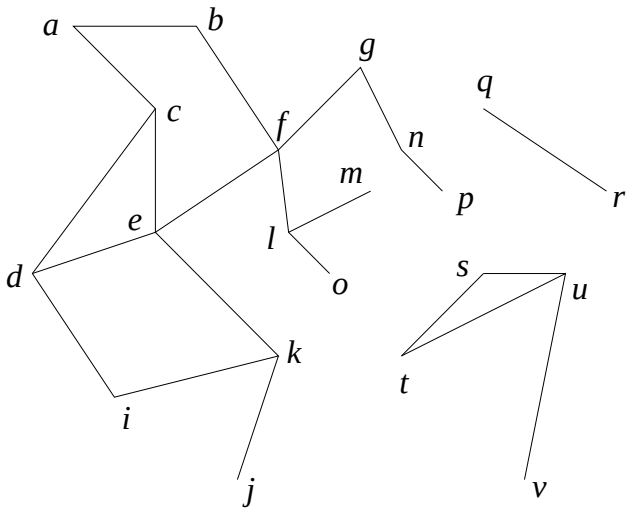
ΘΕΩΡΙΑ ΑΛΓΟΡΙΘΜΩΝ

Περιεχόμενα

1 Γραφήματα

Γράφημα G καλείται ένα ζευγάρι συνόλων (V, E) όπου $V \neq \emptyset$ ένα πεπερασμένο σύνολο και E ένα σύνολο από μη διατεταγμένα ζευγάρια στοιχείων του συνόλου V. Το V καλείται **σύνολο κορυφών** και το E **σύνολο ακμών** (Σχήμα 1).



Σχήμα 1: Ενα (μη-συνεκτικό) γράφημα $G = (V, E)$.

Αν σε ένα γράφημα $G = (V, E)$, το V είναι ένα μονοσύνολο και $E = \emptyset$, τότε το γράφημα καλείται **τετριμμένο**.
Οταν σε ένα γράφημα $G = (V, E)$, το σύνολο E αποτελείται από διατεταγμένα ζευγάρια στοιχείων του V, τότε το γράφημα καλείται **κατευθυνόμενο**.
Δύο σημεία $A, B \in V$ καλούνται **γειτονικά**, αν (A, B) είναι μιά ακμή του E.
Δύο γραφήματα καλούνται **ισομορφικά** αν υπάρχει μιά $1-1$ αμφιμονοσήμαντη αντιστοιχία μεταξύ των κορυφών τους που διατηρεί την γειτνίαση. Δύο ισομορφικά γραφήματα συνήθως ταυτίζονται.
Υπογράφημα ενός γραφήματος $G = (V, E)$ καλούμε ένα γράφημα $G_1 = (V_1, E_1)$, τέτοιο ώστε: $V_1 \subseteq V$ και $E_1 \subseteq E$.
Εστω ένα γράφημα $G = (V, E)$ και $V_1 \subseteq V$. Το **επαγόμενο υπογράφημα** $< G_1 >$ είναι το υπογράφημα με κορυφές το σύνολο V_1 και ακμές όλες τις ακμές του E που συνδέουν τα στοιχεία του V_1. Δηλαδή, το $< G_1 >$ είναι το μέγιστο υπογράφημα του G με κορυφές τα στοιχεία του V_1.

Καλούμε **δρόμο** ενός γραφήματος, με αρχή μιά κορυφή $v \in V$ και τέλος μιά κορυφή $w \in V$, μιά οποιαδήποτε ακολουθία από κορυφές με τις εξής ιδιότητες:
1. Η πρώτη κορυφή είναι η v.
2. Η τελευταία κορυφή είναι η w.
3. Δύο διαδοχικές κορυφές της ακολουθίας είναι γειτονικές.

Ενας τέτοιος δρόμος συμβολίζεται $v - w$. **Μήκος** ενός δρόμου καλείται το πλήθος των ακμών που υπάρχουν στον δρόμο.
Ενα γράφημα καλείται **συνεκτικό** αν δύο οποιεσδήποτε κορυφές του συνδέονται με ένα δρόμο.
Μονοπάτι καλείται ένας δρόμος, στον οποίο δεν υπάρχουν διπλές εμφανίσεις καποιας κορυφής.
Κύκλος καλείται ένα μονοπάτι $v - w$ του οποίου η αρχή και το τέλος (v και w αντίστοιχα) είναι γειτονικές κορυφές.
Δέντρο καλείται ένα γράφημα που είναι συνεκτικό και δεν περιέχει κύκλους. Ενα δέντρο καλείται **δυαδικό** αν κάθε κορυφή του έχει το πολύ τρείς προσπίπτουσες ακμές. Συνήθως σε κάθε δέντρο μία κορυφή του χαρακτηρίζεται σαν **ρίζα**. Στα δυαδικά δέντρα σαν ρίζα παίρνουμε μία κορυφή με το πολύ δύο προσπίπτουσες ακμές. Καλούμε **φύλλα** ενός δέντρου όλες τις κορυφές του, εκτός (ίσως) από τη ρίζα, που έχουν μόνο μιά προσπίπτουσα ακμή.
Αν από ένα γράφημα G αφαιρέσουμε μία κορυφή του v_i (αυτό σημαίνει ότι θα αφαιρεθούν και όλες οι ακμές που προσπίπτουν στην κορυφή v_i) προκύπτει το υπογράφημα $G - v_i$ του G. Αν πάλι αφαιρέσουμε μία ακμή του x_i (χωρίς να αφαιρεθούν οι κορυφές που συνδέει αυτή η ακμή) προκύπτει το υπογράφημα $G - x_i$. Μπορούμε επίσης να προσθέσουμε μιά ακμή x_j στο G, συνδέοντας δύο σημεία που δεν είναι γειτονικά, και να δημιουργήσομε ένα υπερ-γράφημα $G + x_j$. Ομως η πρόσθεση μιάς νέας κορυφής σ'ένα γράφημα G είναι ζήτημα που σχετίζεται με την πρόσθεση γραφημάτων (γιατί μιά κορυφή αποτελεί ένα τετριμμένο γράφημα) και θα αναφερθεί σε άλλη παράγραφο.
Ζώνη $g(G)$ ενός γραφήματος G είναι το μήκος του κοντινότερου κύκλου στο G.
Περιφέρεια $c(G)$ ενός γραφήματος G είναι το μήκος του μεγαλύτερου κύκλου στο G.
Απόσταση $d(v, w)$ δύο κορυφών $v, w \in G$ είναι το μήκος του μικρότερου μονοπατιού που τα ενώνει. Αν δύο σημεία δεν ενώνονται με κάποιο μονοπάτι, τότε ορίζουμε $d(v, w) = \infty$.

Σ'ένα συνεκτικό γράφημα η απόσταση είναι μία μετρική. Δηλαδή, $\forall v, w \in V$:

1. $d(v,w) \geq 0$ και $d(v,w) = 0 \iff v = w$
2. $d(v,w) = d(w,v)$
3. $d(v,w) + d(w,z) \geq d(v,z)$

Γεωδεσική δύο σημείων του G καλείται το κοντινότερο μονοπάτι ανάμεσα στα δύο σημεία.
Διάμετρος $d(G)$ ενός συνεκτικού γραφήματος G είναι το μήκος της μεγαλύτερης γεωδεσικής του γραφήματος.
Βαθμός μιάς κορυφής $v_i \in V$ (συμβολίζεται: $deg(v_i)$) είναι το πλήθος των ακμών που προσπίπτουν σ'αυτή την κορυφή.

Θεώρημα 1 *Εστω ένα γράφημα $G = (V, E)$. Ισχύει ότι:*

$$\sum_{\forall v_i \in V} deg(v_i) = 2|E|.$$

Θεώρημα 2 *Σε κάθε γράφημα $G = (V, E)$ ο αριθμός των κορυφών που έχουν περιττό βαθμό, είναι άρτιος.*

Απόδειξη: Εστω k οι κορυφές του G που έχουν περιττό βαθμό (τις συμβολίζουμε w_i) και l οι κορυφές με άρτιο βαθμό (τις συμβολίζουμε z_i). Τότε:

$$\sum_{i=1}^{k} deg(w_i) + \sum_{j=1}^{l} deg(z_i) = 2|E| \qquad \Longrightarrow$$

$$\sum_{i=1}^{k} deg(w_i) = 2|E| - \sum_{j=1}^{l} deg(z_i) = \text{άρτιος} - \text{άρτιος} = \text{άρτιος}$$

Ομως, μόνον η πρόσθεση αρτίου πλήθους περιττών αριθμών είναι άρτιος αριθμός. Αρα $k =$ άρτιος.

