Teoria grafów i sieci

prof. dr hab. M. Kubale

Katedra Algorytmów i Modelowania Systemów

Hymn teorii grafów

Text by BOHDAN ZELINKA
Music by ZDENĚK RYJÁČEK
English text by DONALD A. PREECE

1.
Seven bridges spanned the River Pregel,
Many more than might have been expected;
Königsberg's wise leaders were delighted
To have built such very splendid structures.

2.
Crowds each ev'ning surged towards the river,
People walked bemused across the bridges,
Pondering a simple-sounding challenge
Which defeated them and left them puzzled.

3.
Here's the problem; see if you can solve it!
Try it out at home an scraps of paper!
STARTING OUT AND ENDING AT THE SAME SPOT,
YOU MUST CROSS EACH BRIDGE JUST ONCE EACH EV'NING.

Refrain:
Eulerian graphs all have this restriction:
THE DEGREE OF ANY POINT IS EVEN.
That's the oldest graph result
That mankind has ever known.

4.
All the folk in Königsberg were frantic!
All their efforts ended up in failure!
Happily, a learn-ed math'matician
Had his house right there within the city.

5.
Euler's mind was equal to the problem:
"Ah", he said, "You're bound to be disheartened.
Crossing each bridge only once per outing
Can't be done, I truly do assure you."

Refrain:
Eulerian graphs …

6.
Laws of Nature never can be altered,
We can'd change them, even if we wish to.
Nor can flooded rivers or great bridges
Interfere with scientific progress.

7.
War brought strife and ruin to the Pregel;
Bombs destroyed those seven splendid bridges.
Euler's name and fame will, notwithstanding,
Be recalled with Königsberg's for ever.

Refrain:
Eulerian graphs …

8.
Thanks to Euler, Graph Th ry is thriving.
Year by year it flourishes and blossoms,
Fertilising much of mathematics
And so rich in all its applications.

9.
Colleagues, let us fill up all our glasses!
Colleagues, let us raise them now to toast the
Greatness and the everlasting glory
Of our Graph Th ry, which we love dearly!

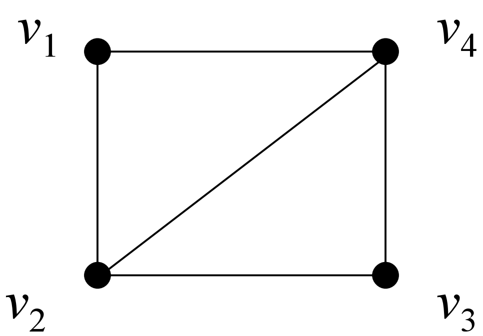
Hymn teorii grafów

Pojęcie grafu

Def. *Graf* prosty G=(V,E) jest uporządkowaną parą dwóch elementów: zbioru wierzchołków V oraz zbioru krawędzi $E \subset V \times V$. Krawędź pomiędzy wierzchołkami u oraz v oznaczamy $\{u,v\}$.

Def. Wierzchołki u i v są *sąsiednie*, gdy $\{u,v\} \in E(G)$. Wierzchołek u oraz krawędź e są *incydentne*, gdy $u \in e$.

Przykład:



$$G = (\ \{ v_1, v_2, v_3, v_4 \},$$

$$\{ \{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}, \{v_2, v_3\} \ \}$$

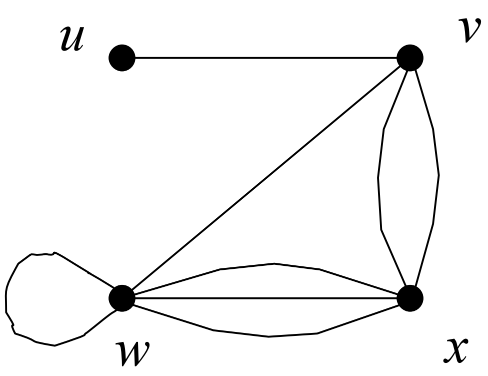
Uwaga. Rysunek grafu jest reprezentacją zbioru wierzchołków oraz sposobu ich połączenia, więc jego własności metryczne nie są istotne.

Multigrafy oraz grafy skierowane

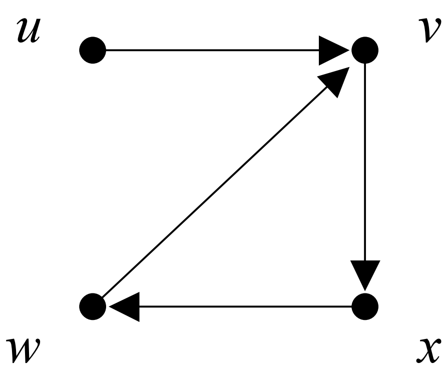
Def. *Multigraf* to graf, w którym pomiędzy dowolną parą wierzchołków może wystąpić więcej niż jedna krawędź oraz dopuszczalne są *pętle*, tzn. krawędzie postaci $\{v,v\}$, gdzie $v \in V(G)$.

Def. Graf nazywamy *digrafem* (*grafem skierowanym*), gdy krawędź łącząca u oraz v jest uporządkowaną parą postaci (u,v).

Przykład:



$G = (\{u, v, w, x\},$
$\{\{u,v\}, \{w,v\}, \{w,w\}, \{w,x\},$
$\{w,x\}, \{w,x\}, \{x,v\}, \{x,v\}\})$



$D = (\{u, v, w, x\},$
$\{(u,v), (w,v), (v,x), (x,w)\})$

Stopień wierzchołka

Def. *Stopień* wierzchołka v jest oznaczany symbolem $\deg(v)$ i jest to ilość wierzchołków sąsiednich z v. Dla danego grafu G definiujemy parametry:

$$\delta = \min \{ \deg(v) : v \in V(G) \}$$
$$\Delta = \max \{ \deg(v) : v \in V(G) \}$$

Lemat o uściskach dłoni. *Niech G będzie grafem prostym. Wówczas*

$$\deg(v_1) + ... + \deg(v_n) = 2m.$$

Dowód: (indukcja ze względu na liczbę krawędzi grafu)

1. Jeśli $m = 0$, to równość jest prawdziwa.
2. Zakładamy, że lemat zachodzi dla pewnego $m \geq 0$.
3. Udowodnimy równanie dla $m + 1$. Niech $e \in E(G)$ będzie dowolną krawędzią. Z założenia indukcyjnego wiadomo, że
$$\deg(v_1) + ... + \deg(v_n) = 2m(G - e),$$
gdzie $\deg(v_i)$ jest stopniem v_i w grafie $G - e$. Dla grafu G suma stopni wierzchołków jest o 2 większa niż dla $G - e$ oraz $m(G) = m(G - e) + 1$, co oznacza, że równanie zachodzi dla grafu G o $m + 1$ krawędziach.

Twierdzenie Havla

Tw. *Ilość wierzchołków nieparzystego stopnia w grafie prostym jest parzysta.*
Dowód: Z lematu o uściskach dłoni wiadomo, że

$$\deg(v_1) + ... + \deg(v_k) + \deg(u_1) + ... + \deg(u_l) = 2m,$$

gdzie v_i jest wierzchołkiem o nieparzystym stopni, natomiast u_i – parzystym. Wiadomo, że suma liczb parzystych $\deg(u_1) + ... + \deg(u_l)$ jest liczbą parzystą, więc suma $\deg(v_1) + ... + \deg(v_k)$ ma również parzystą wartość, co jest spełnione tylko wtedy, gdy ilość składników jest parzysta (k jest parzyste).

Def. *Ciągiem stopniowym* grafu nazywamy uporządkowany nierosnąco ciąg stopni jego wierzchołków.

Def. Ciąg stopniowy jest *graficzny*, gdy jest on ciągiem stopni pewnego grafu.

