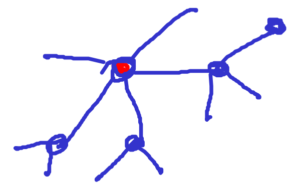
7. Drzewa

Teoria grafów i dendrologia spotykają się w następujących definicjach.

Definicja 7.1. Graf $G = (V, E)$ nazywamy **drzewem** jeżeli G jest grafem spójnym i nie zawiera cykli.

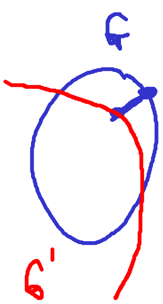
Jak pamiętamy, cykl w grafie to ciąg wierzchołków $x_1, x_2, \ldots, x_n = x_1$, w którym każde dwa kolejne połączone są krawędzią. Poniżej będziemy częściej nazywać cykl poprzez zbiór definiujących go krawędzi.



Definicja 7.2. Wierzchołek stopnia 1 w grafie nazywamy **liściem**.

Twierdzenie 7.3. *Rozważmy graf (prosty) $G = (V, E)$, gdzie $V \neq \emptyset$.*
(a) Jeżeli G jest drzewem i $|V| \geqslant 2$ to G posiada liść.
(b) Jeżeli G jest drzewem to $|E| = |V| - 1$.
(c) Jeżeli $|E| = |V| - 1$ i G nie zawiera cykli to G jest spójny, a więc jest drzewem.

Dowód. Aby sprawdzić (a) załóżmy, że $\deg(x) \geqslant 2$ dla każdego $x \in V$. Wybierzmy dowolny $x_1 \in V$, a nastepnie $x_2 \neq x_1$ połączony z x_1 krawędzią. Ponieważ $\deg(x_2) \geqslant 2$, istnieje $x_3 \in V \setminus \{x_1, x_2\}$ połączony krawędzią z x_2. Poruszamy się tak długo, aż napotkamy x_n, który już raz wystąpił w zbiorze odwiedzanych wierzchołków (co nastąpi, jako że rozważamy tylko grafy skończone). W ten sposób powstanie cykl.



Część (b) sprawdzamy przez indukcję po mocy V; dla $|V| = 1$ mamy $|E| = 0$ więc wzór zachodzi.

Wiemy, że G zawiera liść, powiedzmy że $x \in V$ jest wierzchołkiem stopnia 1 połączonym krawędzią jedynie z $y \in V$. Rozważmy $G' = (V', E')$, gdzie $V' = V \setminus \{x\}$, $E' = E \setminus \{x, y\}$. Zauważmy, że G' pozostaje spójny, a więc też jest drzewem. Z założenia indukcyjnego $|E'| = |V'| - 1$; to daje $|E| = |V| - 1$.

Dowód (c) przeprowadzamy nie wprost. Przypuśćmy, że G nie jest spójny. Oznacza to, że G ma co najmniej dwie składowe spójności. Powiedzmy , że V można podzielić na niepuste i rozłączne zbiory $V_1, \ldots, V_k$ w ten sposób, że każda częsc V_i jest spójna, natomiast nie ma krawędzi pomiędzy wierzchołkami z V_i i tymi z V_j dla $i \neq j$. Graf G rozkłada się w naturalny sposób na grafy $G_i = (V_i, E_i)$; tutaj E_i oznacza te krawędzie z E które łączą pewne wierzchołki z V_i.

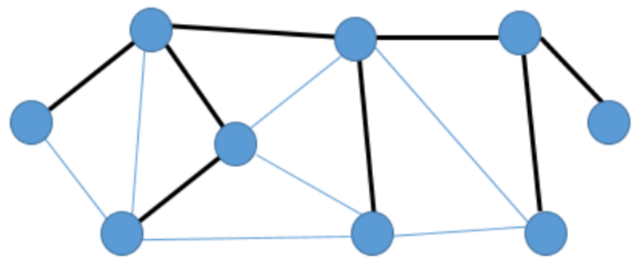
Wtedy G_i jest drzewem (jest spójny i nie zawiera cykli bo G ich nie zawierał). Z (b) wynika $|E_i| = |V_i| - 1$ dla każdego i. Dodając te równania stronami otrzymujemy

$$|E| = \sum_{i \leqslant k} |E_i| = \sum_{i \leqslant k} (|V_i| - 1) = \sum_{i \leqslant k} |V_i| - k = |V| - k < |V| - 1,$$

jako że $k \geqslant 2$; sprzeczność! $\square$

Definicja 7.4. Jeżeli $G = (V, E)$ jest grafem, a graf postaci $T = (V, E')$, gdzie $E' \subseteq E$, jest drzewem to T nazywamy **drzewem rozpinającym** w grafie G.

