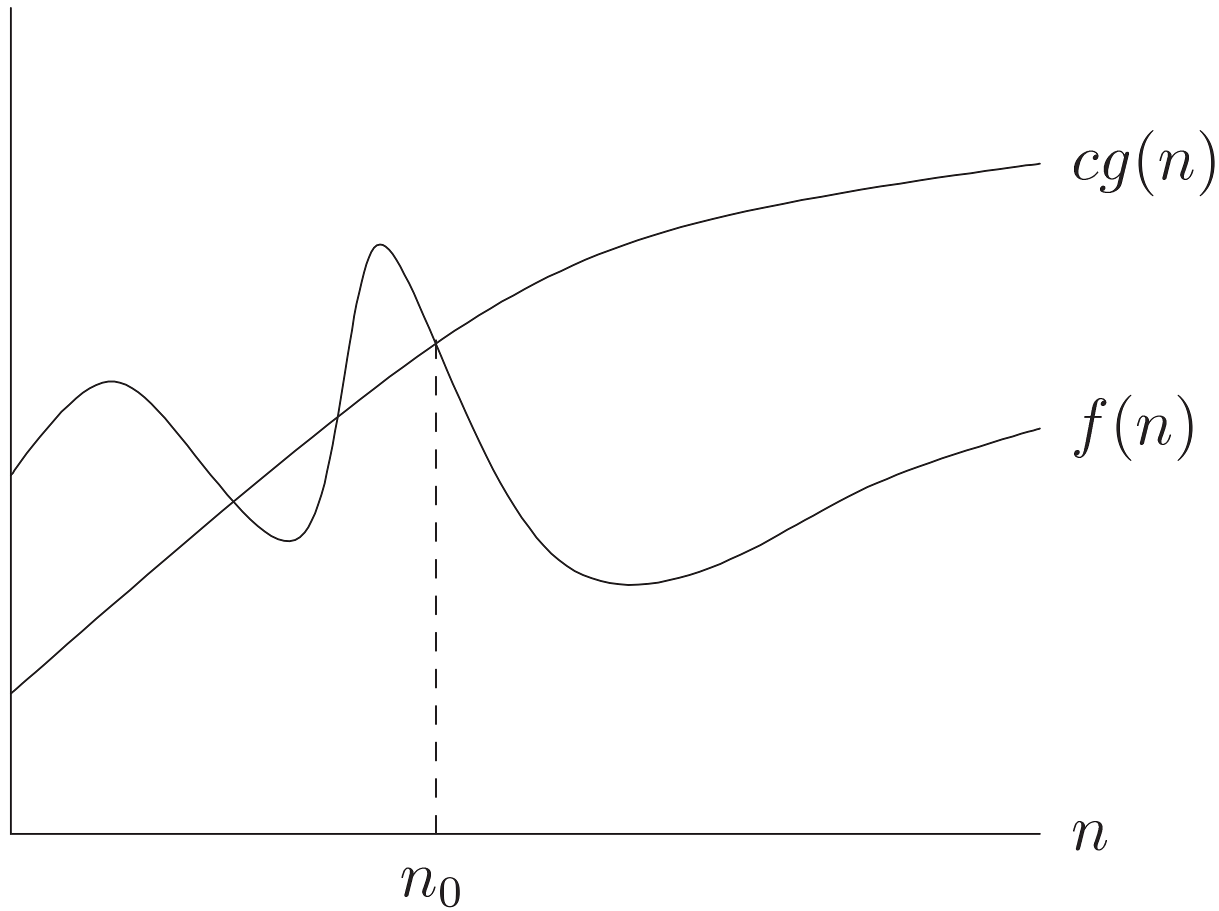
Πολυπλοκότητα Αλγορίθμου - Προβλήματος

Πολυπλοκότητα χειρότερης περίπτωσης

$$\text{κόστος αλγορίθμου } A(n) = \max_{\substack{\text{για όλες τις} \\ \text{δυνατές εισόδους } x \\ \text{μεγέθους } n}} \{\text{κόστος αλγορίθμου } A \text{ για την είσοδο } x$$

$$\text{κόστος προβλήματος } (n) = \min_{\substack{\text{για όλους τους} \\ \text{αλγόριθμους } A \text{ που} \\ \text{επιλύουν το} \\ \text{πρόβλημα}}} \{A(n)\}$$

Συμβολισμοί τάξης μεγέθους: συμβολισμός O



$$f = O(g)$$

Συμβολισμός O, o

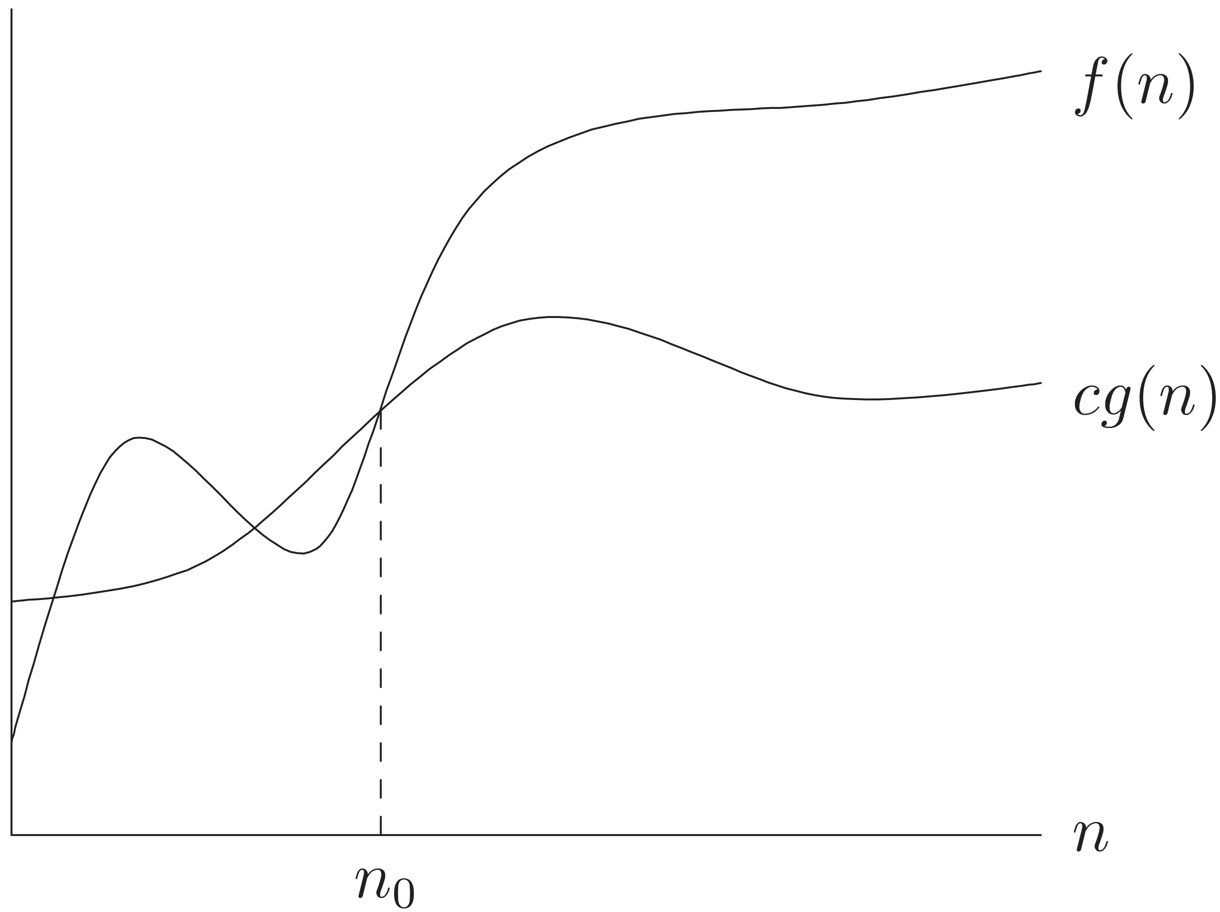
$$O(g) = \{f \mid \exists c > 0,\ \exists n_0 : \forall n > n_0 \ \ f(n) \leq cg(n)\}$$

$$o(g) = \{f \mid \forall c > 0,\ \exists n_0 : \forall n > n_0 \ \ f(n) \leq cg(n)\}$$