Tw. Havla *Ciąg stopniowy* $(d, d_1, ..., d_l)$ *jest graficzny wtedy i tylko wtedy, gdy* $(d_1 - 1, d_2 - 1, ..., d_d - 1, d_{d+1}, d_{d+2}, ..., d_l)$ *jest graficzny.*

Ciągi graficzne

Algorytm sprawdzający, czy ciąg stopni jest graficzny:

1. Jeśli $d = d_1 = \ldots = d_l = 0$, to ciąg jest graficzny
2. Jeśli $d < 0$, to ciąg nie jest graficzny
3. Uporządkuj ciąg nierosnąco wg stopni
4. Usuń z ciągu pierwszą (największą) liczbę d i odejmij 1 od kolejnych d liczb ciągu
5. Wróć do punktu 1

Przykład: Sprawdzić, czy podany ciąg jest graficzny:

$$(4, 3, 3, 2, 2)$$

$$(2, 2, 1, 1)$$

$$(1, 0, 1) \longrightarrow (1, 1, 0)$$

$$(0, 0)$$

Odpowiedź: podany ciąg stopniowy jest graficzny.

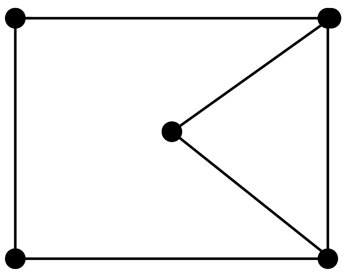
Ciągi graficzne

Przykład

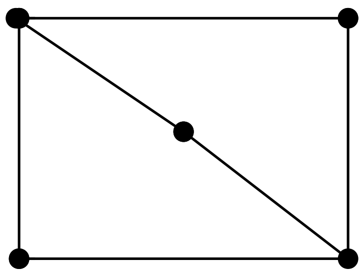
•Ciąg (6,5,4,4,3,3,2) nie jest graficzny, ponieważ suma stopni odpowiedniego grafu wynosiłaby 27, co nie jest możliwe na podstawie lematu o uściskach dłoni.

•Ciąg stopni (3,3,2,2,2) jest graficzny, ponieważ jest on ciągiem grafów prostych G_1, G_2 oraz multigrafu G_3 pokazanych poniżej.

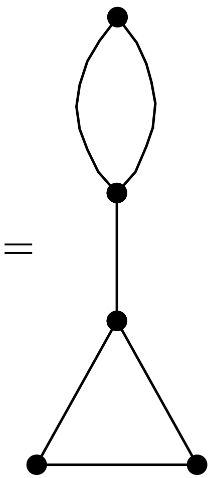
$G_1 =$



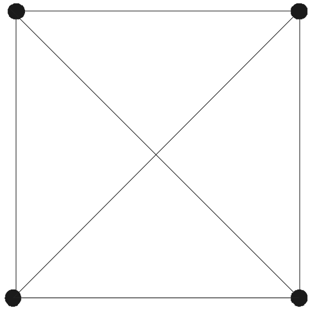
$G_2 =$



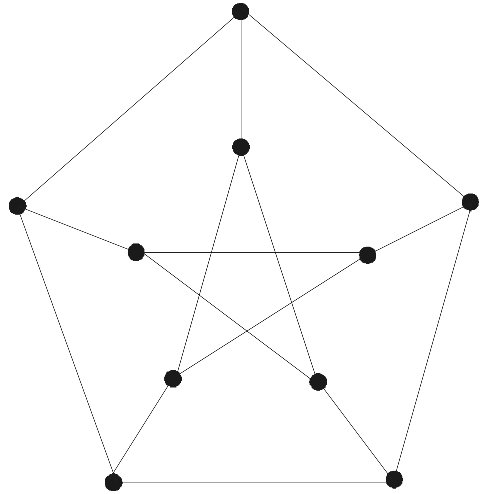
$G_3 =$



Przykłady grafów



Graf K_4



Graf kubiczny
(graf Petersena)

- *Graf pusty* $N_n=(\{v_1,...,v_n\},\varnothing)$.
- *Graf pełny* $K_n=(\{v_1,...,v_n\},\{\{v_i,v_j\}: i,j=1,...,n,\ i\neq j\})$.
- *Graf r-regularny* jest grafem, dla którego zachodzi równość $r = \delta = \Delta$ (tzn. stopień każdego wierzchołka jest równy r). Wówczas $m=(r\cdot n)/2$. Graf 3-regularny nazywamy *grafem kubicznym*. Zdefiniowany wcześniej graf pełny o n wierzchołkach jest $(n-1)$-regularny.

Operacje sumy i zespolenia

Def. (suma grafów)

$$G_1 \cup G_2 = (\, V(G_1) \cup V(G_2),\ E(G_1) \cup E(G_2)\,)$$

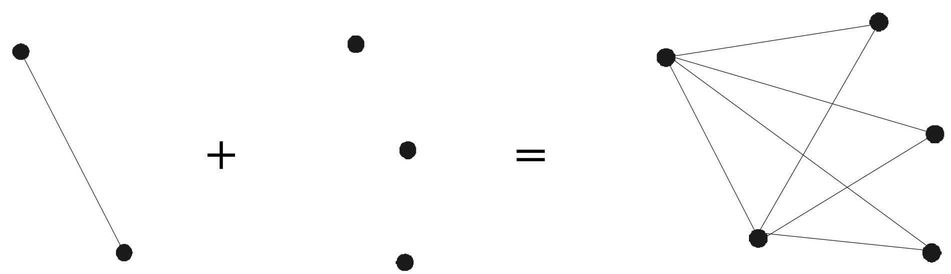
Def. (zespolenie grafów)

$$G_1 + G_2 = (\, V(G_1) \cup V(G_2),\ E(G_1) \cup E(G_2) \cup \{\{v_1, v_2\} : v_1 \in V(G_1), v_2 \in V(G_2)\}\,)$$

Tw. *Jeśli* $G = G_1 + G_2$, *to*

$$n(G) = n(G_1) + n(G_2),$$
$$m(G) = m(G_1) + m(G_2) + n(G_1) \cdot n(G_2).$$

Przykład Zespolenie grafów K_2 i N_3



Dopełnienie grafu

Def. Jeśli G jest grafem prostym, to jego dopełnieniem jest graf $\overline{G}$, o tym samym zbiorze wierzchołków oraz dwa wierzchołki są sąsiednie w G’ wtedy i tylko wtedy, gdy nie są sąsiednie w G.

Uwaga Przykładowe zależności:

$$\overline{\overline{G}} = G,$$

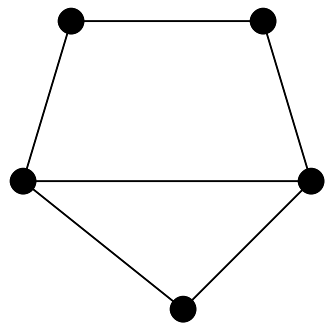
$$\overline{K_n} = N_n,$$

$$\overline{K}_{r,s} = K_r \cup K_s,$$

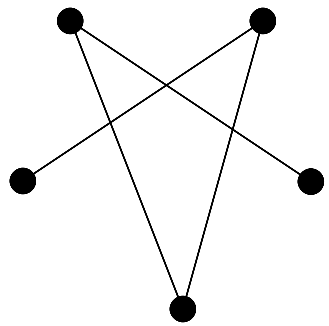
$$\overline{G}_r = G_{n-r-1},$$

gdzie G_r oznacza dowolny graf r–regularny.

$G =$



$\overline{G} =$



Pojęcie podgrafu

Def. *Podgrafem* grafu G nazywamy dowolny graf H taki, że
$V(H) \subset V(G)$ oraz $E(H) \subset E(G)$.

Uwaga Podgraf można otrzymać usuwając pewną liczbę krawędzi i (lub) wierzchołków z grafu G. Jeśli $e \in E(G)$, to przez $G - e$ rozumiemy podgraf powstały przez usunięcie krawędzi e z grafu G. Jeśli $F \subset E(G)$ to $G - F$ jest grafem utworzonym z G w wyniku usunięcia krawędzi należących do zbioru F. Analogicznie, jeśli v, S są odpowiednio wierzchołkiem oraz zbiorem wierzchołków grafu G, to $G - v$ oraz $G - S$ definiujemy jako grafy powstałe poprzez usunięcie odpowiednio v lub S z grafu G.