Drzewo rozpinające jest więc zbudowane na tych samych wierzchołkach co wyjściowy graf, ale ma na ogół mniej krawędzi — wyjściowy graf może zawierać cykle, a my budujemy minimalny szkielet połączeń między wierzchołkami.



Na rysunku czarne krawędzie tworzą drzewo rozpinające natomiast niebieskie krawędzie nie tworzą takiego drzewa (nie można się niebieskimi krawędziami dostać do wszystkich wierzchołków). Warto narysować samemu nieduży graf i przekonać się, że może on posiadać bardzo dużo różnych drzew rozpinających.

Twierdzenie 7.5. *Każdy spójny graf G posiada drzewo rozpinające.*

Dowód. Jeżeli G nie ma cykli to sam jest drzewem. Jeśli G ma cykl C to można z C usunąć jedną krawędź i taki graf G' pozostaje spójny. Tę uwagę można zastosować do G'. Postępując w ten sposób odchudzimy graf do grafu spójnego bez cykli, czyli drzewa właśnie. □

Rozważymy teraz pytanie, ile drzew rozpinających ma pełny graf K_n, posiadający n wierzchołków i wszystkie $\binom{n}{2}$ krawędzie. Dla $n = 3$ drzewo rozpinające ma 2 krawędzie i każda para krawędzi takie drzewo tworzy; tym samym są 3 drzewa rozpinające. Dla $n = 4$ narysowanie wszystkich drzew rozpinających jest już uciążliwe. Łatwo jednak je zliczyć: Graf ma $\binom{4}{2} = 6$ krawędzi, a więc jest $\binom{6}{3} = 20$ trójek krawędzi, które mogą stanowić drzewo rozpinające. Są 4 złe układy (4 cykle), a wieć odpowiedź jest taka, że K_4 ma 16 drzew rozpinających. Zachodzi następujące twierdzenie.

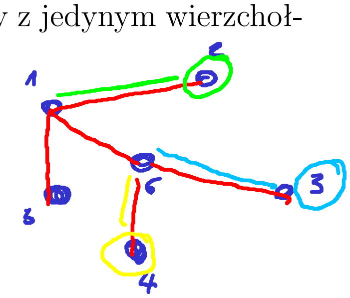
Twierdzenie 7.6 (Caley)**.** *Dla każdego n graf K_n ma n^{n-2} drzew rozpinających.*

Dowód twierdzenia można przeprowadzić w oparciu o tak zwany **kod Prŭfera**. Niech K_n będzie grafem na zbiorze wierzchołków $V = \{1, 2, \ldots, n\}$. Poniższy algorytm koduje drzewa rozpinające w K_n za pomocą ciągów ze zbioru $\{1, 2, \ldots, n\}^{n-2}$.

Zaczynamy od pewnego drzewa rozpinającego $T = (V, E')$ grafu K_n i budujemy ciąg drzew $T_i = (V_i, E_i)$, gdzie $V = V_0 \supseteq V_1 \supseteq V_2 \supseteq \ldots$ i $E' = E_0 \supseteq E_1 \supseteq E_2 \supseteq \ldots$ jednocześnie budując kod $(c_1, c_2, \ldots, c_{n-2})$ drzewa T:

$|\{1, \ldots, n\}^{n-2}| = n^{n-2}$

(1) Jeżeli drzewo T_i ma jedną krawędź to STOP.
(2) Znajdujemy w T_i liść v o najniższym numerze. Jest on połączony z jedynym wierzchołkiem $w \in E_i$.
(3) Definiujemy $c_{i+1} = w$, $V_{i+1} = V_i \setminus \{v\}$, $E_{i+1} = E_i \setminus \{\{v, w\}\}$.
(4) Wracamy do (1).