ή

$$o(g) = \{f \mid \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0\}$$

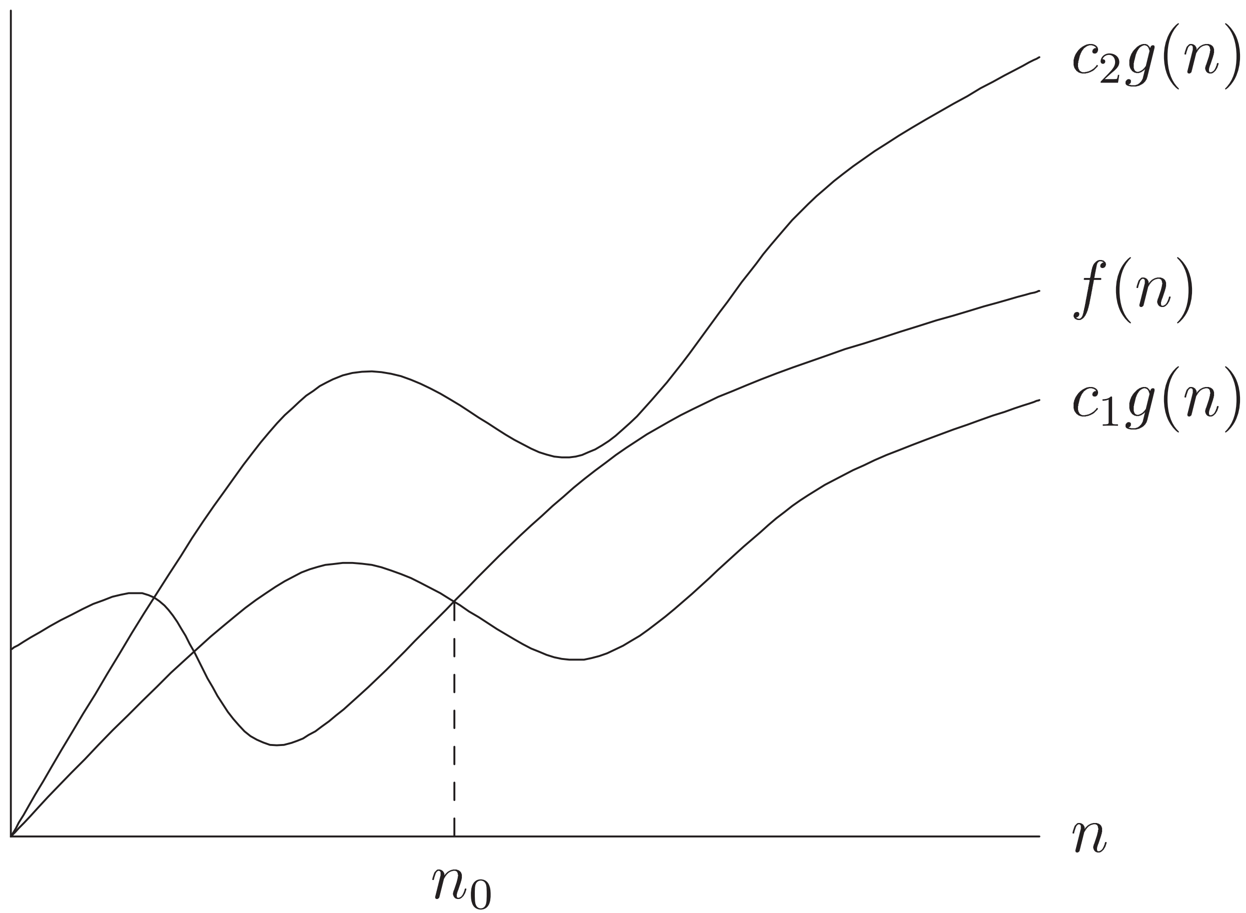
Συμβολισμός Ω, ω



$$\Omega(g) = \{f \mid \exists c > 0,\ \exists n_0 : \forall n > n_0\ \ f(n) \geq cg(n)\}$$

$$\omega(g) = \{f \mid \forall c > 0,\ \exists n_0 : \forall n > n_0\ \ f(n) \geq cg(n)\}$$

Συμβολισμός Θ



$$\Theta(g) = \{f \ \mid \ \exists c_1 > 0,\ \exists c_2 > 0,\ \exists n_0 : \forall n > n_0 \ \ c_1 \leq \frac{f(n)}{g(n)} \leq c_2\}$$

Ιεράρχηση

$$
\begin{gathered}
O(1) < O(\alpha(n)) < O(\log^* n) \\
< O(\log(n)) < O(\sqrt{n}) < O(n) \\
< O(n \log(n)) < O(n^2) < \ldots < O(poly) \\
< O(2^n) < O(n!) < O(n^n) < O(A(n))
\end{gathered}
$$

Αναδρομικές σχέσεις

$$T(n) = \begin{cases} a & \text{, για } n = 1 \\ 2T(n/2) + c & \text{, για } n > 1 \end{cases} \Rightarrow T(n) = n \cdot a + c \cdot (n-1) = O(n)$$

$$T(n) = \begin{cases} a & \text{για } n = 1 \\ 2T(n/2) + cn & \text{για } n > 1 \end{cases} \Rightarrow$$

$$T(n) = n \cdot a + \sum (\frac{n}{2^j}) \cdot c \cdot 2^j = n \cdot a + cn \log_2 n = O(n \log n)$$

Master Theorem

Έστω $a \geq 1$ και $b > 1$ σταθερές, $f(n)$ μια συνάρτηση, και η $T(n)$ ορίζεται στους μη αρνητικούς ακεραίους από την αναδρομική σχέση

$$T(n) = aT(n/b) + f(n)$$

(το n/b σημαίνει είτε $\lfloor n/b \rfloor$ είτε $\lceil n/b \rceil$). Τότε η $T(n)$ μπορεί να φραχτεί ασυμπτωτικά ως εξής:

1. $T(n) = \Theta(f(n))$, αν $f(n) = \Omega(n^{\log_b a+\varepsilon})$ για κάποια σταθερά $\varepsilon > 0$, και αν $af(n/b) \leq cf(n)$ για κάποια σταθερά $c > 1$ και όλα τα αρκετά μεγάλα n.

2. $T(n) = \Theta(f(n)\log_2 n)$, αν $f(n) = \Theta(n^{\log_b a})$.

3. $T(n) = \Theta(n^{\log_b a})$, αν $f(n) = O(n^{\log_b a-\varepsilon})$ για κάποια σταθερά $\varepsilon > 0$.

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/Cycle (HC)

Traveling Salesman Problem (TSP)