Αν σε ένα γράφημα G όλα τα σημεία έχουν τον ίδιο βαθμό r τότε το γράφημα καλείται **ομαλό** βαθμού r.
Ενα ομαλό γράφημα βαθμού 0 είναι ένα γράφημα που δεν περιέχει καθόλου ακμές ($E = \emptyset$). Τα ομαλά γραφήματα βαθμού 3 καλούνται κυβικά.

Πόρισμα 3 *Κάθε κυβικό γράφημα έχει άρτιο αριθμό σημείων.*

1.1 Το πρόβλημα του Ramsey

Αρχίζουμε μ'ένα πρόβλημα: *"Δείξτε ότι σε μιά παρέα 6 ατόμων θα συμβαίνει τουλάχιστον ένα από τα εξής δύο: θα υπάρχουν 3 γνωστοί μεταξύ τους ή 3 άγνωστοι"*.
Θεωρούμε τα 6 άτομα σαν 6 σημεία ενός γραφήματος και την γνωριμία (αν υπάρχει) σαν μιά ακμή που ενώνει τα σημεία που "γνωρίζονται". Πριν διατυπώσουμε το παραπάνω πρόβλημα με όρους της θεωρίας γραφημάτων, δίνουμε μερικούς ορισμούς:
Συμπλήρωμα $\overline{G}$ ενός γραφήματος G, καλείται το γράφημα που έχει το ίδιο σύνολο κορυφών με το G και επιπλέον, δύο σημεία είναι γειτονικά στο $\overline{G}$ τότε και μόνον τότε όταν δεν είναι στο G.
Ενα γράφημα p σημείων καλείται **πλήρες** αν όλα τα σημεία του ανά δύο είναι γειτονικά (συμβόλίζεται: K_p). Αρα ένα πλήρες γράφημα K_p περιέχει $p(p-1)/2$ ακμές και είναι ομαλό βαθμού $p-1$. Το K_3 καλείται τρίγωνο, το $\overline{K_p}$ είναι ομαλό βαθμού 0.
Με βάση του παραπάνω ορισμούς, διατυπώνουμε το αρχικό πρόβλημα ως εξής:

Θεώρημα 4 *Σε κάθε γράφημα G έξι σημείων θα ισχύει τουλάχιστον ένα από τα εξής δύο: το G ή το $\overline{G}$ θα περιέχει ένα τρίγωνο.*

Απόδειξη: Εστω $G = (V, E)$ ένα τυχόν γράφημα με έξι σημεία και $v \in V$. Επειδή το σημείο v είναι γειτονικό με τα άλλα πέντε σημεία είτε στο G είτε στο $\overline{G}$, υποθέτουμε χωρίς περιορισμό της γενικότητας ότι υπάρχουν 3 σημεία u_1, u_2, u_3 γειτονικά του v στο G. Αν οποιαδήποτε δύο από αυτά τα σημεία είναι γειτονικά στο G, τότε αυτά τα δύο σημεία μαζί με το v σχηματίζουν ένα τρίγωνο. Αν πάλι δεν υπάρχουν δύο σημεία, από αυτά τα τρία, που να είναι γειτονικά στο G, τότε τα u_1, u_2, u_3 σχηματίζουν ένα τρίγωνο στο $\overline{G}$.

Το παραπάνω θεώρημα γεννάει ένα ερώτημα, γνωστό ως πρόβλημα του Ramsey: Αν μας δώσουν δύο τυχόντες ακεραίους m, n, ποιός είναι ο μικρότερος ακέραιος $r(m, n)$ έτσι ώστε κάθε γράφημα με $r(m, n)$ σημεία να περιέχει τουλάχιστον ένα από τα γραφήματα K_m, $\overline{K}_n$.
Οι αριθμοί $r(m, n)$ καλούνται Αριθμοί του Ramsey και ο υπολογισμός τους για

τυχόντα m και n είναι ένα πρόβλημα που δεν έχει λυθεί. Εχει αποδειχθεί ότι:

$$r(m,n) \leq \begin{pmatrix} m+n-2 \\ m-1 \end{pmatrix}$$

Λήμμα 5 $r(m,n) = r(n,m)$

Απόδειξη: Πρέπει να δείξουμε ότι κάθε γράφημα $r(m,n)$ σημείων είναι γράφημα $r(n,m)$ σημείων και αντίστροφα. Δηλαδή με άλλα λόγια, κάθε γράφημα r σημείων θα περιέχει τουλάχιστον ένα από τα γραφήματα K_m, $\overline{K}_n$ και τουλάχιστον ένα από τα γραφήματα K_n, $\overline{K}_m$. Χωρίς περιρισμό της γενικότητας υποθέτουμε $m \leq n$. Αρα, αν κάποιο γράφημα r σημείων περιέχει το K_n θα περιέχει και το K_m. Ακόμα, αν περιέχει το $\overline{K}_n$ θα περιέχει και το $\overline{K}_m$. Εστω ότι περιέχει το K_m και δεν περιέχει και το K_n τότε θα δείξουμε ότι περιέχει το $\overline{K}_m$. Εστω G ένα τέτοιο γράφημα και $K_n \not\subseteq G \Rightarrow \overline{K}_n \not\subseteq \overline{G} \Rightarrow K_m \subseteq \overline{G}$[1] $\Rightarrow$ $\overline{K}_m \subseteq G$.
Με τον ίδιο τρόπο:
$(K_m \subseteq G$ και $\overline{K}_m \not\subseteq G) \Rightarrow (K_n \subseteq G)$
$(\overline{K}_m \subseteq G$ και $\overline{K}_n \not\subseteq G) \Rightarrow (K_m \subseteq G)$
$(\overline{K}_m \subseteq G$ και $K_m \not\subseteq G) \Rightarrow (\overline{K}_n \subseteq G)$

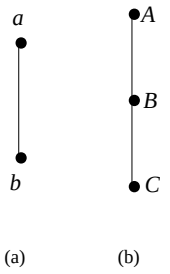
Ο υπολογισμός του $r(m,n)$ για δεδομένα m και n γίνεται με πρακτικό τρόπο (δοκιμές), π.χ. ο παρακάτω πίνακας.

$n \backslash m$	2	3	4	5	6	7
2	2	3	4	5	6	7
3	3	6	9	14	18	23
4	4	9	18			

[1] Γιατί το $\overline{G}$ είναι γράφημα r σημείων και θα περιέχει τουλάχιστον ένα από τα K_m, $\overline{K}_n$.

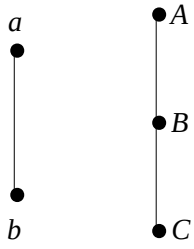
1.2 Πράξεις Γραφημάτων

Εστω δύο γραφήματα $G_1 = (V_1, E_1)$ ($|V_1| = p_1$, $|E_1| = q_1$) και $G_2 = (V_2, E_2)$ ($|V_2| = p_2$, $|E_2| = q_2$) (Σχήμα 2).



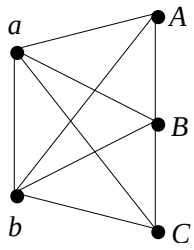
Σχήμα 2: Δύο γραφήματα (a) $G_1 = (V_1, E_1)$ και (b) $G_2 = (V_2, E_2)$.

1. **Ενωση:** $G_1 \cup G_2 = (V, E)$, με $V = V_1 \cup V_2$ και $E = E_1 \cup E_2$.
Πλήθος στοιχείων: $|V| = p_1 + p_2$ και $|E| = q_1 + q_2$.



Σχήμα 3: Η ένωση $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2$.

2. **Πρόσθεση:** $G_1 + G_2$ αποτελείται από το $G_1 \cup G_2$ και επί πλέον περιέχει όλες τις ακμές που συνδέουν κάθε σημείο του G_1 με όλα τα σημεία του G_2.
Πλήθος στοιχείων: $|V| = p_1 + p_2$ και $|E| = q_1 + q_2 + p_1 p_2$.
3. **Καρτεσιανό Γινόμενο:** $G_1 \times G_2 = (V, E)$, με $V = V_1 \times V_2$. Για να ορίσουμε το E θεωρούμε δύο σημεία $v = (v_1, v_2)$, $w = (w_1, w_2) \in V$. Τότε, τα v και w

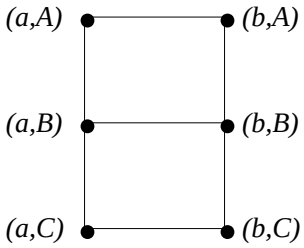


Σχήμα 4: Η πρόσθεση $G_1 + G_2$.

είναι γειτονικά στο $G_1 \times G_2$ αν και μόνον αν:

$[(v_1 = w_1)$ και $(v_2$ γειτονικό με $w_2)]$ ή $[(v_2 = w_2)$ και $(v_1$ γειτονικό με $w_1)]$.