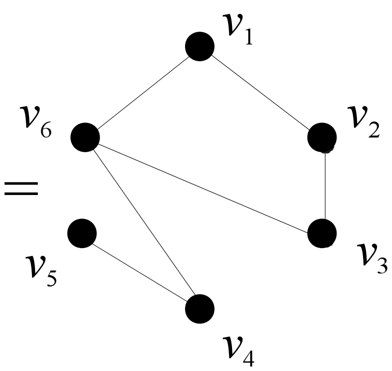
Podgraf indukowany

Def. *Podgraf indukowany* $G[S]$ grafu G, gdzie $S \subset V(G)$, definiujemy następująco:

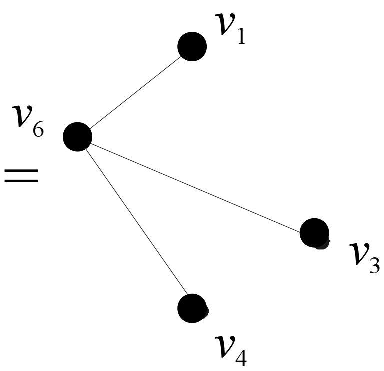
$$G[S] = (\ S,\ \{\{u,v\} : u,v \in S,\ \{u,v\} \in E(G)\}\ ).$$

Przykład (podgraf indukowany)

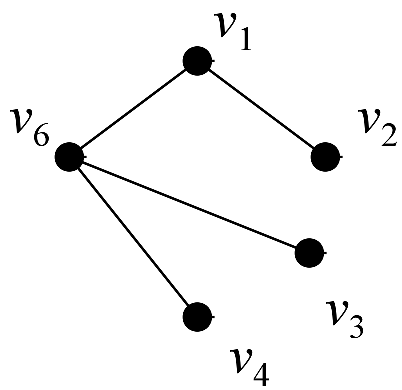
$G\ =$



$G[\{v_1, v_3, v_4, v_6\}] =$



Podgraf G, który nie jest podgrafem indukowanym:

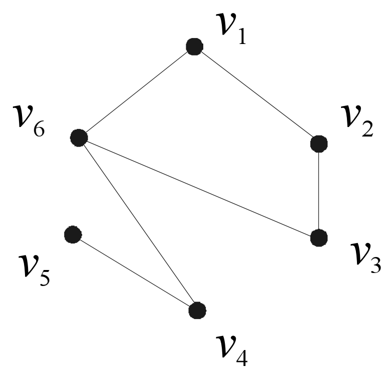


Drogi i cykle

Def.

- *Marszruta* to taki ciąg krawędzi grafu G, że dwie kolejne krawędzie tego ciągu są sąsiednie w G lub są identyczne.
- *Łańcuch* jest marszrutą, w której żadna krawędź nie występuje dwukrotnie.
- *Droga* to marszruta w której każdy wierzchołek występuje co najwyżej raz.
- *Cykl* to łańcuch, którego wierzchołek początkowy jest równy końcowemu.

Przykład

$G =$ 

marszruta: $v_1, v_6, v_3, v_6, v_4, v_5$

łańcuch: $v_6, v_1, v_2, v_3, v_6, v_4$

droga: $v_1, v_2, v_3, v_6, v_4, v_5$

cykl: v_1, v_2, v_3, v_6, v_1

Drogi i cykle

Tw. *Jeśli* $\delta = 2$, *to graf* G *zawiera cykl*.

Dowód:

(a) Mamy $m \geq n$, gdyż z lematu o uściskach dłoni wiadomo, że

$$2m = \deg(v_1) + \ldots + \deg(v_n) \geq n\delta = 2n.$$

(b) Udowodnimy indukcyjnie względem n, że jeśli $m \geq n$, to G zawiera cykl.

- Jeśli $n = 2$, to G zawiera cykl.
- Zakładamy, że twierdzenie zachodzi dla wszystkich grafów, których rząd nie przekracza n.
- Rozważmy $n + 1$ wierzchołkowy graf G oraz niech $e = \{u,v\}$ będzie jego dowolną krawędzią. Jeśli $G - e$ jest spójny, to istnieje ścieżka z u do v w $G - e$ więc ta ścieżka w połączeniu z e tworzy cykl w G. Jeśli G nie jest spójny to oznaczmy $G - e = G_1 \cup G_1$. Gdyby zachodziło $m(G_1) \leq n(G_1) - 1$ oraz $m(G_1) \leq n(G_1) - 1$, to

$$m(G) = m(G_1) + m(G_1) + 1 \leq n(G_1) + n(G_2) - 1 = n(G) - 1,$$

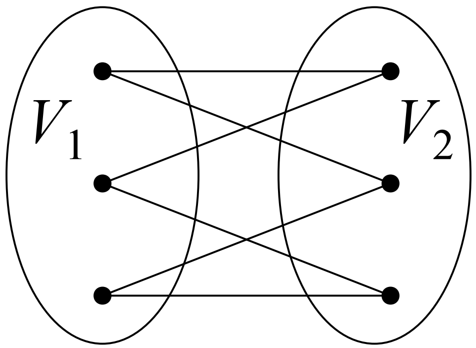
co daje sprzeczność z zał. indukcyjnym. Stąd można przyjąć, że $m(G_1) \geq n(G_1)$ i z założenia indukcyjnego G_1 (więc również G) zawiera cykl.

(c) Z (a) oraz (b) wynika teza.

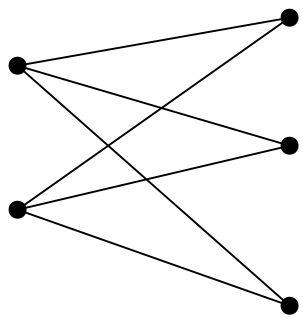
Grafy dwudzielne

- *Graf dwudzielny* $G = (V_1 \cup V_2, E)$, gdzie $G[V_1]$ oraz $G[V_2]$ są grafami pustymi (istnieje podział V na podzbiory V_1 i V_2 takie, że każda krawędź grafu łączy wierzchołki z różnych zbiorów V_i). Szczególnym przypadkiem są pełne grafy dwudzielne $K_{a,b} = N_a + N_b$. W takim przypadku zachodzą wzory.

$$m(K_{a,b}) = a \cdot b,$$
$$n(K_{a,b}) = a + b.$$



Graf dwudzielny (2-regularny)

$K_{2,3} =$ 

Grafy dwudzielne

Tw. *Graf G jest dwudzielny wtedy i tylko wtedy, gdy każdy cykl w G ma parzystą długość.*

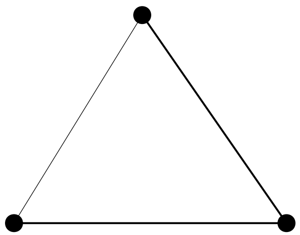
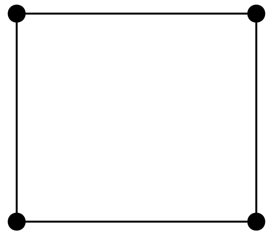
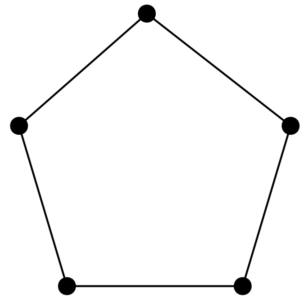
Dowód:

Załóżmy, że $G=(V_1 \cup V_2, E)$ jest dwudzielny oraz $v_1,\ldots,v_k$ jest dowolnym jego cyklem. Bez straty ogólności można założyć, że $v_1 \in V_1$. Stąd, że v_1 i v_2 są sąsiednie wynika, że $v_2 \in V_2$. Ogólnie, $v_{2p} \in V_2$ oraz $v_{2p+1} \in V_1$, co oznacza, że wierzchołek $v_k \in V_2$ ponieważ jest sąsiedni z v_1.