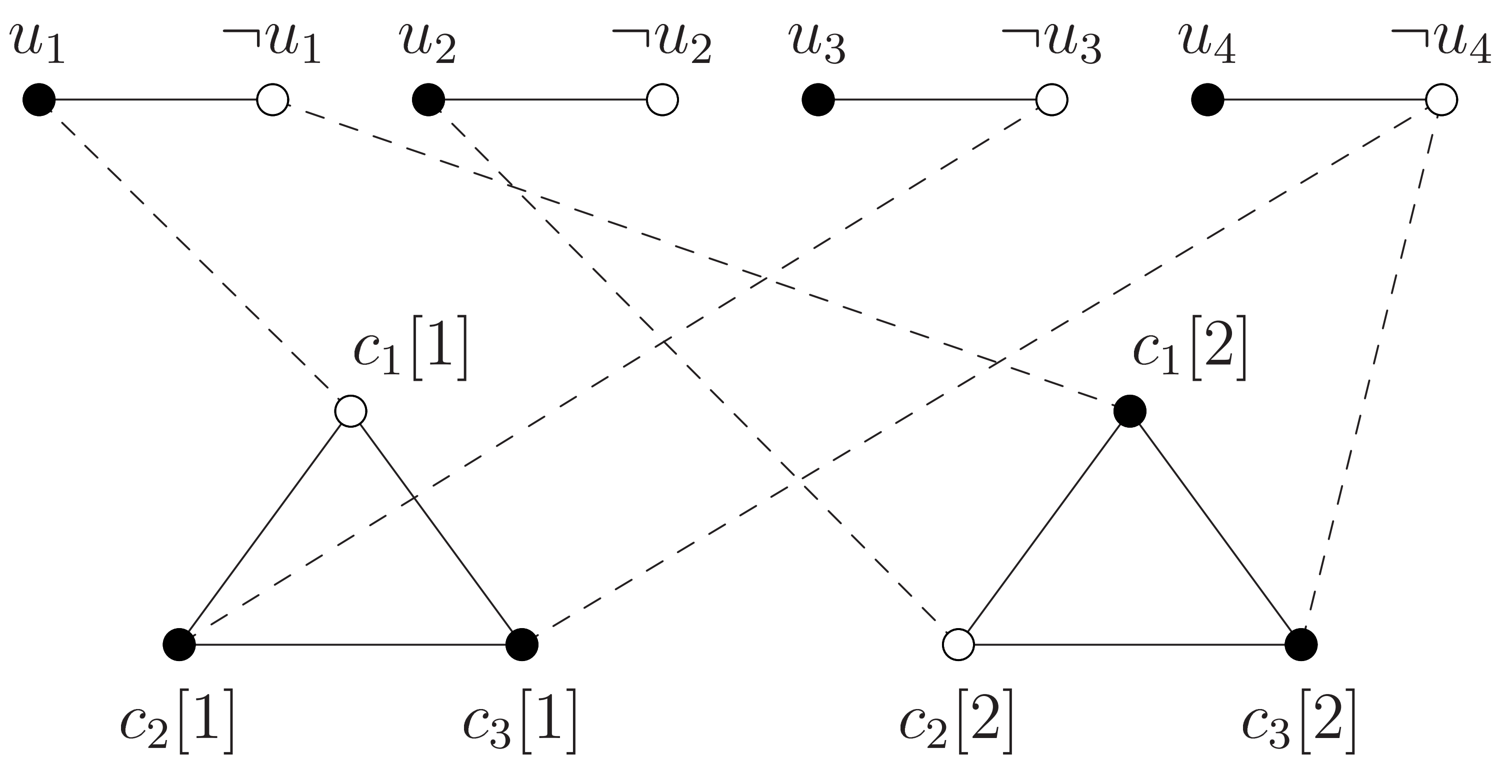
3-Colorability

Subgraph Isomorphism

3-Dimensional Matching (3DM)

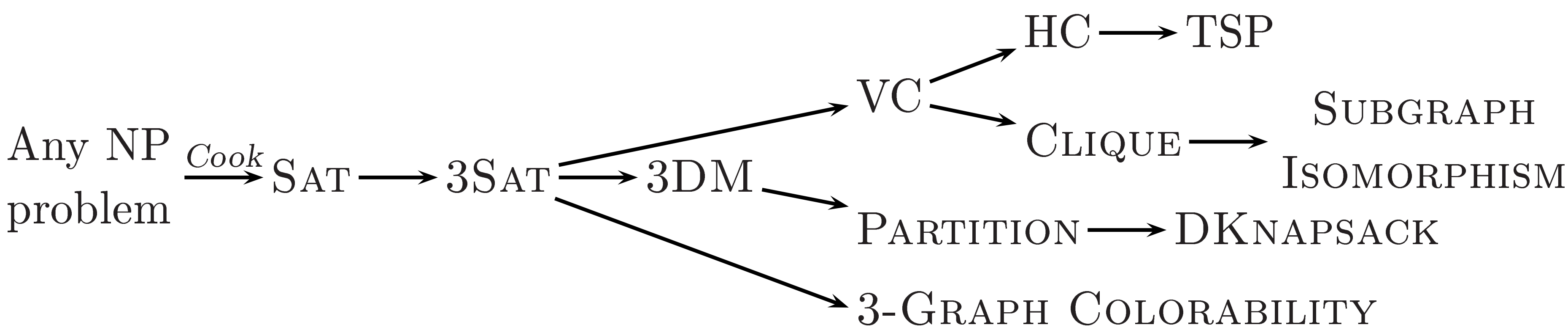
Αναγωγή 3SAT $\leq$ VERTEX COVER

$$\Phi\colon (u_1 \vee \neg u_3 \vee \neg u_4) \wedge (\neg u_1 \vee u_2 \vee \neg u_4)$$



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.

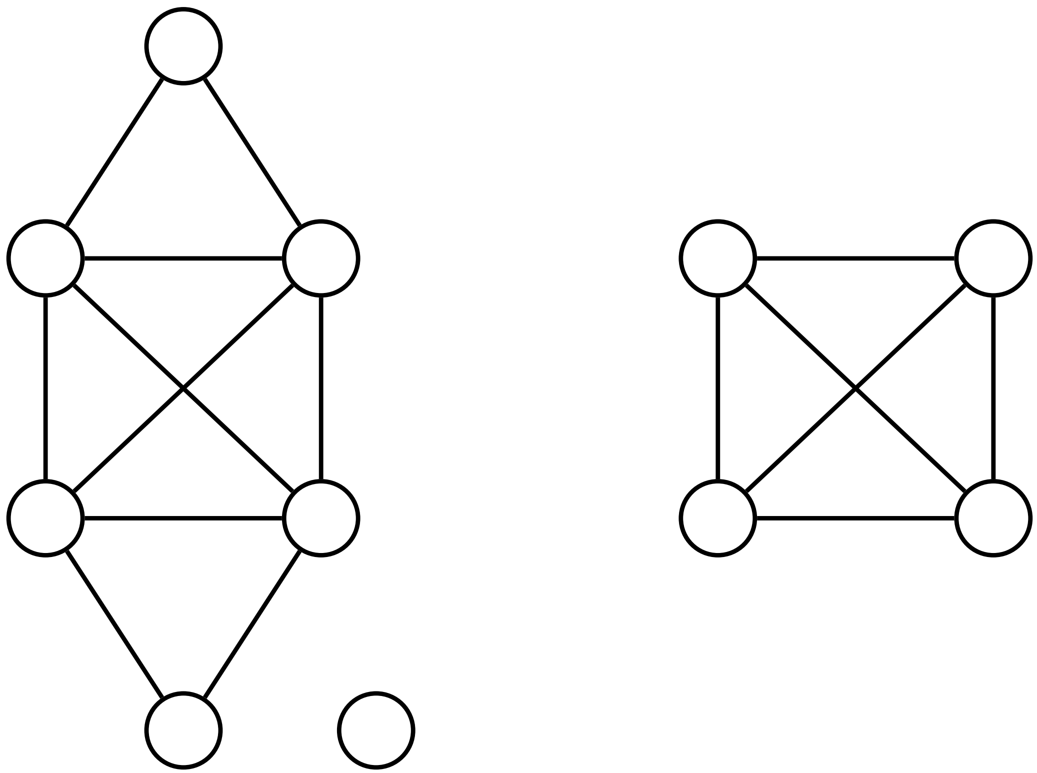
Συντομότερα μονοπάτια. Συνεκτικές συνιστώσες.

Ελάχιστο συνδετικό δένδρο (minimum spanning tree).

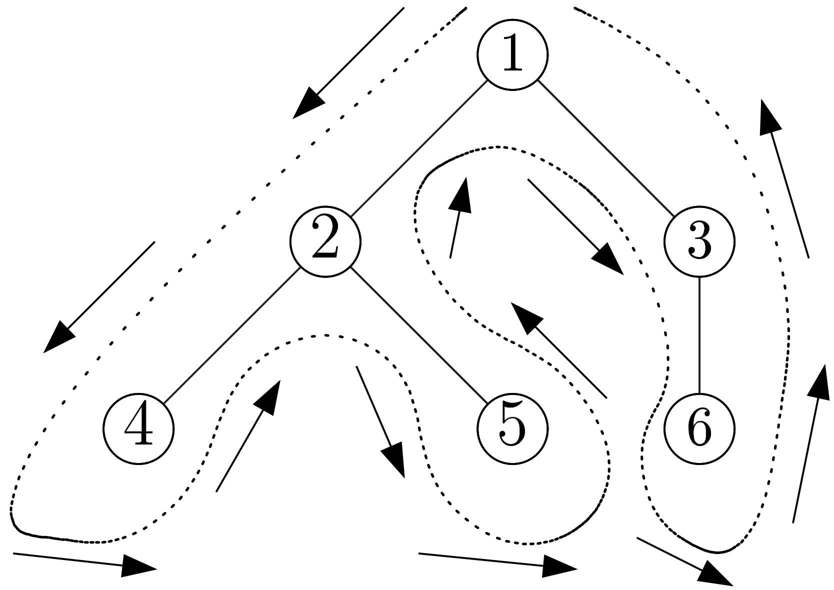
Μέγιστη ροή. Perfect matching.