Πλήθος στοιχείων: $|V| = p_1p_2$ και $|E| = p_1q_2 + p_2q_1$.

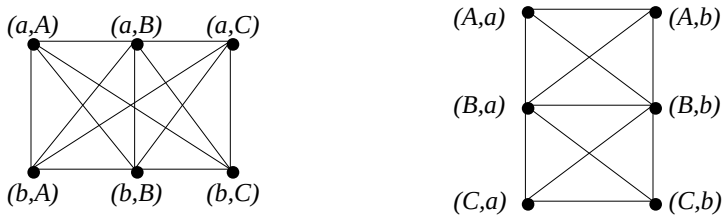


Σχήμα 5: Το καρτεσιανό γινόμενο $G_1 \times G_2$.

4. **Σύνθεση:** $G_1[G_2] = (V, E)$, με $V = V_1 \times V_2$. Εστω $v = (v_1, v_2)$, $w = (w_1, w_2)$ $\in V$. Τότε, v γειτονικό με w στο $G_1[G_2]$ αν και μόνον αν:

$$[v_1 \text{ γειτονικό με } w_1] \text{ ή } [(v_1 = w_1) \text{ και } (v_2 \text{ γειτονικό με } w_2)].$$

Προφανώς $G_1[G_2] \neq G_2[G_1]$.
Πλήθος στοιχείων: $|V| = p_1p_2$ και $|E| = p_1q_2 + p_2^2q_1$.

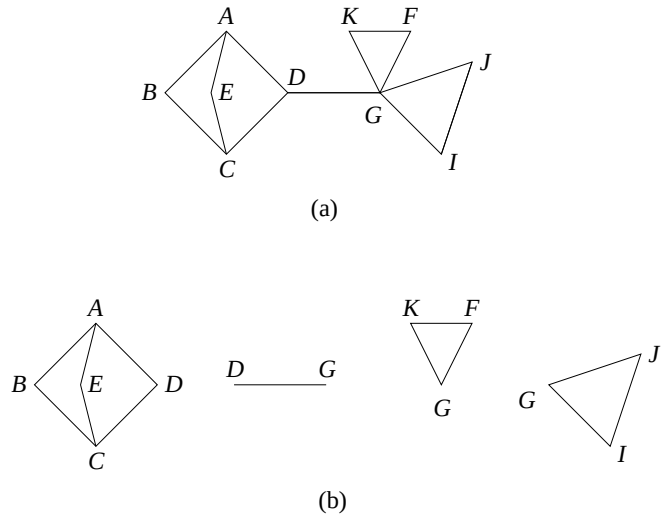


(a) (b)

Σχήμα 6: Οι συνθέσεις (a) $G_1[G_2]$ και (b) $G_2[G_1]$.

1.3 Δισυνεκτικές Συνιστώσες

Σημείο διαμέρισης ενός γραφήματος είναι το σημείο εκείνο που η αφαίρεσή του προκαλεί αύξηση του αριθμού των συνεκτικών υπογραφημάτων του γραφήματος. Μιά ευθεία με την ίδια ιδιότητα καλείται **γέφυρα**.



Σχήμα 7: (a) Ενα γράφημα και (b) οι δισυνεκτικές συνιστώσες του.

Αρα, αν v και x είναι αντίστοιχα σημείο διαμέρισης και γέφυρα ενός συνεκτικού γραφήματος G, τότε τα γραφήματα $G - v$ και $G - x$ είναι μη συνεκτικά.
Αν ένα γράφημα δεν έχει σημεία διαμέρισης τότε δεν έχει και γέφυρες. Το αντίθετο προφανώς δεν ισχύει.
Ενα γράφημα καλείται **μη-διαχωρίσιμο** αν είναι συνεκτικό, μη-τετριμμένο και δεν έχει σημεία διαμέρισης. **Δισυνεκτική συνιστώσα** ενός γραφήματος καλείται ένα μέγιστο μη-διαχωρίσιμο υπογράφημά του.
Αν ένα γράφημα $G = (V, E)$ είναι μη-διαχωρίσιμο τότε το ίδιο καλείται **δισυνεκτικό**.
Αποδεικνύεται ότι, κάθε δισυνεκτική συνιστώσα $G_i = (V_i, E_i)$ ενός γραφήματος G έχει την εξής ιδιότητα: *κάθε τριάδα κορυφών $v, w, u \in V_i$ θα ανήκει σ'ένα κύκλο.*

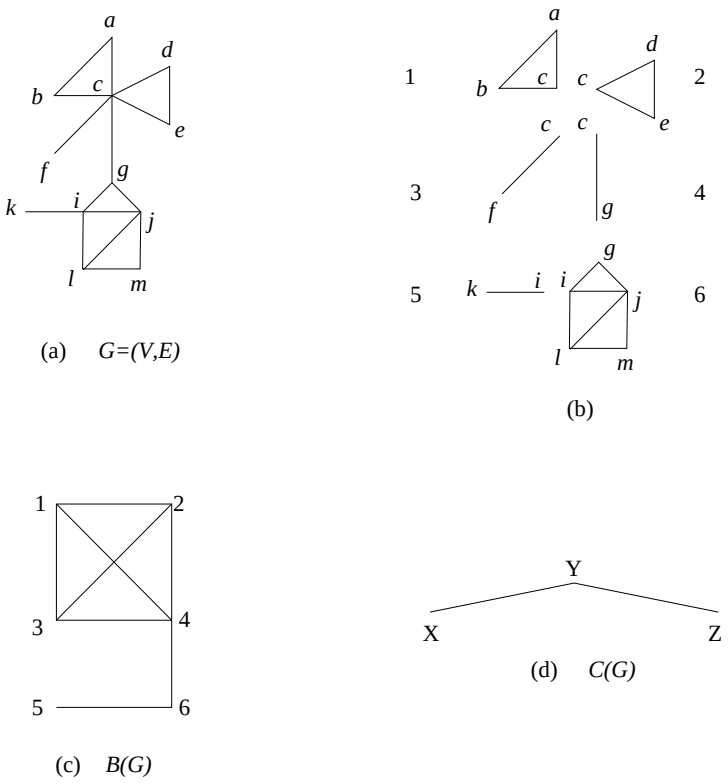
Θεώρημα 6 *Εστω v ένα σημείο ενός συνεκτικού γραφήματος $G = (V, E)$. Τα παρακάτω είναι ισοδύναμα:*
1. Το v είναι σημείο διαμέρισης του G.

2. Υπάρχουν σημεία $u, w \in V$ διαφορετικά του v έτσι ώστε το v να ανήκει σε κάθε μονοπάτι $u - w$.
3. Υπάρχει μιά διαμέριση του συνόλου $V - v$ σε υποσύνολα U και W έτσι ώστε: για κάθε $u \in U$ και για κάθε $w \in W$, το v να ανήκει σε κάθε μονοπάτι $u - w$.

Θεώρημα 7 *Εστω $x \in E$ μιά ακμή ενός συνεκτικού γραφήματος $G = (V, E)$. Τα παρακάτω είναι ισοδύναμα:*
1. Το x είναι μιά γέφυρα του G.
2. Το x δεν ανήκει σε κανένα κύκλο του G.
3. Υπάρχουν σημεία $u, w \in V$ έτσι ώστε το x να ανήκει σε κάθε μονοπάτι $u - w$.
4. Υπάρχει μιά διαμέριση του συνόλου V σε υποσύνολα U και W έτσι ώστε: για κάθε $u \in U$ και για κάθε $w \in W$, η ακμή x να ανήκει σε κάθε μονοπάτι $u - w$.

Θεώρημα 8 *Εστω $G = (V, E)$ ένα συνεκτικό γράφημα με τουλάχιστον 3 σημεία. Τα παρακάτω είναι ισοδύναμα:*
1. Το G είναι δισυνεκτικό.
2. Κάθε δύο σημεία του G ανήκουν σ'ένα κοινό κύκλο.
3. Κάθε σημείο και κάθε ακμή του G ανήκουν σ'ένα κοινό κύκλο.
4. Κάθε δύο ακμές του G ανήκουν σ'ένα κοινό κύκλο.
5. Για κάθε δύο σημεία και μία ακμή του G, υπάρχει μονοπάτι που ενώνει τα δύο σημεία και περιέχει την ακμή.
6. Για κάθε 3 σημεία του G, υπάρχει μονοπάτι που ενώνει ανά δύο τα σημεία και περιέχει το τρίτο σημείο.
7. Για κάθε 3 σημεία του G, υπάρχει μονοπάτι που ενώνει ανά δύο τα σημεία και δεν περιέχει το τρίτο σημείο.