Załóżmy, że każdy cykl w G ma parzystą długość. Definiujemy podział V następująco: $A_v = \{w \in V : d(v,w) \text{ jest parzyste}\}$, $B_v = V \setminus A_v$, gdzie $d(v,w)$ jest długością najkrótszej ścieżki łączącej v z w. Wierzchołki w A_v są parami niesąsiednie, gdyż sytuacja $x, y \in A_v$ oraz $\{x,y\} \in E$ oznacza, że w G występuje nieparzysty cykl o długości $d(v,x) + d(v,y) + 1$, sprzeczność. Podobnie można uzasadnić że krawędź pomiędzy elementami B_v implikuje istnienie nieparzystego cyklu w G.

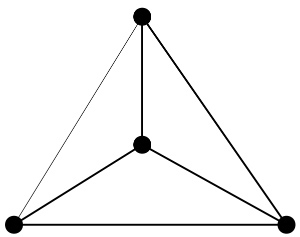
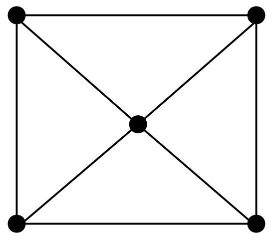
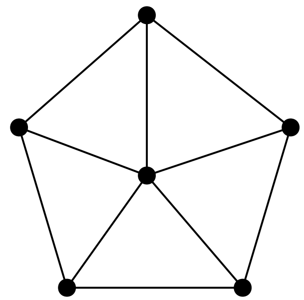
Przykłady grafów

- *Cykl* $C_n = (\{v_1,\dots,v_n\} , \{\{v_1,v_2\},\dots,\{v_{n-1},v_n\}, \{v_n,v_1\}\})$

$C_3 =$  $C_4 =$  $C_5 =$ 

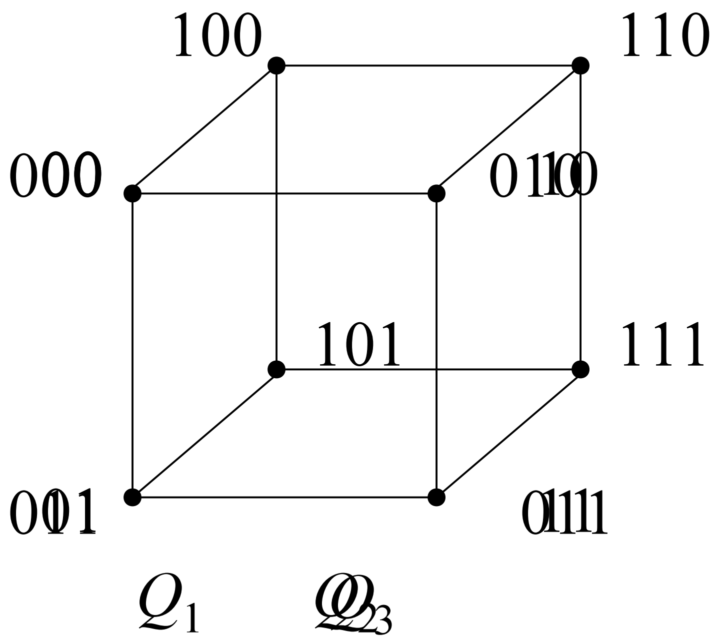
- *Koło* $W_n = (\{v_1,\dots,v_n\} , \{\{v_1,v_2\},\dots,\{v_{n-1},v_n\}, \{v_n,v_1\}\})$

$$W_n = C_{n-1} + N_1$$

$W_4 =$  $W_5 =$  $W_6 =$ 

Hiperkostki

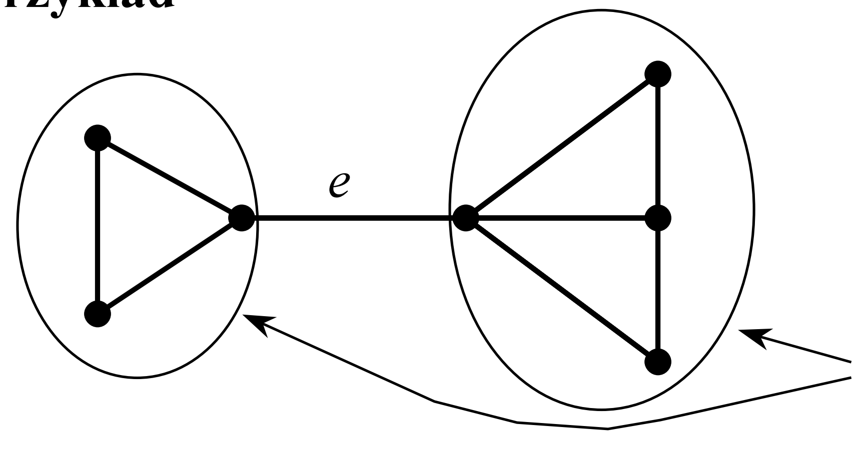
• hiperkostka Q_n jest grafem, którego wierzchołki odpowiadają wszystkim ciągom zero-jedynkowym długości n. Dwa wierzchołki są sąsiednie, o ile odpowiadające im ciągi różnią się na jednej pozycji.



Spójność krawędziowa

Def. Graf jest *spójny*, gdy pomiędzy każdą parą wierzchołków istnieje droga. *Zbiór rozspajający* grafu to zbiór krawędzi, których usunięcie rozspaja graf. *Rozcięcie* to zbiór rozspajający, którego żaden właściwy podzbiór nie rozspaja grafu. *Spójność krawędziowa* jest mocą najmniejszego rozcięcia (oznaczenie: $\lambda(G)$). *Składowa spójności* grafu G jest podgrafem H takim, że $G = H \cup G'$, dla pewnego G'. Krawędź e nazywamy *mostem*, gdy $\{e\}$ jest rozcięciem.

Przykład



Zbiór rozspajający

Rozcięcie

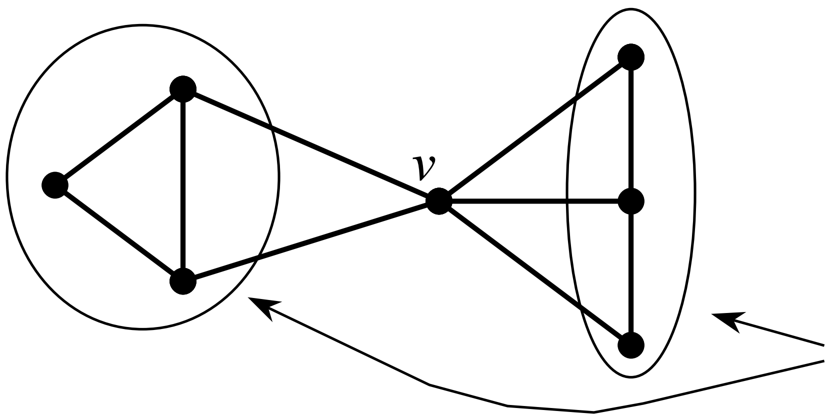
Most

Składowe spójności grafu $G - e$

Spójność wierzchołkowa

Def. *Zbiór separujący* grafu to zbiór wierzchołków, których usunięcie rozspaja graf. *Separator* to zbiór separujący, którego żaden właściwy podzbiór nie rozspaja grafu. *Spójność wierzchołkowa* jest mocą najmniejszego rozcięcia (oznaczenie: $\chi(G)$). Wierzchołek v nazywamy *przegubem*, gdy $\{v\}$ jest separatorem.

Przykład



Zbiór separujący

Separator

Przegub

Składowe spójności grafu $G - v$

Spójność grafu

Tw. *Jeśli graf G posiada k składowych spójności, to*

$$n - k \leq m.$$

Dowód: Dowodzimy twierdzenie indukcyjnie względem k.