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: $O(|V| + |E|)$.

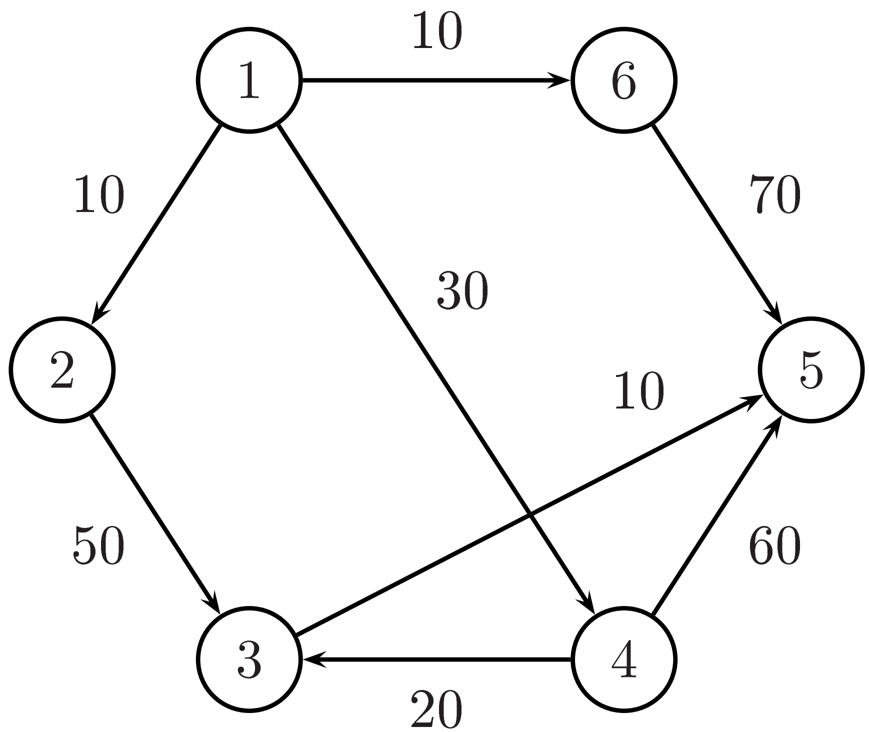
Συντομότερα μονοπάτια: Dijkstra

```
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;)

Αλγόριθμος Bellman-Ford

Εκτελείται σε $|V| - 1$ στάδια.

Στο στάδιο i ενημερώνεται κάθε κόμβος v (που βρίσκεται σε απόσταση το πολύ i ακμών από τον αρχικό) με το συντομότερο μονοπάτι από τον s στον v που έχει το πολύ i ακμές.

Αυτό επιτυγχάνεται με εκτέλεση για κάθε ακμή $(w, v) \in E$ της εντολής:

if $D[v] > D[w] + C[w, v]$ then
begin
  $P[v] := w;$
  $D[v] := D[w] + C[w, v]$
end

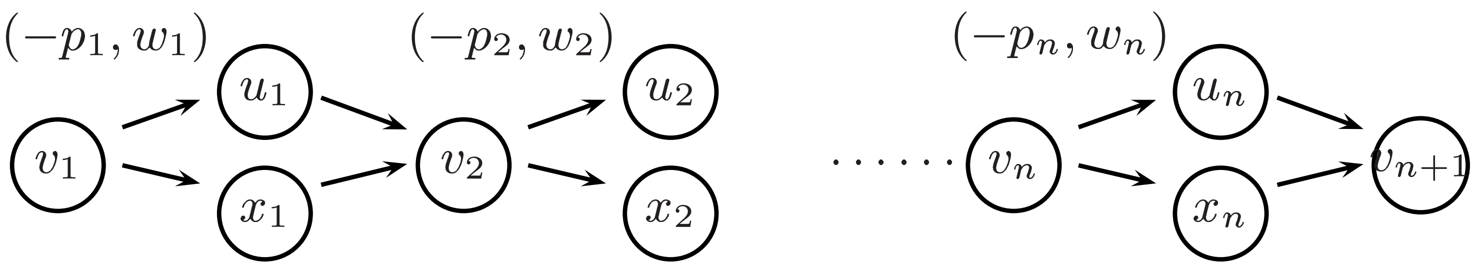
Πολυπλοκότητα: $O(|V||E|)$

Παρατήρηση: δεν δουλεύει αν υπάρχουν κύκλοι αρνητικού βάρους. Μπορεί όμως να τους εντοπίζει με κατάλληλη τροποποίηση (*Ασκηση: βρείτε πώς*).

Restricted (Constrained) Shortest Path: $\mathbf{NP}$-hard

Το πρόβλημα ορίζεται σε γράφους με δύο συναρτήσεις κόστους στις ακμές (π.χ. κόστος σε χρήμα c_{ij} και σε χρόνο t_{ij}). Ζητείται να ελαχιστοποιηθεί ο συνολικός χρόνος χωρίς να ξοδευτούν πάνω από ένα ποσό K χρημάτων (ή, ισοδύναμα, να ελαχιστοποιηθεί το κόστος χωρίς ο χρόνος να ξεπεράσει κάποιο όριο).

Αναγωγή από το D-Knapsack:



Good news: admits an FPTAS [Warburton, 1987].

Ελάχιστο Συνδετικό Δένδρο
(Minimum Spanning Tree - MST)

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

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