Θεώρημα 9 *Κάθε μη-τετριμμένο συνεκτικό γράφημα έχει το λιγότερο δύο σημεία που δεν είναι σημεία διαμέρισης.*



Σχήμα 8: (a) Ενα συνεκτικό γράφημα $G = (V, E)$, (b) οι δισυνεκτικές συνιστώσες του 1,2,3,4,5 και 6, (c) το γράφημα $B(G)$ των δισυνεκτικών συνιστωσών του και (d) το γράφημα $C(G)$ των σημείων διαμέρισης του G: $X = 1 \cup 2 \cup 3 \cup 4$, $Y = 4 \cup 6$ και $Z = 5 \cup 6$.

Εστω $G_i = (V_i, E_i)$ οι δισυνεκτικές συνιστώσες (αν υπάρχουν) ενός γραφήματος $G = (V, E)$. Στη συνέχεια, ας αντιστοιχίσουμε τη κάθε δισυνεκτική συνιστώσα με ένα σημείο. Δύο σημεία $G_i = (V_i, E_i)$ και $G_j = (V_j, E_j)$ συνδέονται με μιά ακμή, αν υπάρχει ένα κοινό σημείο $v \in V_i \cap V_j$ (προφανώς, ένα τέτοιο σημείο v θα είναι σημείο διαμέρισης του G). Το γράφημα που προκύπτει συμβολίζεται $B(G)$ και καλείται **γράφημα των δισυνεκτικών συνιστωσών** του G.
Εξ'άλλου, ας θεωρήσουμε τις ενώσεις S_i των δισυνεκτικών συνιστωσών που περιέχουν ένα κοινό σημείο διαμέρισης v_i. Τότε, μπορούμε να αντιστοιχίσουμε σε κάθε S_i ένα σημείο. Δυό τέτοια σημεία S_i και S_j γίνονται γειτονικά τότε και μόνον τότε αν $S_i \neq S_j$ και $S_i \cap S_j \neq \emptyset$. Με τον τρόπο αυτό προκύπτει ένα γράφημα που συμβολίζεται $C(G)$ και καλείται **γράφημα των σημείων διαμέρισης** του G (Βλέπε σχήμα 8).

1.4 Δέντρα

Δέντρο καλείται ένα μη-κυκλικό και συνεκτικό γράφημα. **Δάσος** καλείται ένα γράφημα μη-κυκλικό και μη-συνεκτικό. Ενα δάσος αποτελείται από ένα σύνολο δέντρων που δεν συνδέονται μεταξύ τους με ακμές.

Θεώρημα 10 *Σε ένα γράφημα $G = (V, E)$ τα ακόλουθα είναι ισοδύναμα.*
1. Το G είναι δέντρο.
2. Κάθε δύο κορυφές του G συνδέονται μ'ένα μοναδικό μονοπάτι.
3. Το G είναι συνεκτικό και $|V| = |E| + 1$.
4. Το G είναι μη κυκλικό και $|V| = |E| + 1$.
5. Το G είναι μη κυκλικό και αν δύο οποιαδήποτε μη γειτονικά σημεία του ενωθούν με μία ακμή x, τότε το $G + x$ έχει ακριβώς ένα κύκλο.
6. Το G είναι συνεκτικό, δεν είναι το $K_{|V|}$ για $|V| \geq 3$ και αν δύο οποιαδήποτε μη γειτονικά σημεία του ενωθούν με μία ακμή x, τότε το $G + x$ έχει ακριβώς ένα κύκλο.
7. Το G δεν είναι το $K_3 \cup K_1$ ή το $K_3 \cup K_2$, ισχύει η σχέση $|V| = |E| + 1$ και αν δύο οποιαδήποτε μη γειτονικά σημεία του ενωθούν με μία ακμή x, τότε το $G + x$ έχει ακριβώς ένα κύκλο.

2 Ασκήσεις

1. Να κατασκευάσετε ένα κυβικό γράφημα με $2n$ $(n \geq 3)$ σημεία που δεν έχει τρίγωνα.

2. Αν το G δεν είναι συνεκτικό τότε το $\overline{G}$ είναι. Το αντίθετο ισχύει;

3. Εστω $G = (V, E)$ ένα γράφημα του οποίου όλα τα σημεία έχουν βαθμό k ή $k+1$. Αν το G έχει p_k σημεία βαθμού k και p_{k+1} σημεία βαθμού $k+1$, τότε $p_k = (k+1)|V| - 2|E|$.

4. Να βρεθεί ο μέγιστος αριθμός ακμών σ'ένα γράφημα p σημείων και χωρίς άρτιους κύκλους.

5. Κάθε γράφημα με διάμετρο d και ζώνη $d+1$ είναι ομαλό.

6. Δείξτε αν ισχύει: Αν G_1 και G_2 είναι ομαλά γραφήματα, τότε θα είναι ομαλά και τα γραφήματα:
   (α) $G_1 + G_2$
   (β) $G_1 \times G_2$
   (γ) $G_1[G_2]$.

7. Δείξτε αν ισχύει:
   (α) $\overline{G_1 + G_2} = \overline{G}_1 + \overline{G}_2$
   (β) $\overline{G_1 \times G_2} = \overline{G}_1 \times \overline{G}_2$
   (γ) $\overline{G_1[G_2]} = \overline{G}_1[\overline{G}_2]$.

8. Υπολογίστε τον αριθμό των κύκλων στο γράφημα K_p.

9. Εστω δύο γραφήματα G_1 και G_2 με p σημεία v_i και w_i αντίστοιχα ($p \geq 3$). Αν για κάθε i τα υπογραφήματα $G - v_i$ και $G - w_i$ είναι ισόμορφα, τότε και τα G_1, G_2 είναι ισόμορφα.

10. Ποιός είναι ο μέγιστος αριθμός σημείων διαμέρισης σ'ένα γράφημα p σημείων;

11. Ενα κυβικό γράφημα έχει ένα σημείο διαμέρισης τότε και μόνον τότε αν έχει μιά γέφυρα.

12. Ο μικρότερος αριθμός σημείων σ'ένα κυβικό γράφημα με μιά γέφυρα είναι 10.

13. Δείξτε ότι αν v είναι σημείο διαμέρισης ενός γραφήματος G, τότε το v δεν είναι σημείο διαμέρισης του $\overline{G}$. Το αντίθετο ισχύει;

14. Δείξτε αν ισχύει: ένα συνεκτικό γράφημα G με $p \geq 3$ σημεία είναι δισυνεκτικό αν και μόνον αν για κάθε δύο σημεία και μιά ακμή του υπάρχει ένα μονοπάτι που ενώνει τα σημεία και δεν περιέχει την ακμή.

15. Δείξτε ότι ένα συνεκτικό γράφημα με το λιγότερο τρεις ακμές είναι δισυνεκτικό αν και μόνον αν κάθε δύο γειτονικές ακμές του ανήκουν σ'ένα κύκλο.

16. Δείξτε ότι τα ακόλουθα είναι ισοδύναμα:
(α) Το γράφημα G είναι μη κυκλικό γράφημα.
(β) Κάθε ακμή του G είναι γέφυρα.
(γ) Κάθε δισυνεκτική συνιστώσα του G είναι K_2.
(δ) Κάθε μη-κενή τομή δύο συνεκτικών υπογραφημάτων του G είναι συνεκτικό υπογράφημα.

17. Τα ακόλουθα είναι ισοδύναμα:

(α) Το γράφημα $G = (V, E)$ έχει μόνο ένα κύκλο και είναι συνεκτικό.

(β) Το γράφημα $G = (V, E)$ είναι συνεκτικό και $|V| = |E|$.

(γ) Υπάρχει ακμή $x \in E$ έτσι ώστε το $G - x$ να είναι δέντρο.

(δ) Το γράφημα G είναι συνεκτικό και το σύνολο των ακμών του G που δεν είναι γέφυρες σχηματίζουν ένα κύκλο.