1) Niech $k = 1$. Graf ma minimalną ilość krawędzi wówczas, gdy usunięcie dowolnej z nich rozspaja graf. Wybierzmy dowolny wierzchołek będący liściem i usuńmy go z grafu wraz z incydentną krawędzią. W wyniku tej operacji liczba krawędzi i wierzchołków maleją o 1. Po pewnej liczbie kroków otrzymujemy graf K_2.
2) Zakładamy, że twierdzenie zachodzi dla pewnego $k > 1$.
3) Dowodzimy dla $k + 1$. Dodajemy do grafu krawędź e tak, aby łączyła dwie składowe spójności. Wówczas korzystając z założenia indukcyjnego otrzymujemy:

$$n(G+e) - (k - 1) \leq m(G+e).$$

Po uwzględnieniu, że $n(G+e) = n(G)$ oraz $m(G+e) = m(G) + 1$ otrzymujemy tezę.

Spójność grafu

Tw. *Jeśli graf G posiada k składowych spójności, to*

$$m \le (n-k)(n-k+1)/2.$$

Dowód: Rozważmy n-wierzchołkowy graf G, o największej możliwej liczbie krawędzi, który posiada k składowych spójności $G_1,\dots,G_{\mathrm{k}}$ takich, że $|V(G_{i+1})| \le |V(G_i)|$. Załóżmy, że wierzchołek u należy do najliczniejszej składowej spójności , natomiast v należy do składowej G_2. Zauważmy, że jeśli $|V(G_2)| > 1$, to przeniesienie wierzchołka v z G_2 do G_1 nie zmieni liczby składowych spójności, liczba krawędzi natomiast wzrośnie, co daje sprzeczność. Stąd $|V(G_i)| = 1$ dla $i > 1$. Stąd, $m(G) = m(K_{n-k}) = (n-k)(n-k+1)/2$, co kończy dowód.

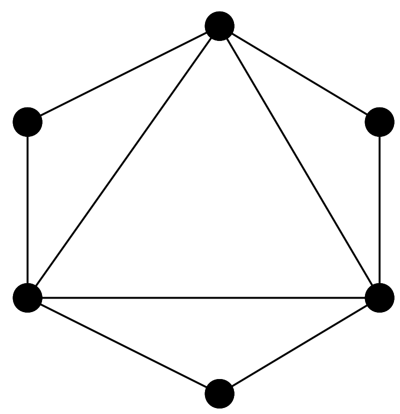
Wniosek *Dowolny n-wierzchołkowy graf posiadający więcej niż* $(n-2)(n-1)/2$ *krawędzi jest spójny*.

Sprawdzanie spójności grafu

Procedure Spójny(G)
begin
  S := { v }; (* gdzie v jest dowolnym wierzchołkiem grafu G *)
  while istnieje nieoznaczony wierzchołek $v \in S$ **do begin**
    $S := S \cup N(v)$;
    oznacz wierzchołek v;
  end;
  if $S = V(G)$ **then**
    return „graf G jest spójny";
  else
    return „graf G nie jest spójny";
end

Przykład Oznaczenia:

- oznaczone elementy S;
- nieoznaczone elementy S;
- bieżący wierzchołek v;



Odp: „graf G jest spójny"

Drzewa

- *Drzewo* – graf spójny, który nie zawiera podgrafu będącego cyklem. Dla tej klasy grafów zachodzi $m = n - 1$. Wierzchołek o stopniu równym 1 nazywamy *liściem*.

- ścieżki: $P_n = (\{v_1,...,v_n\},\{\{v_1,v_2\}, ...,\{v_{n\text{-}1},v_n\}\})$
- gwiazdy: $K_{1,n} = (\{v_1,...,v_n\},\{\{v_1,v_2\}, ...,\{v_1,v_n\}\}) = N_1 + N_{n\text{-}1}$
- kometa – graf powstały poprzez połączenie pewnego wierzchołka gwiazdy z liściem należącym do ścieżki
- gąsienica – graf, w którym można wyróżnić taką ścieżkę, że każdy liść w grafie jest sąsiedni z pewnym wierzchołkiem ścieżki
- dwugwiazda – wszystkie wierzchołki, z wyjątkiem dwóch mają stopień równy 1
- drzewo binarne – drzewo, w którym stopień każdego wierzchołka jest równy co najwyżej 3

Drzewa

Tw. Niech T będzie drzewem o n wierzchołkach. Wówczas następujące warunki są równoważne:

(1) T jest drzewem;

(2) T nie zawiera cykli i ma $n - 1$ krawędzi;

(3) T jest grafem spójnym i ma $n - 1$ krawędzi;

(4) T jest grafem spójnym i każda krawędź jest mostem;

(5) każde dwa wierzchołki T są połączone dokładnie jedną drogą;

(6) T nie zawiera cyklu, ale po dodaniu dowolnej krawędzi otrzymany graf ma dokładnie jeden cykl.

Dowód: Wykażemy twierdzenie indukcyjnie względem n. Jeśli $n = 1$, to twierdzenie jest prawdziwe. Załóżmy, że jest prawdziwe również dla wszystkich drzew T, których rząd jest mniejszy niż n. Następnie wykażemy równoważność warunków dla drzewa T o n wierzchołkach.

Drzewa

Dowód (cd.):

(1) $\Rightarrow$(2) [T jest drzewem $\Rightarrow$ T nie zawiera cykli i ma $n - 1$ krawędzi]

T jest drzewem, więc nie zawiera cykli. Po usunięciu dowolnej krawędzi e otrzymujemy $T - e = T_1 \cup T_2$. Z założenia indukcyjnego mamy $m(T_i) = n(T_i) - 1$, $i = 1,2$. Stąd

$$m(T) = n(T_1) - 1 + n(T_2) - 1 + 1 = n(T) - 1.$$

(2) $\Rightarrow$(3) [T nie zawiera cykli i ma $n - 1$ krawędzi $\Rightarrow$ T jest grafem spójnym i ma $n - 1$ krawędzi]

Załóżmy, że T nie jest grafem spójnym. Wtedy $T = T_1 \cup T_2$. To oznacza, że

$$m(T) = m(T_1) + m(T_2) = n(T_1) + n(T_2) - 2,$$

na mocy założenia indukcyjnego. Stąd $m(T) = n(T) - 2$. Ale z warunku (2) wynika, że $m(T) = n(T) - 1$, co daje sprzeczność.

Drzewa

Dowód (cd.):

(3) $\Rightarrow$(4) [T jest grafem spójnym i ma $n - 1$ krawędzi $\Rightarrow$ T jest grafem spójnym i każda krawędź jest mostem]

Graf $T - e$ nie jest spójny, gdyż podaliśmy wcześniej twierdzenie mówiące, iż dla grafu o k składowych spójności zachodzi $n - k \leq m$. W przypadku grafu $T - e$ mamy $k = 1$ oraz $m = n - 2$.

(4) $\Rightarrow$(5) [T jest grafem spójnym i każda krawędź jest mostem $\Rightarrow$ każde dwa wierzchołki T są połączone dokładnie jedną drogą]

Gdyby pomiędzy pewną parą wierzchołków istniały dwie drogi, to tworzyłyby one cykl. Usunięcie dowolnej krawędzi z tego cyklu nie spowoduje rozspojenia grafu, co przeczy założeniu, iż każda krawędź jest mostem.

Drzewa

Dowód (cd.):

(5) $\Rightarrow$(6) [każde dwa wierzchołki T są połączone dokładnie jedną drogą $\Rightarrow$ T nie zawiera cyklu, ale po dodaniu dowolnej krawędzi otrzymany graf ma dokładnie jeden cykl]

Załóżmy, że graf T zawiera cykl. Wówczas pewne dwa wierzchołki, należące do tego cyklu, są połączone dwiema różnymi drogami. Sprzeczność. Po operacji dodania krawędzi graf $T + \{u,v\}$ zawiera cykl, ponieważ wierzchołki u oraz v są połączone w grafie T drogą.