Κοινή ιδέα των δύο αλγορίθμων: ξεκινώντας από τον γράφο χωρίς ακμές, και ενώνοντας επαναληπτικά δύο οποιαδήποτε συμπληρωματικά υποσύνολα κόμβων S και $V \setminus S$ που ακόμη δεν έχουν ακμή μεταξύ τους με ελαχίστου βάρους ακμή καταλήγουμε σε ελάχιστο συνδετικό δένδρο.

Ελάχιστο Συνδετικό Δένδρο (συν.)

Γιατί δουλεύει η ιδέα; (ένωση ενός οποιουδήποτε υποσυνόλου κόμβων S με υπόλοιπο γράφο $V \setminus S$ με την ελαφρύτερη ακμή)

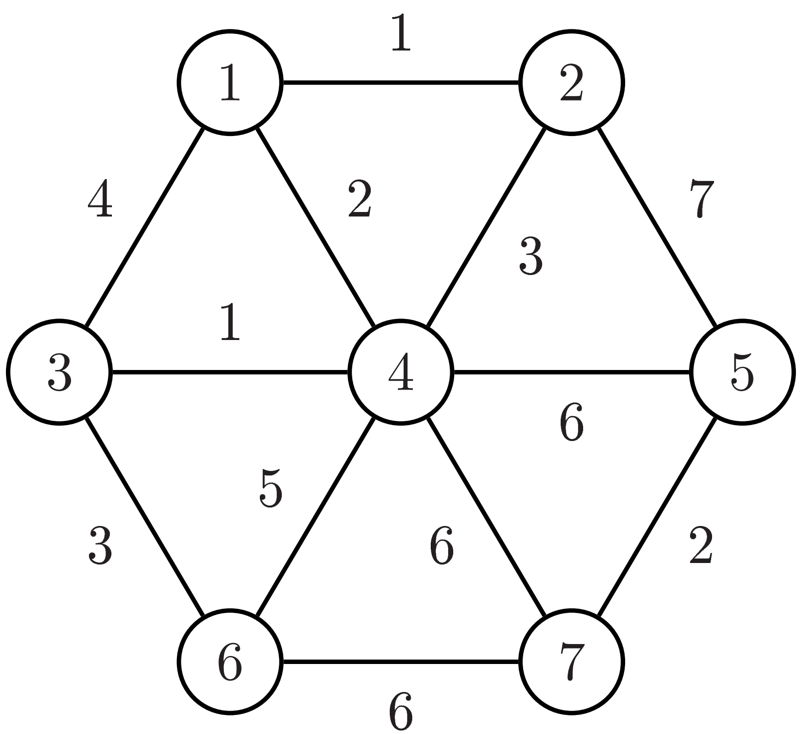
Λήμμα. Ένα σύνολο ακμών που είναι υποσχόμενο (υποσύνολο ενός MST) παραμένει υποσχόμενο αν του προσθέσουμε ελάχιστη ακμή μεταξύ *οποιουδήποτε* συνόλου συνεκτικών συνιστωσών του γράφου (που ορίζεται από τις ακμές του συνόλου) και του υπόλοιπου γράφου.

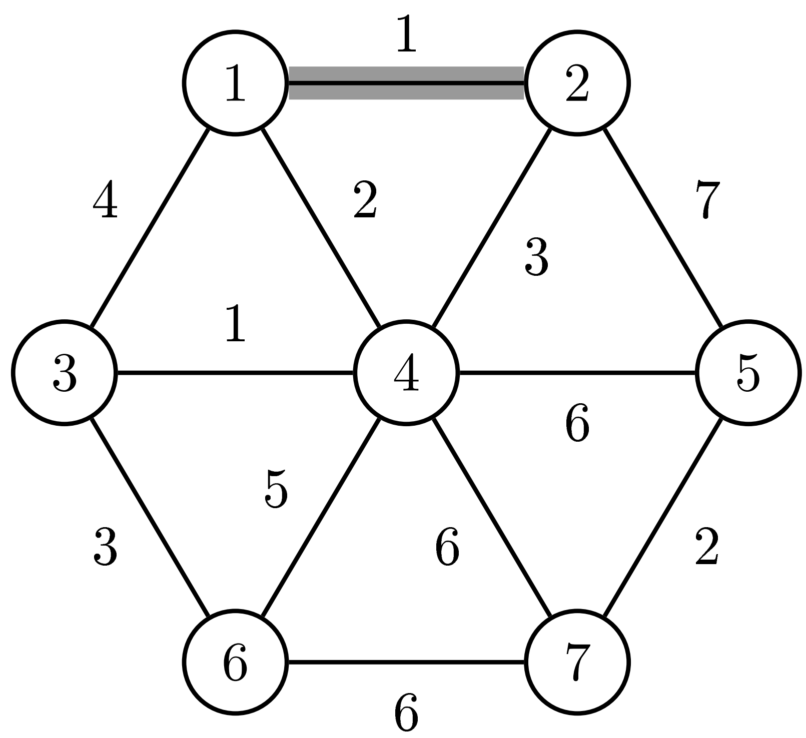
Απόδειξη. Στον πίνακα.

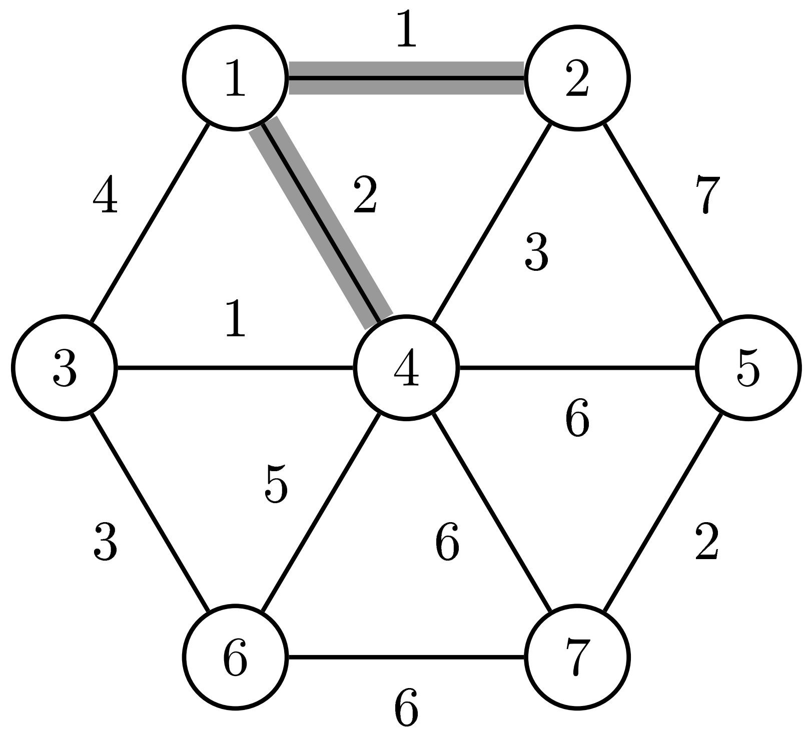
Άλλη εφαρμογή της ιδέας: Αλγόριθμος Borůvka, προσφέρεται για παραλληλοποίησ
Κάθε συνεκτική συνιστώσα (connected component) συνδέεται με την ελαφρύτερη δυνατή ακμή με κάποια από τις υπόλοιπες συνιστώσες. Αρχικά κάθε κόμβος είναι συνιστώσα. Σε κάθε 'γύρο' ο αριθμός των συνιστωσών μειώνεται στο μισό. Χρειάζεται διαφορετικά βάρη στις ακμές, ή τρόπο επίλυσης 'ισοπαλιών'.

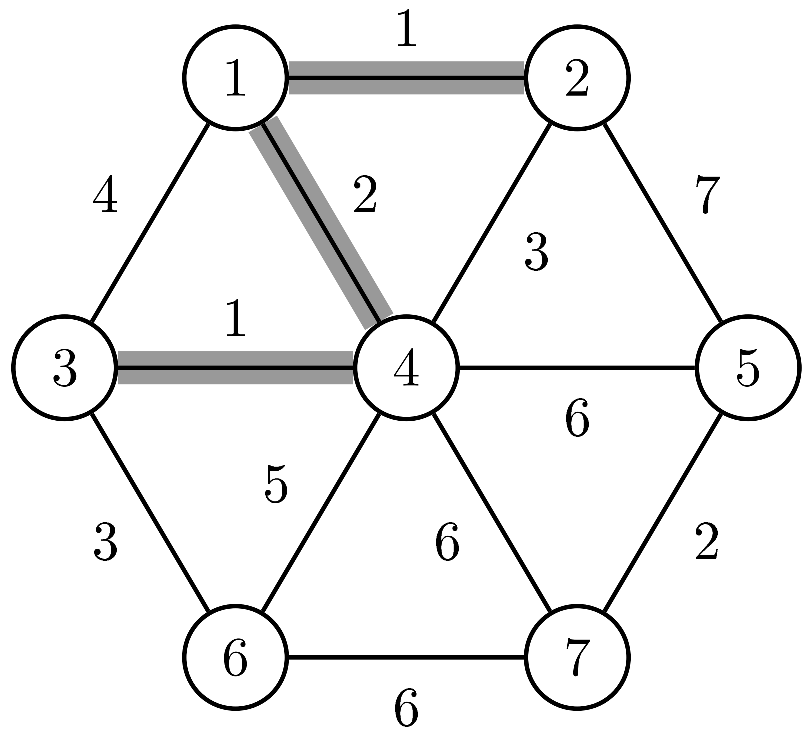
Πολυπλοκότητα: $O(|E| \log |V|)$.

Αλγόριθμος Prim: παράδειγμα

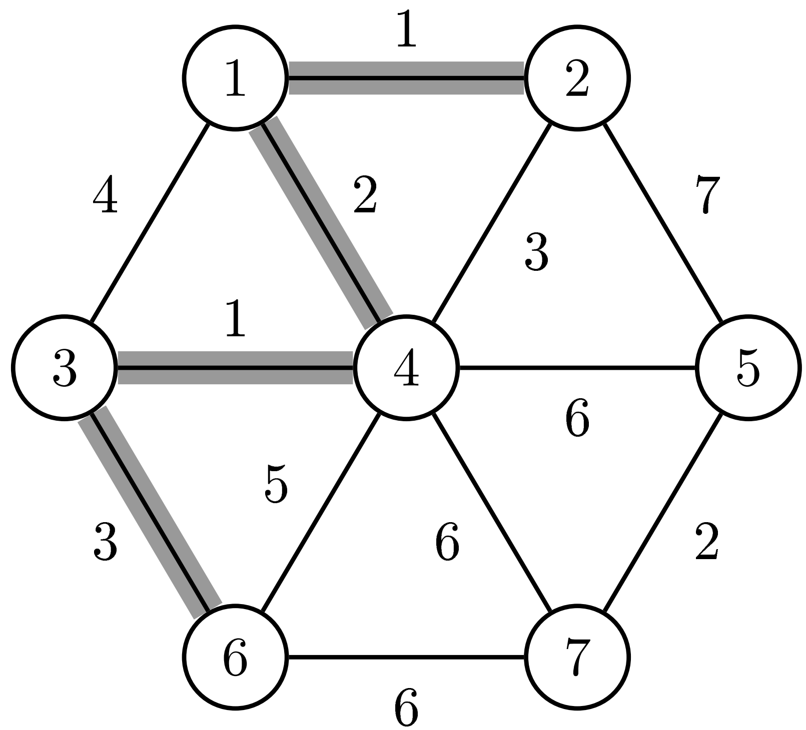


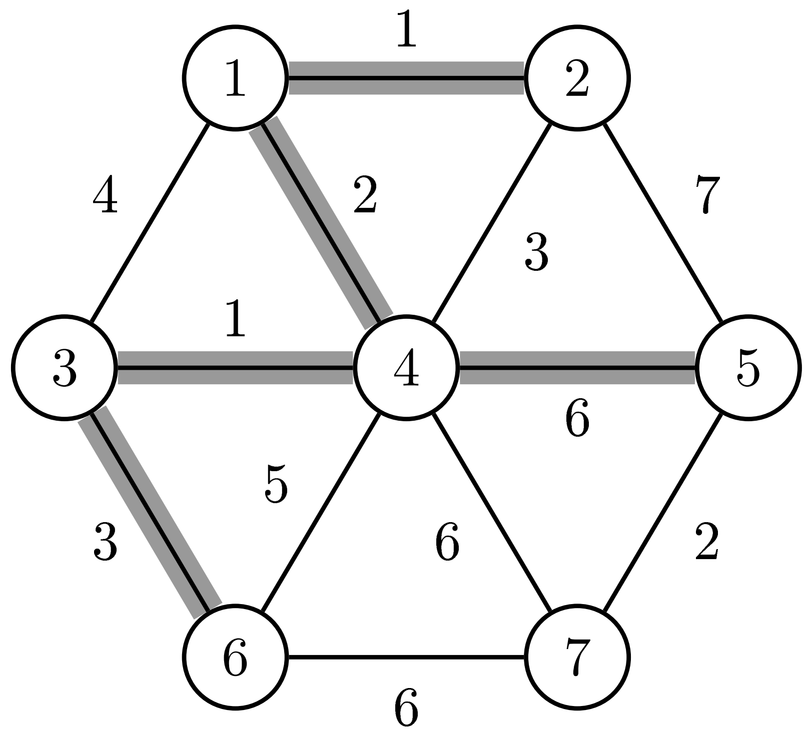

1
2
3
4
5
6
7
1
4
2
3
7
1
6
5
3
6
2
6

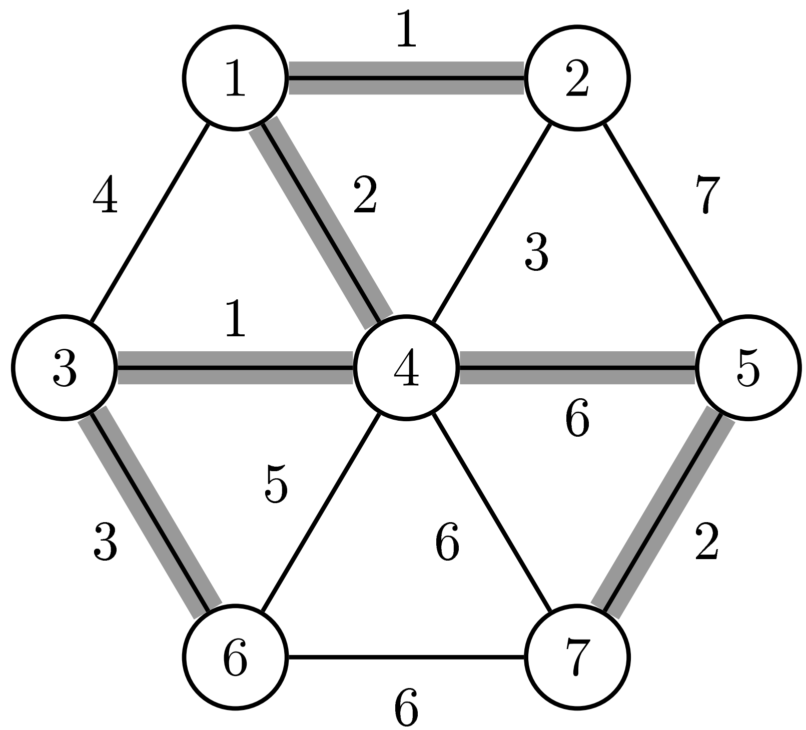

1
2
3
4
5
6
7
1
4
2
3
7
1
6
5
3
6
2
6



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

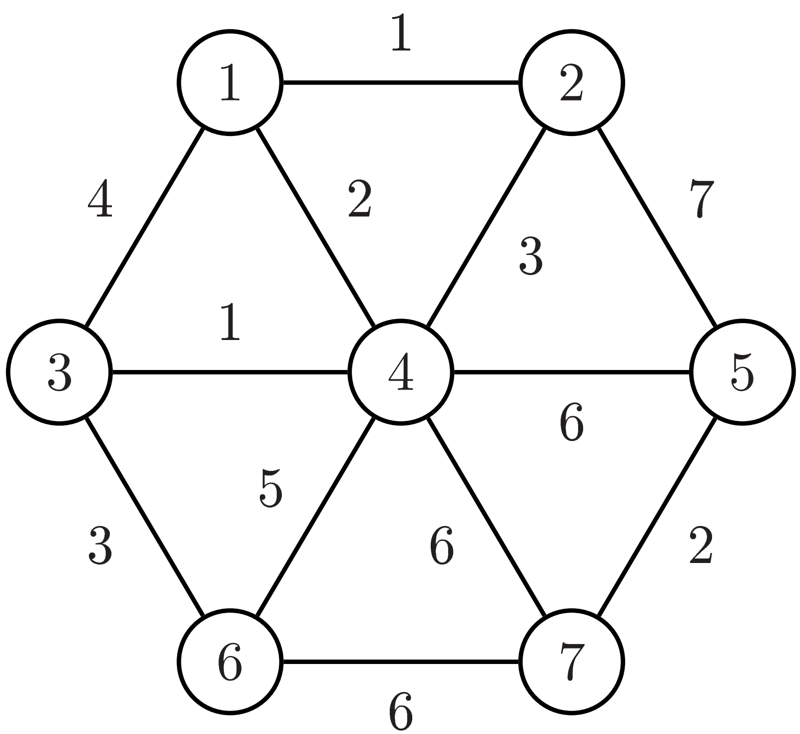

1
2
3
4
5
6
7
1
4
2
3
7
1
6
5
3