3 Αλγόριθμοι σε γραφήματα

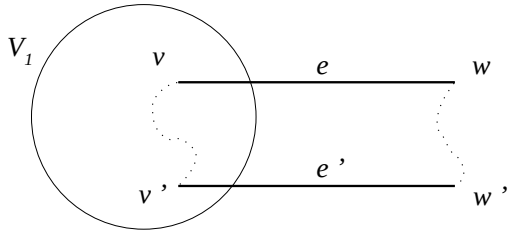
3.1 Παράγοντα δέντρα ελαχίστου κόστους

Ενα γράφημα καλείται **διατιμημένο** αν έχουμε αντιστοιχίσει ένα πραγματικό αριθμό σε κάθε ακμή του. Δηλαδή, αν υπάρχει μιά συνάρτηση $c : E \longrightarrow \mathcal{R}$. Καλούμε **κόστος** της ακμής e, την τιμή $c(e)$. Επίσης, κόστος του γραφήματος G καλούμε το άθροισμα του κόστους όλων των ακμών του.
Καλούμε **παράγον δέντρο** ενός γραφήματος $G = (V, E)$, ένα δέντρο $S = (V, T)$, με $T \subseteq E$ (δηλαδή, το σύνολο των κορυφών του S είναι ολόκληρο το V και οι ακμές του S είναι υποσύνολο του E).
Παράγον δέντρο **ελαχίστου κόστους** ενός διατιμημένου συνεκτικού γραφήματος G, καλούμε ένα παράγον δέντρο με το ελάχιστο δυνατό κόστος. Ασφαλώς, σε ένα γράφημα μπορεί να υπάρχουν περισσότερα από ένα παράγοντα δέντρα ελαχίστου κόστους.
Στη συνέχεια θα εξετάσουμε δύο αλγορίθμους για την κατασκευή ενός παράγοντος δέντρου ελαχίστου κόστους. Και οι δύο αλγόριθμοι στηρίζονται στο παρακάτω θεώρημα:

Θεώρημα 11 *Εστω $G = (V, E)$ ένα μη-κατευθυνόμενο συνεκτικό γράφημα με μιά συνάρτηση κόστους c στις ακμές του. Εστω ακόμα $\{(V_1, T_1), \cdots, (V_k, T_k)\}$ ένα παράγον δάσος του G, με $k > 1$. Εστω $T = \cup_{i=1}^{k} T_i$ και $e = (v, w)$ μιά ακμή ελαχίστου κόστους στο σύνολο $E - T$ έτσι ώστε $v \in V_1$ και $w \notin V_1$. Τότε υπάρχει ένα παράγον δέντρο του G που περιέχει τις ακμές του συνόλου $T \cup \{e\}$ και το οποίο έχει κόστος μικρότερο ή ίσο από το κόστος κάθε παράγοντος δέντρου του G που περιέχει τις ακμές του T.*

Απόδειξη: Ας υποθέσουμε ότι δεν υπάρχει ένα τέτοιο ελάχιστο παράγον δέντρο. Εστω $S' = (V, T')$ ένα παράγον δέντρο του G τέτοιο ώστε: $T \subseteq T'$, $e \notin T'$ και το οποίο έχει κόστος μικρότερο απ'όλα τα παράγοντα δέντρα του G που περιέχουν τις ακμές $T \cup e$. Αν προσθέσουμε την e στο δέντρο S' θα προκύψει ένας κύκλος, όπως στο Σχήμα 9. Ο κύκλος αυτός θα πρέπει να περιέχει μιά ακμή $e' = (v', w')$ με $v' \in V_1$ και $w' \notin V_1$, άρα $e' \notin T$. Επειδή e είναι ακμή ελαχίστου κόστους στο σύνολο $E - T$, συμπεραίνουμε ότι $c(e) \leq c(e')$. Αρα, αν από το S' αφαιρέσουμε την e' και προσθέσουμε την e προκύπτει ένα νέο παράγον δέντρο S του G που

έχει κόστος μικρότερο ή ίσο από το κόστους του δέντρου S'. Επειδή το S περιέχει το $T \cup \{e\}$, το συμπέρασμα απορρίπτει την αρχική μας υπόθεση.



Σχήμα 9:

3.2 Δύο αλγόριθμοι κατασκευής παράγοντος δέντρου ελαχίστου κόστους

Είσοδος: Ενα συνεκτικό, διατιμημένο και μη-κατευθυνόμενο γράφημα $G = (V, E)$.
Εξοδος: Ενα παράγον δέντρο ελαχίστου κόστους $S = (V, T)$ $(T \subseteq E)$ του G.

3.2.1 Ο αλγόριθμος του Kruskal

Βήμα 1: Κατασκευή μιάς λίστας Q που περιέχει τις ακμές του συνόλου E διατεταγμένες σε αύξουσα σειρά σύμφωνα με το κόστος τους.
Βήμα 2: Κατασκευή ενός συνόλου VS. Είναι το σύνολο των μονοσυνόλων των στοιχείων του V: $\forall v \in V \Longrightarrow \{v\} \in VS$.
Βήμα 3: Διάλεξε την πρώτη ακμή, έστω $e = (v, w)$, της λίστας Q.
Βήμα 4: Αν η ακμή e δημιουργεί κύκλο στο δάσος S, απορρίπτεται. Αν όχι, προστίθεται στο σύνολο T. Το βήμα αυτό υλοποιείται με την εκτέλεση των εντολών FIND και UNION στο σύνολο VS.

$A \longleftarrow$FIND(v)
$B \longleftarrow$FIND(w)
if $A \neq B$ **then** UNION(A,B,C) /* Τα ύψη των δέντρων με ρίζες A και B
$T \longleftarrow T \cup \{e\}$ στο VS, καθορίζουν αν $C = A$ ή $C = B$ */

Βήμα 5: Διέγραψε την ακμή $e = (v, w)$ από την λίστα Q.
Βήμα 6: Αν $|VS| > 1$ επανέλαβε το Βήμα 3.

Θεώρημα 12 *Σε ένα συνεκτικό γράφημα $G = (V, E)$, ο αλγόριθμος του Kruskal εκτελείται σε χρόνο $O(|E|log|E|)$.*

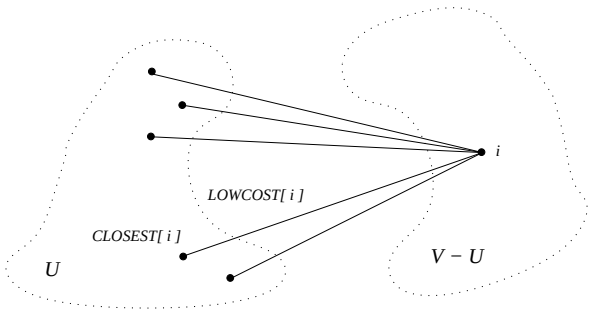
3.2.2 Ο αλγόριθμος του Prim

Θα χρησιμοποιηθούν τρία arrays:

1. Εστω C ένα $|V| \times |V|$ array, έτσι ώστε η τιμή $C[i,j]$ να ισούται με το κόστος της ακμής $(i,j) \in E$. Αν η ακμή (i,j) δεν υπάρχει στο E τότε θέτουμε $C[i,j] = \infty$.

2. Εστω U και $V - U$ μιά διαμέριση του συνόλου V, τότε:
   το array $CLOSEST[i] \in U$ συμβολίζει το σημείο του συνόλου U για το οποίο η ακμή $(i, CLOSEST[i])$ $(i \in V - U)$ έχει το ελάχιστο κόστος ανάμεσα στις ακμές (i,k) $(\forall k \in U)$.

3. Το array $LOWCOST[i]$ συμβολίζει το κόστος της ακμής $(i, CLOSEST[i])$.



Σχήμα 10:

Βήμα 1: Διάλεξε μιά τυχαία κορυφή, έστω $v \in V$ και δημιούργησε τα σύνολα $U = \{v\}$ και $V - U$.
Βήμα 2: Θέσε $CLOSEST[i] = v$, $(\forall i \in V - U)$.
Βήμα 3: Υπολόγισε το ελαχιστο κόστος:

$$LOWCOST[k] \longleftarrow \text{MIN}(\{LOWCOST[i] \ : \ \forall i \in V - U\})$$

Σ' αυτό το βήμα, εντοπίζεται η ακμή $(k, CLOSEST[k])$ με το ελάχιστο κόστος.
Βήμα 4: Πρόσθεσε την κορυφή k στο σύνολο U (και ενημέρωσε αντίστοιχα το σύνολο $V - U$).

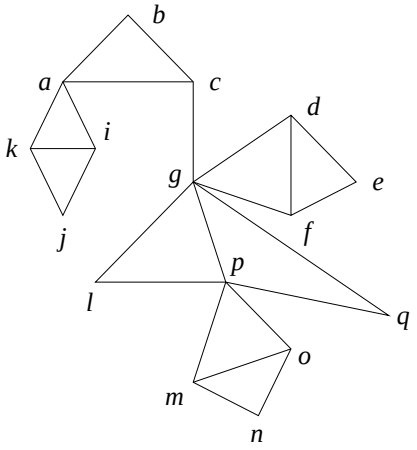
Βήμα 5: Πρόσθεσε την ακμή $(k, CLOSEST[k])$ στο σύνολο T.
Βήμα 6:

if $V - U \neq \emptyset$ **then**
  for κάθε $i \in V - U$ **do**
    if $LOWCOST[i] > C[i,k]$ **then** $CLOSEST[i] \longleftarrow k$
  επανέλαβε το **Βήμα 3**
else ΤΕΛΟΣ

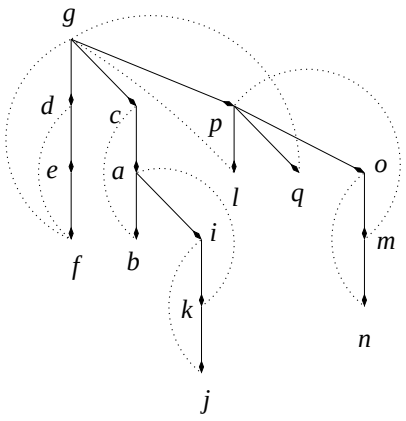
Θεώρημα 13 *Ο αλγόριθμος του Prim εκτελείται σε χρόνο* $O(|V|^2)$.