(6) $\Rightarrow$(1) [T nie zawiera cyklu, ale po dodaniu dowolnej krawędzi otrzymany graf ma dokładnie jeden cykl $\Rightarrow$ T jest drzewem]

Wystarczy wykazać, że graf T jest spójny. Gdyby tak nie było, to T graf $T + \{u,v\}$ nie zawiera cyklu, gdzie $T = T_1 \cup T_2$ oraz $u \in V(T_1)$, $v \in V(T_2)$. Sprzeczność.

Macierz sąsiedztwa

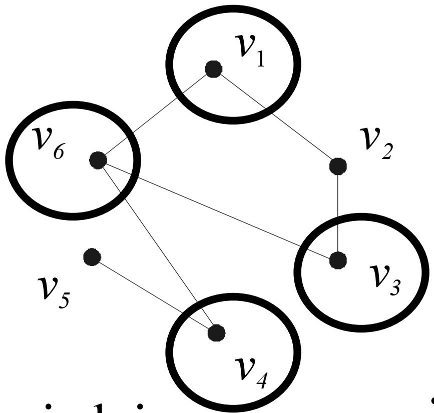
Macierz sąsiedztwa jest strukturą danych służącą do reprezentacji grafów w pamięci komputera. Dla danego grafu G określamy kwadratową tablicę M o wymiarach $n\times n$ taką, że

$$M[i,j]=\begin{cases}0, & \text{gdy } \{v_i,v_j\}\notin E(G)\\ 1, & \text{gdy } \{v_i,v_j\}\in E(G)\end{cases}$$

Przykład

Przykład: sąsiedztwo wierzchołka v_6

$G =$



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

Wierzchołek v_6 jest sąsiedni z: v_1, v_3 i v_4

Macierz sąsiedztwa

Zalety:

- sprawdzenie, czy $\{v_i, v_j\} \in E(G)$, dodanie oraz usunięcie krawędzi to operacje dokonywane w stałym czasie
- struktura danych łatwa w implementacji

Wady:

- przechowywanie macierzy wymaga O(n^2) pamięci
- przejrzenie zbioru krawędzi dokonywane w czasie O(n^2) zamiast O(m)

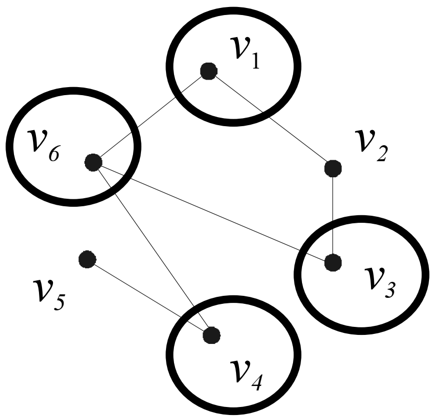
Lista sąsiedztwa

Lista sąsiedztwa jest strukturą danych, w której występuje n-elementowy wektor N zwany nagłówkiem taki, że $N[i]$ jest wskaźnikiem na listę zawierającą sąsiadów wierzchołka v_i.

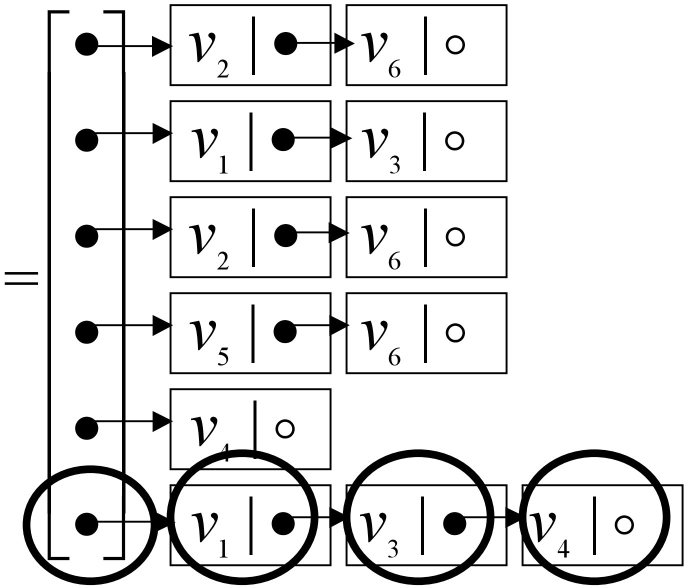
Przykład

Przykład: lista sąsiadów wierzchołka v_6

$G =$



$N =$



Wierzchołek v_6 jest sąsiedni z: v_1, v_3 oraz v_4

Lista sąsiedztwa

Zalety:

- przejrzenie zbioru krawędzi dokonywane w czasie O(m)
- oszczędność pamięci – wymagana pamięć rzędu O(m)

Wady:

- sprawdzenie, czy $\{v_i, v_j\} \in E(G)$ wymaga czasu proporcjonalnego do min{deg(v_i), deg(v_j)}
- usunięcie krawędzi $\{u, v\}$ wymaga czasu proporcjonalnego do max{ deg(u), deg(v) }

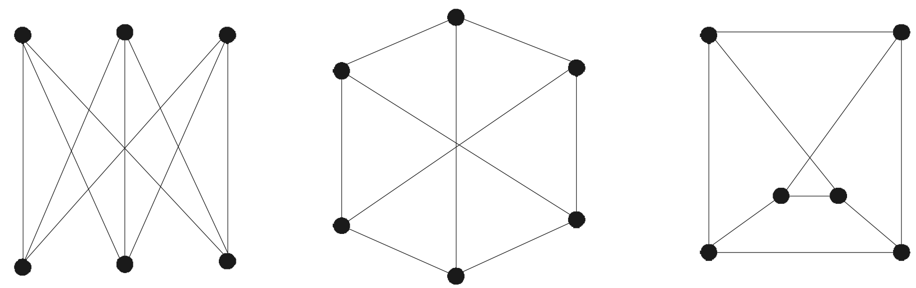
Izomorfizm grafów

Def. Niech będą dane dwa grafy G_1 i G_2 o tej samej liczbie wierzchołków. Powyższe grafy są izomorficzne o ile istnieje bijekcja $f{:}V(G_1)\to V(G_2)$ taka, że wierzchołki u i v są sąsiednie w grafie G_1 wtedy i tylko wtedy, gdy wierzchołki $f(u)$ i $f(v)$ są sąsiednie w G_2.

Tw. *Jeśli grafy G_1 i G_2 są izomorficzne, to*

1. $|V(G_1)| = |V(G_2)|$
2. $|E(G_1)| = |E(G_2)|$
3. *Ciągi stopniowe tych grafów są sobie równe*

Przykład Następujące trzy grafy są izomorficzne:



Izomorfizm grafów

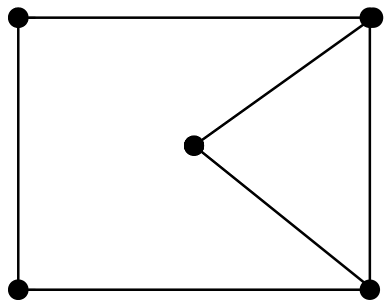
Uwaga. Warunki podane w powyższym twierdzeniu nie są wystarczające do tego, aby dwa grafy były izomorficzne.

Dla poniższych dwóch grafów G_1, G_2 mamy

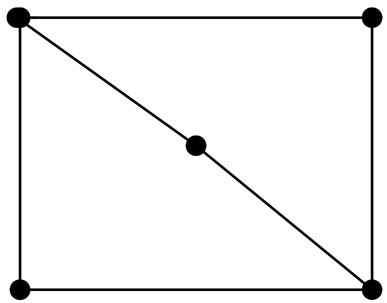
1. $|V(G_1)| = |V(G_2)|$
2. $|E(G_1)| = |E(G_2)|$
3. *Ciągi stopniowe tych grafów są sobie równe,*

lecz nie są one izomorficzne.