6
2
6

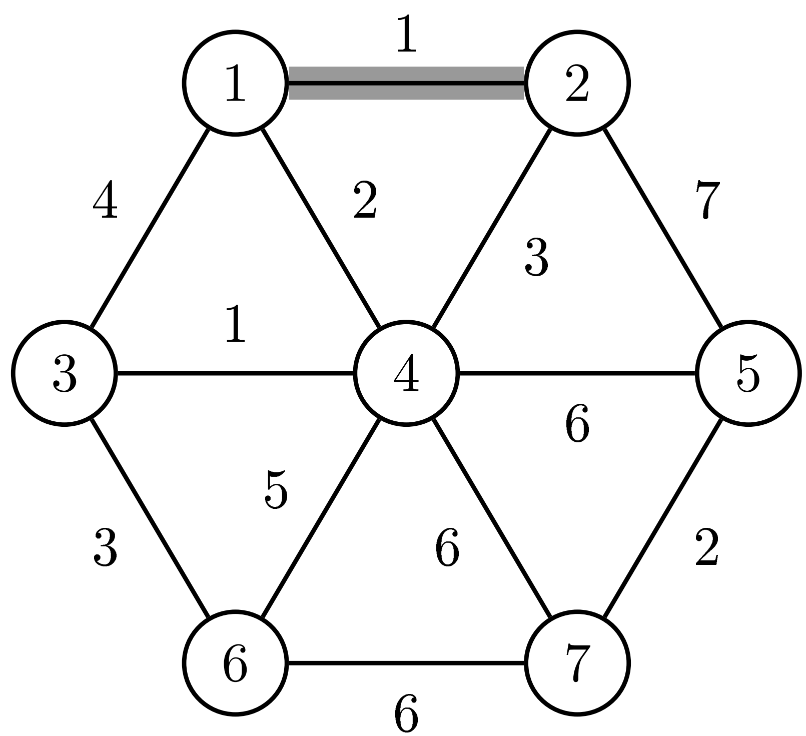
Αλγόριθμος Prim: υλοποίηση

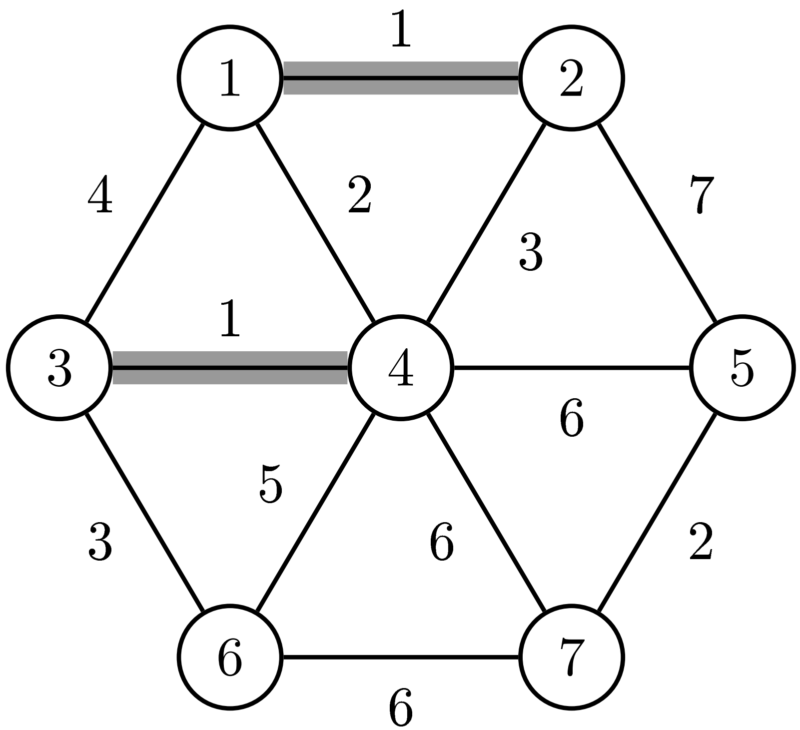
Κάθε φορά επιλέγεται ο κόμβος με την ελάχιστη απόσταση από το μέχρι στιγμής κατασκευασμένο δένδρο και προστίθεται στο δένδρο.

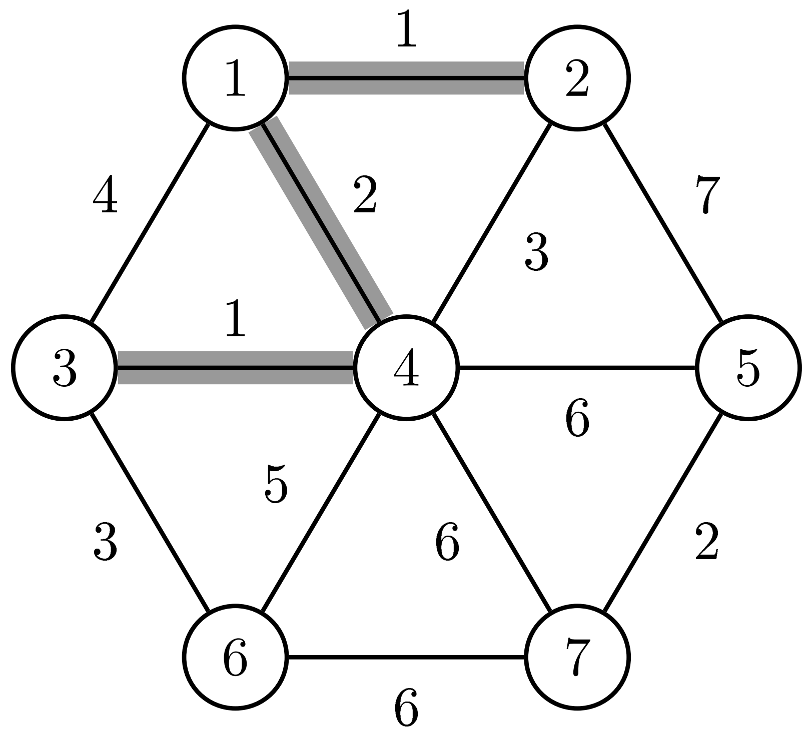
Πολυπλοκότητα: $O(|V|^2)$ (απλή υλοποίηση), $O(|E|\log|V|)$ με binary heap, $O(|V|\log|V| + |E|)$ με fibonacci heap,

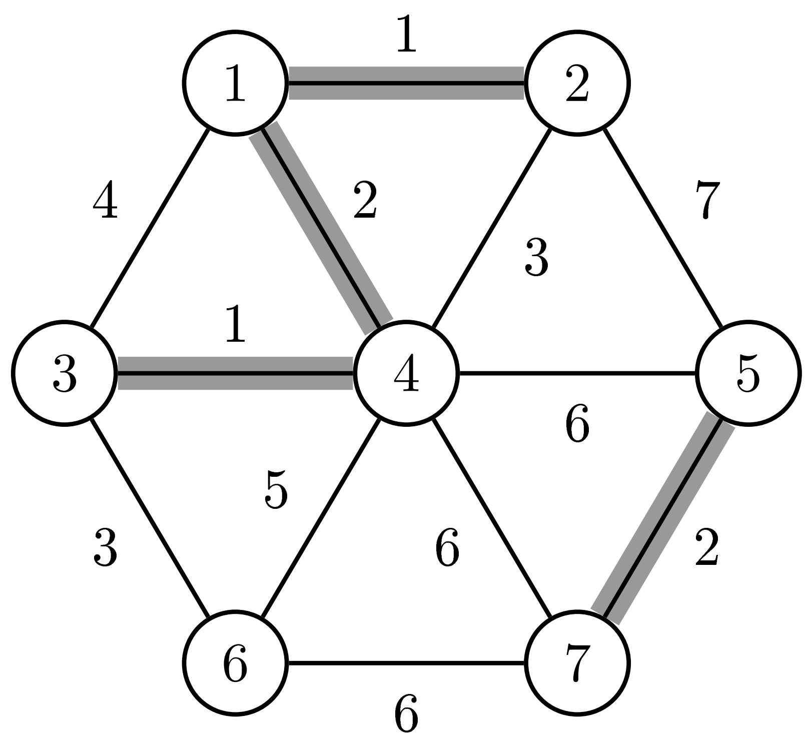
Αλγόριθμος Kruskal: παράδειγμα

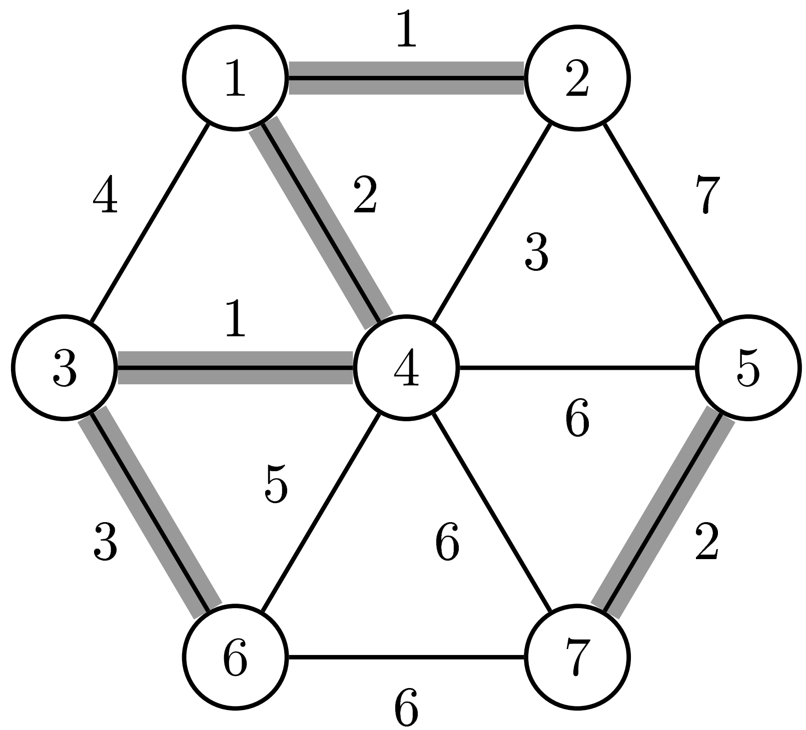


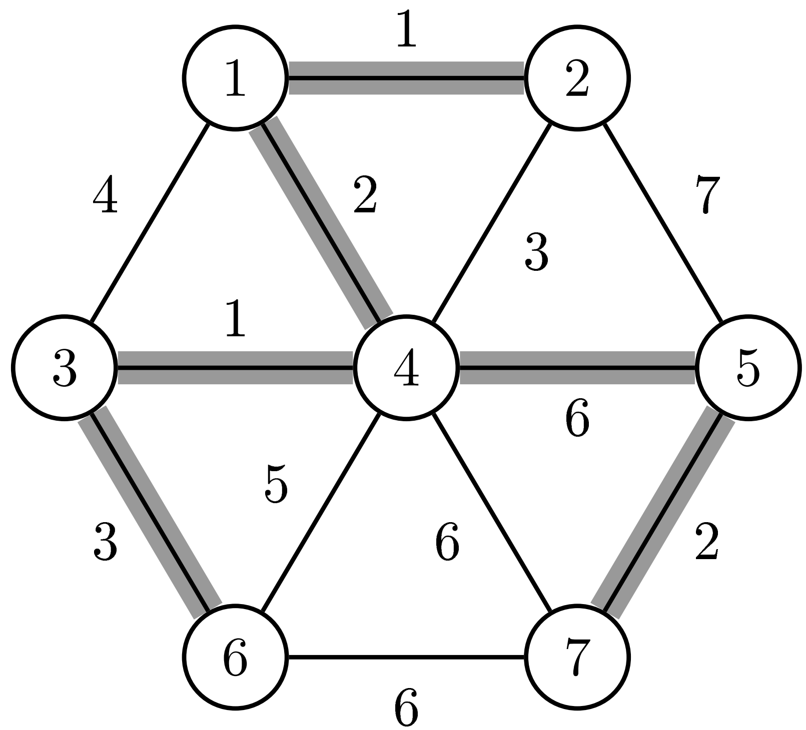

1
1
2
4
2
7
3
3
1
4
5
6
5
3
6
2
6
7
6


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


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


1
2
3
4
5
6
7
1
4
2
3
7
1
6
5
3
6
2
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: $O(|E| \log |V|)$ (υλοποίηση με Union-Find, Union by Rank)

Μέγιστη ροή (Max Flow)

Δοθέντος γράφου με βάρη που αντιπροσωπεύουν *χωρητικότητες* (network) και δύο κόμβων s, t, ζητείται να δρομολογηθεί όσο το δυνατόν μεγαλύτερη ροή από τον s στον t.

Θεώρημα. *(Max Flow – Min Cut) Η μέγιστη ροή ισούται με την ελάχιστη (ως προς χωρητικότητα) τομή (σύνολο ακμών) που διαχωρίζει τον s από τον t.*

Αλγόριθμος Ford-Fulkerson

Επιλογή μονοπατιού από τον s στον t. Δρομολόγηση ροής ίσης με την ελάχιστη χωρητικότητα ακμής στο μονοπάτι.

Επανάληψη της διαδικασίας στο παραμένον δίκτυο (residual network) ώσπου να μην υπάρχει πλέον μονοπάτι από τον s στον t.

Ορολογία: Τα μονοπάτια που χρησιμοποιεί ο αλγόριθμος λέγονται συνήθως μονοπάτια επαύξησης (augmenting paths).