3.3 Depth-First Search

Είσοδος: Ενα γράφημα $G = (V, E)$.
Εξοδος: Ενα depth-first παράγον δάσος $D = (V, T)$, $(T \subseteq E)$.
Το σύνολο E των ακμών του G διαμερίζεται σε δυό υποσύνολα T και B: T είναι το σύνολο των ακμών του depth-first παραγόμενου δάσους και $B = E - T$.



(a) *G=(V,E)*



(b) *D=(V,T)*

Σχήμα 11: (a) Ενα συνεκτικό γράφημα $G = (V, E)$ και (b) το DFS παράγον δάσος $D = (V, T)$, $(T \subseteq E)$. Οι διακεκομμένες ακμές ανήκουν στο σύνολο $B \subseteq E$.

Αλγόριθμος DFS
begin
Για κάθε κορυφή $v \in V$ δημιουργείται μιά λίστα $L(v)$ που περιέχει τις γειτονικές της κορυφές.
$T \longleftarrow \emptyset$
for κάθε $v \in V$ **do** χαρακτήρισε v *"νέα"*
while $\exists$ κορυφή $v \in V$ χαρακτηρισμένη *"νέα"* **do** SEARCH(v)
end

SEARCH(v)
begin
χαρακτήρισε v *"παλιά"*
for κάθε $w \in L(w)$ **do**
if w είναι *"νέα"* **then**
begin
$T \longleftarrow T \cup \{(v, w)\}$
SEARCH(w)
end
end

Θεώρημα 14 *Η κατασκευή ενός DFS δάσους απαιτεί χρόνο $O(|V| + |E|)$ στη χειρότερη περίπτωση.*

Στο παρακάτω Λήμμα αποδεικνύεται μιά βασική ιδιότητα του depth-first παραγόμενου δάσους.

Λήμμα 15 *Αν (v, w) είναι μιά ακμή του συνόλου $B = E - T$ τότε στο depth-first παραγόμενο δάσος θα ισχύει ένα από τα παρακάτω: το v είναι πρόγονος του w ή το w είναι πρόγονος του v.*

Αποδειξη: Χωρίς περιορισμό της γενικότητας υποθέτουμε ότι ο Αλγόριθμος DFS συναντά την κορυφή v πριν από την κορυφή w (δηλαδή, η SEARCH(v) καλείται πριν την SEARCH(w)). Αρα, όταν η κορυφή v γίνεται *"παλιά* η κορυφή

w είναι ακόμα *"νέα"*. Επειδή $w \in L(v)$, καταλήγουμε ότι η κορυφή w θα γίνει απόγονος της v στο depth-first παραγόμενο δάσος.

3.4 Δισυνεκτικότητα

Ενα συνεκτικό, μη-κατευθυνόμενο γράφημα $G = (V, E)$ καλείται **δισυνεκτικό** αν και μόνον αν δεν περιέχει σημεία διαμέρισης.
Σ'ένα συνεκτικό, μη-κατευθυνόμενο γράφημα $G = (V, E)$ ορίζουμε μιά σχέση ισοδυναμίας $\mathcal{R}$ στο σύνολο των ακμών του: έστω δύο ακμές $e_1, e_2 \in E$:

$$e_1 \mathcal{R} e_2 \Longleftrightarrow \begin{cases} e_1 = e_2 \\ \quad \text{ή} \\ \text{υπάρχει κύκλος που περιέχει τις ακμές } e_1 \text{ και } e_2. \end{cases}$$

Είναι εύκολο να δειχτεί ότι η σχέση $\mathcal{R}$ διαμερίζει το σύνολο E σε κλάσεις ισοδυναμίας $E_1, \cdots, E_k$. Εστω V_i το σύνολο των κορυφών που αντιστοιχούν στις ακμές του συνόλου E_i $(1 \leq i \leq k)$. Κάθε γράφημα $G_i = (V_i, E_i)$ καλείται **δισυνεκτική συνιστώσα** του G.

Λήμμα 16 *Εστω $G_i = (V_i, E_i)$ $(1 \leq i \leq k)$ οι δισυνεκτικές συνιστώσες ενός συνεκτικού μη κατευθυνόμενου γραφήματος $G = (V, E)$. Τότε ισχύει:*
1. Κάθε γράφημα G_i $(1 \leq i \leq k)$ δεν περιέχει σημεία διαμέρισης (είναι δισυνεκτικό γράφημα).
2. Για κάθε $i \neq j$, η τομή $V_i \cap V_j$ περιέχει το πολύ μιά κορυφή.
3. Μιά κορυφή $v \in V$ είναι σημείο διαμέρισης του G αν και μόνον αν $v \in V_i \cap V_j$, για κάποια $i \neq j$.

Λήμμα 17 *Εστω $G = (V, E)$ ένα συνεκτικό, μη-κατευθυνόμενο γράφημα και έστω $D = (V, T)$ ένα depth-first παραγόμενο δέντρο του G. Μιά κορυφή $v \in V$ είναι σημείο διαμέρισης του G αν και μόνον αν ισχύει ένα από τα παρακάτω:*
1. Η κορυφή v είναι η ρίζα του δέντρου S και επιπλέον, v έχει περισσότερα από ένα παιδιά.
2. Η κορυφή v δεν είναι η ρίζα του δέντρου S και v έχει τουλάχιστον ένα παιδί t με την ακόλουθη ιδιότητα: δεν υπάρχει ακμή του συνόλου B $(B = E - T)$ που να συνδέει τους απογόνους του σημείου t (ή το ίδιο το t) με κάποιο πρόγονο του v.

Ο παρακάτω αλγόριθμος βασίζεται στο Λήμμα 17 και υπολογίζει τα σημεία διαμέρισης ενός συνεκτικού μη-κατευθυνόμενου γραφήματος $G = (V, E)$ σε χρόνο $O(|V| + |E|)$.

Βήμα 1: Εφάρμοσε depth-first-search στο γράφημα G. Στο βήμα αυτό θα συμβούν τα εξής:

1. Παράγεται το DFS δέντρο $D = (V, T)$ (αν το γράφημα G είναι μη-συνεκτικό τότε δημιουργείται ένα DFS δάσος).

2. Η κάθε μιά από τις n κορυφές του συνόλου V αποκτά ένα αριθμό-ετικέττα σύμφωνα με την σειρά που θα τις συναντήσει ο αλγόριθμος DFS (οι αριθμοί αυτοί θα είναι από το 1 μέχρι το n).
   Στο εξής, η κάθε κορυφή θα καλείται με την ετικέττα της.

3. Οι n κορυφές εισάγονται σε μιά διπλά διασυνδεδεμένη λίστα $\mathcal{L}$ σε αύξουσα σειρά με βάση τις ετικέττες τους.

4. Το σύνολο E διαμερίζεται στα υποσύνολα T και B. Κάθε ακμή αποκτά μιά ετικέττα που δηλώνει το υποσύνολο στο οποίο ανήκει.