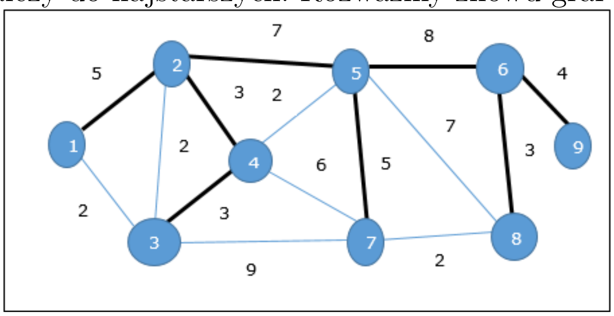
Dowód twierdzenia Caley'a opiera się na dwóch faktach.

$n = 6$ $\quad T \longrightarrow (1, 6, 6, 1)$

Lemat 7.7. *(a) Różne drzewa rozpinające mają różne kody.*
(b) Każdy ciąg $(c_1, c_2, \ldots, c_{n-2}) \in \{1, 2, \ldots, n\}^{n-2}$ *jest kodem pewnego drzewa rozpinającego.*

Analiza tych faktów będzie sugerowanym zadaniem na ćwiczenia; (*b*) wymaga opisania algorytmu tworzenia drzewa na podstawie kodu — można zajrzeć do Wiki

Skoro drzew rozpinających może być w danym grafie tak dużo to, aby wybrać drzewo rozpinające o pewnej własności, warto zaprojektować efektywny algorytm. Poniższy przykład należy do najstarszych. Rozważmy znowu graf



w którym krawędziom przypisano umowne koszty, na przykład fizyczne odległości. Jeżeli jest to schemat połączeń pomiędzy osiedlami, a firma chce wszystkie osiedla połączyć światłowodem, to będzie szukać najtańszego drzewa rozpinającego, czyli takiego że suma kosztów jego krawędzi jest najmniejsza. To współczesność; w XIX wieku rozważano ten problem dla zbudowania najmniej kosztownej sieci połączeń kolejowych pomiędzy miastami (wzdłuż istniejących połączeń drogowych).

Ogólniej problem, zwany *Minimal Spanning Tree Problem (MSTP)*, rozważa graf spójny $G = (V, E)$ i funkcję $c : E \to \mathbb{R}_+$ kosztów krawędzi. Problem polega na znalezieniu drzewa rozpinającego $T = (V, E')$ w tym grafie, dla którego koszt czyli wielkość

$$c(E') := \sum_{e \in E'} c(e),$$

jest najmniejsza.

Algorytm rozwiązujący MSTP jest dość prosty i *chciwy*[1], za każdym razie wybieramy najtańszą opcję w następujących krokach:

(1) Wybieramy krawędź $e_1 \in E$ o najmniejszym koszcie $c(e_1)$ (tych krawędzi może być więcej, wtedy bierzemy jedną z nich).

(2) W następnych krokach dany jest zbiór krawędzi $e_1, \ldots, e_i \in E$ bez cyklu, Jeżeli $i = |V| - 1$ to STOP. W przeciwnym przypadku zbiór $F = E \setminus \{e_1, \ldots, e_i\}$ jest niepusty; wybieramy $e_{i+1} \in F$ taką, że $c(e_{i+1})$ ma najmniejszą wartość dla tych $e \in F$, dla których zbiór $\{e_1, \ldots, e_i, e\}$ nie zawiera cyklu.

Niech $|V| = n$; zauważmy, że algorytm zatrzyma się po kroku $n-1$ wskazując zbiór $n-1$ krawędzi bez cyklu, to jest definiując drzewo rozpinające, patrz Twierdzenie 7.3. Należy się jednak upewnić, że wskazane drzewo jest faktycznie najtańsze (a może łakomiąc się na tanią krawędź w pierwszym kroku, potem jesteśmy skazani na bardzo kosztowne?). Dowód opera się na następującym lemacie *o wymianie krawędzi w drzewie rozpinającym.*

---

[1] algorytm chciwy (ang, *greedy*) to taki, który idzie po najmniejszej linii oporu