Πολυπλοκότητα: $O(|f^*||E|)$, f^* η μέγιστη ροή.

Βελτιώσεις: Αλγόριθμος Edmonds-Karp $O(|V||E|^2)$ (shortest paths), αλγόριθμος Goldberg $O(|V|^2|E|)$ και $O(|V|^3)$ (preflow-push).

Τέλειο ταίριασμα (Perfect Matching)

Σε διμερείς γράφους:

Ανάγεται στο πρόβλημα της μέγιστης ροής.

Μονοπάτια επαύξησης: στην περίπτωση αυτή είναι ουσιαστικά μονοπάτια όπου εναλλάσσονται ακμές εκτός του τρέχοντος matching με ακμές εντός του τρέχοντος matching (alternating paths) και όπου η πρώτη και τελευταία ακμή είναι εκτός matching.

Πολυπλοκότητα: $O(|V||E|)$ (επειδή $|f^*| \leq |V|/2$). Βελτίωση Hopcroft-Karp: $O(|V|^{5/2})$.

Σχετικό πρόβλημα: Stable Marriage.

Πρόβλημα Stable Marriage/Matching

Δίνεται πλήρης διμερής γράφος, όπου οι κόμβοι κάθε συνόλου έχουν σειρά προτίμησης για τους κόμβους του άλλου συνόλου (ή απλά λίστες/πίνακας προτίμησης).

Ζητείται matching M όπου να μην υπάρχει ζευγάρι εκτός ταιριάσματος, όπου και οι δύο κόμβοι να επιθυμούν τον άλλο περισσότερο από το 'ταίρι' τους στο M.

Αλγόριθμος Gale-Shapley (1962): οι άντρες προτείνουν, οι γυναίκες αποδέχονται ή όχι (ή και αντίστροφα). Ευνοεί τους προτείνοντες.

Πολυπλοκότητα: $O(n^2)$.

Παραλλαγές (εκτός γάμου): ανάθεση ειδικευόμενων γιατρών σε νοσοκομεία (hospital/residents problem), επιλογή φοιτητών σε σχολές, Stable Roommates. Μερικές παραλλαγές είναι NP-complete.

Χρωματισμός ακμών

Σε γενικούς γράφους: αρκούν $\Delta + 1$ χρώματα (απλοί γράφοι) και $\Delta + \mu$ χρώματα (πολυγραφήματα) [Vizing, 1964].