Βήμα 2: **for** $s = n$ **to** 1 **do** διάλεξε την κορυφή s από την λίστα $\mathcal{L}$ και υπολόγισε τον χαρακτηριστικό αριθμό $LOW[s]$:

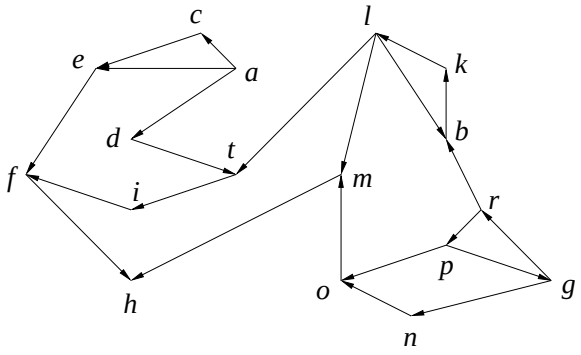
$LOW[s] = \text{MIN}(\{s\} \cup \{LOW[p] \,/\, p \text{ είναι παιδί της } s \text{ στο } D\} \cup \{w \,/\, (s, w) \in B\})$

Βήμα 3: Εντόπισε τα σημεία διαμέρισης στο γράφημα G (τα σ. δ. εισάγονται σε μιά λίστα $\mathcal{A}$):

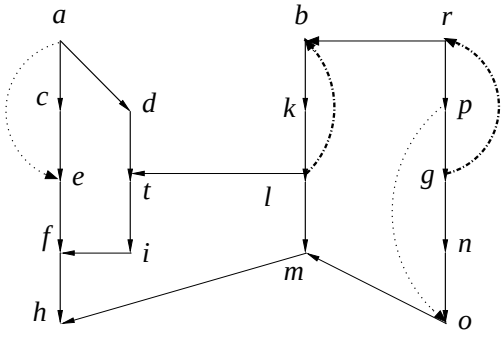
$\mathcal{A} \longleftarrow \emptyset$
διάλεξε την κορυφή 1 από την λίστα $\mathcal{L}$ (1 είναι η ρίζα του δέντρου D).
if 1 έχει τουλάχιστον δύο παιδιά στο δέντρο D **then** $\mathcal{A} \longleftarrow \mathcal{A} \cup \{1\}$
for $s = 2$ **to** n **do**
  διάλεξε την κορυφή s από την λίστα $\mathcal{L}$.
  if s έχει κάποιο παιδί w στο D, με $LOW[w] \geq s$ **then** $\mathcal{A} \longleftarrow \mathcal{A} \cup \{s\}$

3.5 Depth-First Search σε κατευθυνόμενα γραφήματα

Είσοδος: Ενα κατευθυνόμενο γράφημα $G = (V, E)$.
Εξοδος: Ενα DFS παραγόμενο δάσος $D = (V, T)$. Το σύνολο E διαμερίζεται σε τέσσερα υποσύνολα T, F, B και C.



(a)



(b)

T = { ac, ce, ef, fh, ad, dt, ti, bk, kl, lm, rp, pg, gn, no }

F = { ae, po }

B = { lb, gr }

C = { lt, if, mh, rb, om }

Σχήμα 12: (a) Ενα κατευθυνόμενο γράφημα $G = (V, E)$ και (b) το DFS παράγον δάσος $D = (V, T)$,. Το σύνολο E διαμερίζεται στα σύνολα T, F, B και C.

T: Το σύνολο των ακμών του DFS παραγόμενου δάσους $D = (V, T)$.

F: Οι ακμές που συνδέουν προγόνους με απογόνους του δάσους D, αλλά δεν ανήκουν στο σύνολο T.

B: Οι ακμές που συνδέουν απογόνους με προγόνους του δάσους D.

C: Οι ακμές που συνδέουν δύο κορυφές που η μία δεν είναι ούτε πρόγονος ούτε απόγονος της άλλης.

Ο Αλγόριθμος DFS σε κατευθυνόμενα γραφήματα είναι ίδιος με τον Αλγόριθμο DFS της Σελ. 27 με τη μοναδική διαφορά ότι για κάθε κορυφή $v \in V$ δημιουργούμε την λίστα

$$L[v] = \{w \in V \ : \text{η ακμή } (v, w) \in E \text{ έχει φορά από τη κορυφή } v \text{ στη } w\}.$$

Λήμμα 18 *Αν (v, w) είναι μιά ακμή του συνόλου C, τότε στο DFS παραγόμενο δάσος η ακμή (v, w) θα έχει φορά από τα δεξιά στα αριστερά.*

Θεώρημα 19 *Από ένα κατευθυνόμενο γράφημα $G = (V, E)$, κατασκευάζεται ένα DFS δάσος σε χρόνο $O(|V| + |E|)$, στη χειρότερη περίπτωση.*

3.6 Ισχυρή Συνεκτικότητα

Εστω $G = (V, E)$ ένα κατευθυνόμενο γράφημα. Μπορούμε να διαμερίσουμε το σύνολο V σε κλάσεις ισοδυναμίας V_i, έτσι ώστε δύο κορυφές v και w να είναι ισοδύναμες αν και μόνον αν υπάρχει μονοπάτι από την v στην w και μονοπάτι από την w στην v. Εστω E_i το σύνολο των ακμών που συνδέουν ζευγάρια κορυφών της κλάσης V_i. Το γράφημα $G_i = (V_i, E_i)$ καλείται **ισχυρά συνεκτική συνιστώσα** του γραφήματος G.
Κάθε κορυφή του V θα ανήκει σε κάποια κλάση V_i. Ομως, είναι δυνατόν να υπάρχει ακμή του E που να μην ανήκει σε κανένα σύνολο E_i.
Ενα γράφημα G καλείται **ισχυρά συνεκτικό** αν και μόνον αν περιέχει ακριβώς μιά ισχυρά συνεκτική συνιστώσα.
Στη συνέχεια, δίνονται δύο αλγόριθμοι για την εύρεση των ισχυρά συνεκτικών συνιστωσών ενός κατευθυνόμενου γραφήματος $G = (V, E)$.

Αλγόριθμος 1

1. Εφαρμόζουμε τον αλγόριθμο DFS στο γράφημα $G = (V, E)$. Κατα την διάρκεια της κατασκευής του DFS δάσους, αποθηκεύουμε σε κάθε κορυφή $w \in V$ δύο ακέραιους θετικούς αριθμούς a_1^w/a_2^w: a_1^w είναι ο αριθμός του βήματος που εισάγει την κορυφή w στο DFS δάσος και a_2^w είναι ο αριθμός του βήματος που τελειώνει την εξέταση της κορυφής w.

2. Κατασκεύασε το γράφημα $G^T = (V, E^T)$, όπου $E^T = \{(v, w) : (w, v) \in E\}$.

3. Εκτέλεσε τον αλγόριθμο DFS στο γράφημα $G^T = (V, E^T)$. Σ'αυτό το βήμα, οι κορυφές $w \in V$ εισάγονται στο DFS δάσος σε φθίνουσα ακολουθία με βάση τη τιμή a_2^w.

4. Οι κορυφές κάθε ξεχωριστού δέντρου στο DFS δάσος (και οι μεταξύ τους ακμές από το σύνολο E), αποτελούν μιά ισχυρά συνεκτική συνιστώσα.

Αλγόριθμος 2
begin
  $COUNT \longleftarrow 1$
  for κάθε $v \in V$ **do** χαρακτήρισε v *"νέα"*
  $STACK \longleftarrow \emptyset$
  while $\exists$ κορυφή $v \in V$ χαρακτηρισμένη *"νέα"* **do** SEARCHC(v)
end

procedure SEARCHC(v)
begin
  χαρακτήρισε v *"παλιά"*
  $DFNUMBER[v] \longleftarrow COUNT$
  $COUNT \longleftarrow COUNT + 1$
  $LOWLINK[v] \longleftarrow DFNUMBER[v]$
  push v στο $STACK$
  for κάθε $w \in L[v]$ **do**
    if w είναι *"νέα"* **then**
      SEARCHC(w)
      $LOWLINK[v] \longleftarrow$ MIN($LOWLINK[v], LOWLINK[w]$)
    else if $DFNUMBER[w] < DFNUMBER[v]$ **and** $w \in STACK$ **then**
      $LOWLINK[v] \longleftarrow$ MIN($DFNUMBER[w], LOWLINK[v]$
  if $LOWLINK[v] = DFNUMBER[v]$ **then**
      repeat
          pop x από το $STACK$
          print x
      until $x = v$
      print *"Τέλος εκτύπωσης μιάς ισχυράς συνεκτικής συνιστώσας"*
end

3.7 Τρεις αλγόριθμοι για την εύρεση ελαχίστων μονοπατιών από μιά κορυφή-πηγή $v_0 \in V$ σε κατευθυνόμενα διατιμημένα γραφήματα $G = (V, E)$

3.7.1 Ο αλγόριθμος του Dijkstra

Είσοδος: Ενα κατευθυνόμενο γράφημα $G = (V, E)$, μιά συνάρτηση κόστους l από το σύνολο E στο σύνολο των θετικών πραγματικών αριθμών και μιά κορυφή-πηγή $v_0 \in V$.
Εξοδος: Για κάθε $v \in V$, το ελάχιστο μονοπάτι από το v_0 στο v.