Lemat 7.8. *Niech $T_1 = (V, E_1)$ i $T_2 = (V, E_2)$ będą drzewami rozpinającymi w grafie $G = (V, E)$. Jeżeli $e_1 \in E_1 \setminus E_2$ to istnieje taka krawędź $e_2 \in E_2$, że krawędzie $E_1 \setminus \{e_1\} \cup \{e_2\}$ tworzą drzewo rozpinające.*

Dowód. Zbiór krawędzi $E_2 \cup \{e_1\}$ ma moc $|V|$, a więc zawiera cykl C - patrz Twierdzenie 7.3 (tutaj cykl rozumiemy jako zbiór krawędzi, a nie wierzchołków). Wtedy $C \setminus E_1$ jest niepusty (E_1 są krawędzimi drzewa więc nie zawierają cyklu). Wybieramy $e_2 \in C \setminus E_1$; wtedy $E_1 \setminus \{e_1\} \cup \{e_2\}$ są faktycznie krawędziami drzewa rozpinającego na mocy Twierdzenia 7.3(c). □

Twierdzenie 7.9. *Opisany wyżej algorytm wskazuje drzewo rozpinajace o najmniejszym koszcie.*

Dowód. Niech T będzie drzewem o krawędziach $e_1, \ldots, e_{n-1}$ wskazanym przez algorytm. Niech $T_0 = (V, E_0)$ będzie optymalnym drzewem rozpinającym, minimalizującym koszty.

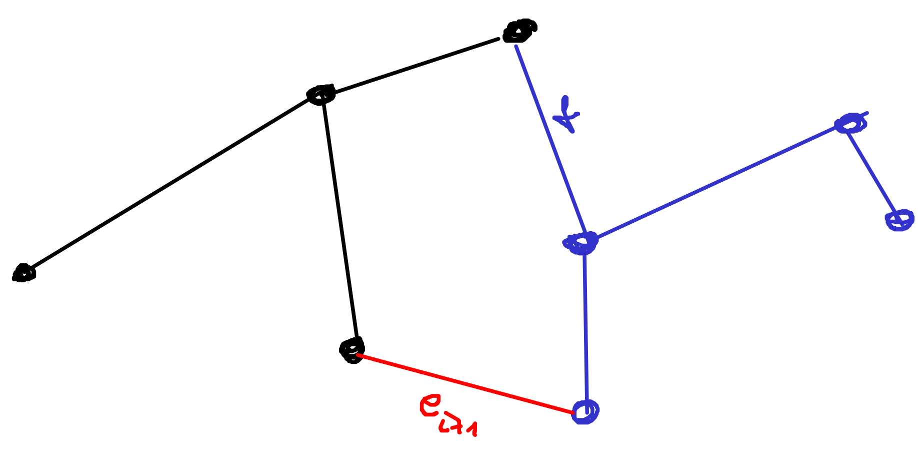
Jeżeli te dwa drzewa są różne to wybieramy pierwsze taki i, że e_{i+1} nie jest krawędzią w drzewie optymalnym T_0. Na mocy Lematu 7.8 istnieje krawedź f w T_0, taka że krawędzie $E_1 = E_0 \setminus \{f\} \cup \{e_{i+1}\}$ też tworzą drzewo rozpinające. Wtedy $c(E_0) \leqslant C(E_1)$ (jako że E_0 tworzą optymalne drzewo); dlatego

$$c(E_0) \leqslant c(E_1) = c(E_0) - c(f) + c(e_{i+1}).$$

$\Longrightarrow c(f) = c(e_{i+1})$

Stąd $c(f) \leqslant c(e_{i+1})$. Z drugiej strony, $c(e_{i+1}) \leqslant c(f)$ bo tak działa algorytm w tym kroku. Ostatecznie $c(E_0) = c(E_1)$, czyli krawędzie E_1 tez tworzą optymalne drzewo.

To rozumowanie pokazało, że możemy zwiększać ilość wspólnych krawędzi w drzewie wskazanym prze algorytm i w drzewie optymalnym. Po skończonej ilości takich operacji dojdziemy do wniosku, że wybrane przez algorytm drzewo też jest optymalne. □



czarne: wspólne krawędzie T i T_0
niebieskie: pozostałe krawędzie optymalnego drzewa T_0