$G_1 =$



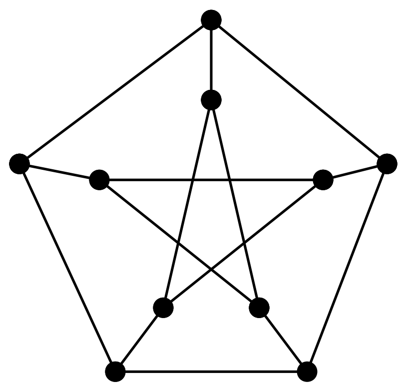
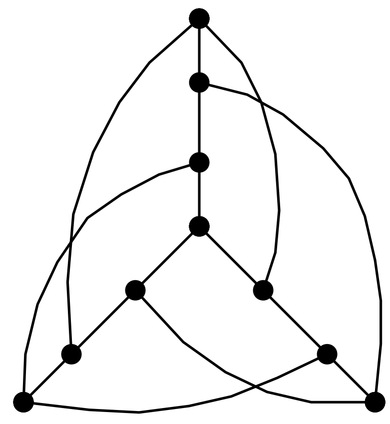
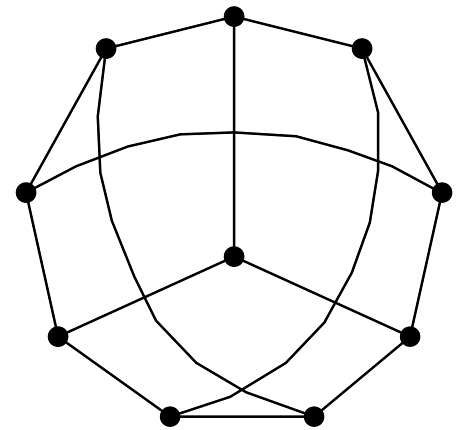
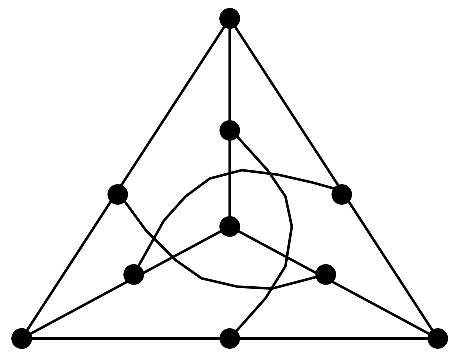
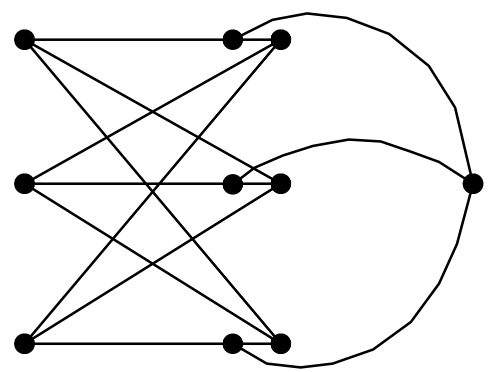
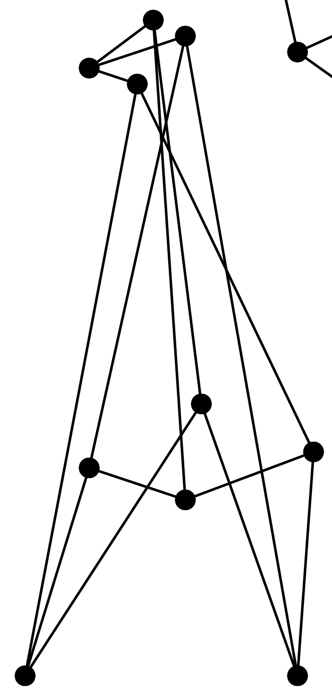
$G_2 =$



Uwaga Nie wiadomo, czy istnieje wielomianowy algorytm, stwierdzający, czy podane na wejściu dwa grafy są izomorficzne.

Izomorfizm grafów

Przykład Sześć postaci grafu Petersena

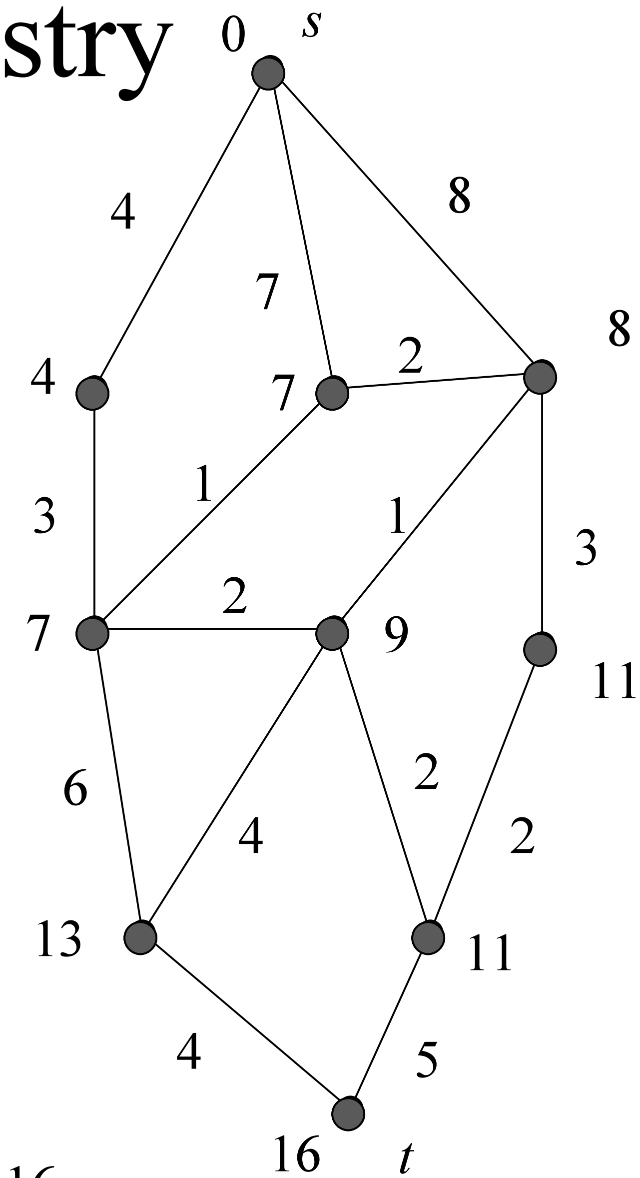
Szukanie najkrótszych dróg

Algorytm Dijkstry – znajduje najkrótszą drogę z wierzchołka s do dowolnego innego wierzchołka w grafie G. Zakładamy, że krawędź $\{u,v\}$ posiada etykietę $d_{u,v}$, będącą jej długością. Etykieta lab(v) jest szukaną długością drogi z s do v.

Procedure Dijkstra(G, s)
begin
    $S := \emptyset$; lab(s) = 0; lab(v) = $+\infty$ dla $v \neq s$;
    for $i := 1$ **to** $n - 1$ **do begin**
        znajdź $v \in V \setminus S$, posiadający minimalną etykietę lab(v);
        $S := S \cup \{v\}$;
        for każdy sąsiad $u \in V \setminus S$ **do begin**
            lab(v) := min{ lab(u), lab(v) + $d_{u,v}$ };
        end
    end
end

Algorytm Dijkstry

Procedure Dijkstra(G, s)
begin
$S := \emptyset$; lab(s) = 0; lab(v) = $+\infty$ dla $v \neq s$;
for i := 1 **to** n – 1 **do begin**
    znajdź $v \in V \backslash S$, o minimalnym lab(v);
    $S := S \cup \{v\}$;
    for każdy sąsiad $u \in V \backslash S$ **do begin**
        lab(v) := min{ lab(u), lab(v) + $d_{u,v}$ };
    end
  end
end



Szukamy długości najkrótszej drogi z s do v.