Το ερώτημα εάν αρκούν Δ χρώματα είναι NP-complete.

Καλύτερος μέχρι στιγμής προσεγγιστικός αλγόριθμος για πολυγραφήματα: $(1 + \frac{3}{\sqrt{2OPT}})$-apx. (APTAS) [Sanders and Steurer, 2005]

Σε διμερείς γράφους (και πολυγραφήματα): αρκούν πάντοτε Δ χρώματα (König, 1916, αλγόριθμος $O(|E||V|)$).

Καλύτερος μέχρι στιγμής αλγόριθμος: $O(|E|\log\Delta)$ [Cole, Ost, and Schirra, 2001].

Χρωματισμός ακμών σε διμερείς γράφους

Θεώρημα. [König, 1916] Σε διμερείς γράφους (και πολυγραφήματα) αρκούν πάντοτε Δ χρώματα.

Απόδειξη (ιδέα): θεωρώντας maximal matching με Δ χρώματα, αν υποθέσουμε ότι υπάρχει αχρωμάτιστη ακμή μπορούμε να βρούμε augmenting path, φτάνοντας σε αντίφαση.

Από το θεώρημα προκύπτουν 2 αλγόριθμοι:

(α) Επαναληπτική εύρεση και αφαίρεση perfect matching (σε Δ-κανονικό γράφο): $O(\Delta n^{\frac{5}{2}})$.

(β) Επαναληπτικός χρωματισμός ακμών βασισμένος στην απόδειξη: $O(|V||E|) = O(\Delta n^2)$.

Παραμετρική πολυπλοκότητα

Fixed Parameter Tractability: ένα πρόβλημα λέγεται *fixed parameter tractable* ως προς κάποια παράμετρο k που δίνεται στην είσοδο αν υπάρχει αλγόριθμος που το λύνει, με πολυπλοκότητα:

$$O(f(k) \cdot poly(n))$$

όπου n το συνολικό μήκος της εισόδου.

Παράδειγμα: το Vertex Cover (decision) ανήκει στην κλάση **FPT** με παράμετρο k (το επιθυμητό μέγεθος του cover) γιατί λύνεται σε χρόνο $O(2^k poly(n))$ (άσκηση).

Ãλλα προβλήματα στην FPT: Sat wrt n, πολλά wrt treewidth (3-Coloring, Weighted Independent Set, Hamilton Cycle, ...)

΄Οχι στην FPT: Vertex Coloring wrt # colors (γιατί;)