Ο αλγόριθμος κατασκευάζει ένα σύνολο $S \subseteq V$ έτσι ώστε το ελάχιστο μονοπάτι από την κορυφή-πηγή v_0 σε κάθε κορυφή $v \in V$ να αποτελείται μόνον από κορυφές του S.
Στον αλγόριθμο, η μεταβλητή $D[v]$ $(v \in V)$ συμβολίζει το κόστος του τρέχοντος ελαχίστου μονοπατιού από την κορυφή-πηγή v_0 στην κορυφή v. Η μεταβλητή $P[v]$ αποθηκεύει την προηγούμενη κορυφή της v στο τρέχον ελάχιστο μονοπάτι από την v_0 στην v.
Κατά την διάρκεια του αλγορίθμου, το τρέχον ελάχιστο μονοπάτι από την v_0 στην v μπορεί να αλλάξει. Ετσι, οι τιμές των $D[v]$ και $P[v]$ μπορεί να μεταβληθούν.

Θεώρημα 20 *Αλγόριθμος του Dijkstra απαιτεί χρόνο $O(|V|^2)$ στη χειρότερη περίπτωση.*

Αλγόριθμος Dijkstra$(G, l, v_0$
begin
  INITIALIZE-SINGLE-SOURCE(G, v_0)
  $S \longleftarrow \emptyset$
  while $S \neq V$ **do**
    διάλεξε μιά κορυφή $w \in V - S$ με την ελάχιστη τιμή $D[w]$
    $S \longleftarrow S \cup \{w\}$
    for κάθε $v \in V - S$ **do** RELAX(w, v)
end

procedure INITIALIZE-SINGLE-SOURCE(G, v_0)
begin
  for κάθε $v \in V$ **do**
      $D[v] \longleftarrow \infty$
      $P[v] \longleftarrow$NIL
  $D[v_0] \longleftarrow 0$
end

procedure RELAX(w, v)
begin
  if $D[v] > D[w] + l(w, v)$ **then**
      $D[v] \longleftarrow D[w] + l(w, v)$
      $P[v] \longleftarrow w$
end

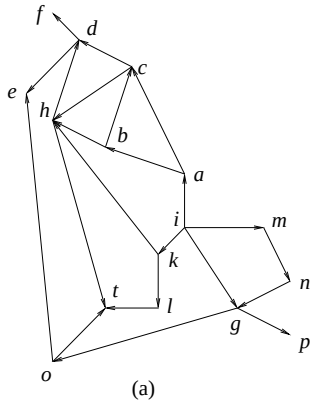
3.7.2 Ο αλγόριθμος των Bellman-Ford

Αυτός ο αλγόριθμος έχει εφαρμογή σε γραφήματα που οι ακμές τους μπορεί να περιέχουν και αρνητικές τιμές. Επιστρέφει μια Boolean τιμή FALSE όταν εντοπίσει στο γράφημα G ένα κύκλο αρνητικού κόστους. Στην περίπτωση αυτή το πρόβλημα δεν έχει λύση. Αν δεν υπάρχει τέτοιος κύκλος, ο αλγόριθμος παράγει όλα τα ελάχιστα μονοπάτια από την κορυφή-πηγή $v_0 \in V$ και επιστρέφει την Boolean τιμή TRUE.

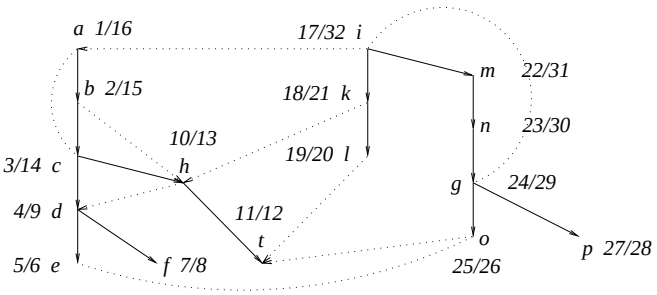
Αλγόριθμος Bellman-Ford(G, l, v_0)
begin
  INITIALIZE-SINGLE-SOURCE(G, v_0)
  for $i = 1$ **to** $|V| - 1$ **do**
    for κάθε ακμή $(w, v) \in E$ **do** RELAX(w, v)
  for κάθε ακμή $(w, v) \in E$ **do**
    if $D[v] > D[w] + l(w, v)$ **then return** FALSE
  return TRUE
end

Θεώρημα 21 *Αλγόριθμος των Bellman-Ford απαιτεί χρόνο $O(|V||E|)$, στη χειρότερη περίπτωση.*

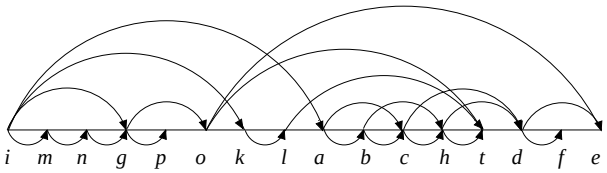
3.7.3 Κατευθυνόμενα μη-κυκλικά γραφήματα (directed acyclic graphs - DAG)



(a)



(b)



(c)

Σχήμα 13: (a) Ενα DAG $G = (V, E)$, (b) το DFS παράγον δάσος και (c) η λίστα $K[V]$ με τη τοπολογική διάταξη των κορυφών του V.

Μιά γραμμική διάταξη των κορυφών ενός DAG $G = (V, E)$ τέτοια ώστε, για κάθε ακμή $(u, w) \in E$ (με φορά από το u στο w) η κορυφή u να προηγείται της κορυφής w, καλείται **τοπολογική διάταξη** των κορυφών του V.

Αν το γράφημα $G = (V, E)$ ειναι κυκλικό τότε καμιά γραμμική διάταξη δεν μπορεί να οριστεί στις κορυφές του.

Μπορούμε να θεωρήσουμε την τοπολογική διάταξη σαν τη διάταξη των κορυφών του V κατά μήκος μιάς οριζόντιας γραμμής, έτσι ώστε όλες οι κατευθυνόμενες ακμές του E να έχουν φορά από τα αριστερά προς τα δεξιά.

Στη συνέχεια θα ορίσουμε μιά τοπολογική διάταξη σ'ένα DAG με Depth-First-Search.

Αλγόριθμος τοπολογικής διάταξης: Εφαρμόζουμε τον αλγόριθμο DFS στο γράφημα $G = (V, E)$. Κατα την διάρκεια της κατασκευής του DFS δάσους, αποθηκεύουμε σε κάθε κορυφή $w \in V$ δύο ακέραιους θετικούς αριθμούς a_1/a_2: a_1 είναι ο αύξων αριθμός του βήματος που εισάγει την κορυφή w στο παραγόμενο δάσος και a_2 είναι ο αύξων αριθμός του βήματος που τελειώνει την εξέταση της κορυφής w.
Οταν τελειώσει η εξέταση μιάς κορυφής, αυτή εισάγεται στην αρχή μιάς διασυνδεδεμένης λίστας $K[V]$. Μετά το τέλος του αλγορίθμου, η λιστα $K[V]$ περιέχει τις κορυφές του V σε τοπολογική διάταξη.

Θεώρημα 22 *Σε ένα γράφημα $G = (V, E)$, ο αλγόριθμος τοπολογικής διάταξης εκτελειται σε χρόνο $O(|V| + |E|)$.*

Λήμμα 23 *Ενα κατευθυνόμενο γράφημα G είναι μη-κυκλικό, αν και μόνον αν το σύνολο B των ακμών ενός DFS παραγόμενου δασους είναι κενό.*

Αλγόριθμος εύρεσης ελαχίστων μονοπατιών σε διατιμημένα DAG, από μιά κορυφή-πηγή $v_0 \in V$

DAG-SHORTEST-PATHS(G, l, v_0)
begin
κατασκεύασε την λίστα $K[V]$ που περιέχει τις κορυφές του V σε τοπολογική διάταξη
INITIALIZE-SINGLE-SOURCE(G, v_0)
σάρωσε τις κορυφές της λιστας $K[V]$ από την αρχή προς το τέλος
for καθε κορυφή $u \in K[V]$ **do**
    for κάθε κορυφή $w \in L[u]$ **do** RELAX(u, w)
end

Θεώρημα 24 *Ο αλγόριθμος DAG-SHORTEST-PATHS(G, l, v_0) εκτελείται σε χρόνο $O(|V| + |E|)$.*

4 Βιβλιογραφία

1. Fundamental Algorithms, D. E. Knuth, Addison-Wesley, 1968.

2. Sorting and Searching, D. E. Knuth, Addison-Wesley, 1973.

3. The Design and Analysis of Computer Algorithms, A. Aho, J. Hopcroft and J. Ullman, Addison-Wesley, 1974.

4. Data Structures and Algorithms, A. Aho, J. Hopcroft and J. Ullman, Addison-Wesley, 1983.

5. The Design of Dynamic Data Structures, M. H. Overmars, Lecture Notes in Computer Science, Springer-Verlag, 1983.

6. ALgorithms + Data Structures = Programs, N. Wirth, Computer Science Press, 1984.

7. Intoduction to Algorithms, T. Cormen, C. Leiserson and R. Rivest, The MIT Press, 1991.