Elementy zbioru S oznaczamy kolorem czerwonym.

Odpowiedź: najkrótsza droga z s do t ma długość 16.

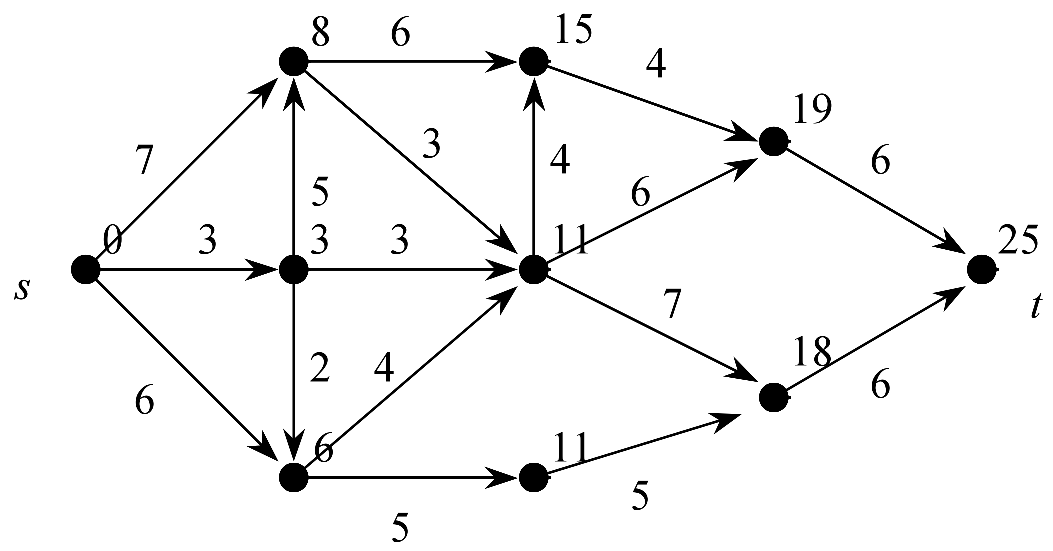
Szukanie najdłuższej drogi

D – acykliczny digraf z jednym źródłem s. Rozwiązujemy problem znalezienia najdłuższej drogi z s do pewnego wierzchołka t. Długość (waga) łuku (u,v) jest oznaczana symbolem $d_{(u,v)}$

Procedure NajdłuższaDroga(G, s, t)
begin
  lab(s) = 0;
  while wierzchołek t nie jest oznaczony **do begin**
    znajdź $v \in V \backslash S$ osiągalny wyłącznie z zaetykietowanych wierzchołków;
    lab(v) := max{ lab(u) + $d_{(u,v)}$: $(u,v) \in E(D)$ };
  end
end

Szukanie najdłuższej drogi

Procedure NajdłuższaDroga(G, s, t)
begin
  lab(s) = 0;
  while wierzchołek t nie jest oznaczony **do begin**
    znajdź $v \in V \setminus S$ osiągalny wyłącznie z zaetykietowanych wierzchołków;
    lab(v) := max{ lab(u) + $d_{(u,v)}$: $(u,v) \in E(D)$ };
  end
end



Drzewa spinające

Def. Niech G będzie grafem spójnym o n wierzchołkach. *Drzewo spinające* grafu G to dowolny jego n-wierzchołkowy podgraf będący drzewem. Jeśli G jest niespójny, to powyższy podgraf nazywamy *lasem spinającym*.

Def. *Liczba cyklomatyczna* grafu G to liczba krawędzi, których usunięcie z G prowadzi do utworzenia drzewa (lasu) spinającego i jest oznaczana symbolem $\gamma(G)$.

Przykłady

$\gamma(T) = 0$, T – drzewo,

$\gamma(L) = 0$, L – las,

$\gamma(U) = 1$, U – graf jednocykliczny,

$\gamma(K_n) = n\,(n - 1)/2 - n + 1$,

$\gamma(W_n) = n$.

Drzewa spinające

Tw. *Dla dowolnego grafu G o k składowych spójności zachodzi*

$$\gamma(G) = m - n + k.$$

Dowód: Rozważmy najpierw przypadek, gdy G jest spójny. Wiadomo, że n wierzchołkowe drzewo ma $n - 1$ krawędzi. Aby w grafie G znaleźć podgraf będący drzewem spinającym, usuwamy wszystkie krawędzie, z wyjątkiem tych $n - 1$, co oznacza, że musimy usunąć $m - n + 1$ krawędzi.

Załóżmy teraz, że G nie jest spójny. Wówczas $G = G_1 \cup \ldots \cup G_k$, dla pewnego $k > 1$. Możemy założyć, że każdy podgraf G_i jest spójny. Udowodniliśmy powyżej, że $\gamma(G_i) = m_i - n_i + 1$, dla każdego $i = 1, \ldots, k$, gdzie $n_i = |V(G_i)|$, $m_i = |E(G_i)|$. Dodając równania stronami otrzymujemy:

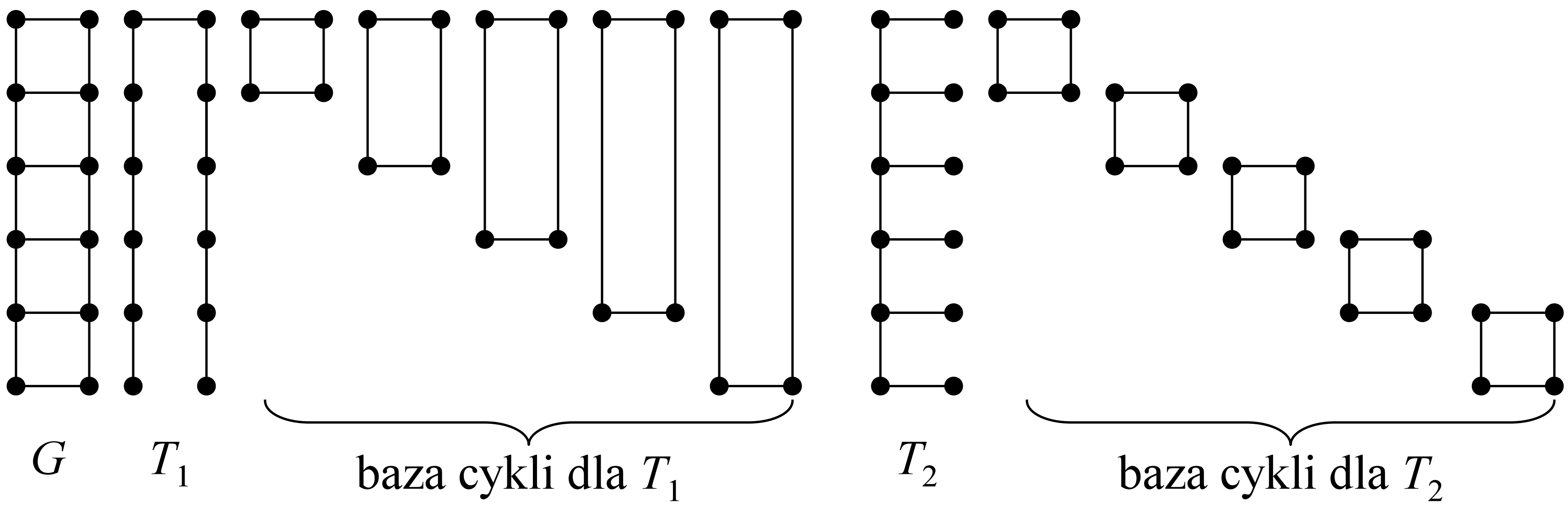
$$\gamma(G_1) + \ldots + \gamma(G_k) = (m_1 + \ldots + m_k) - (n_1 + \ldots + n_k) + k = m - n + k,$$

co kończy dowód, gdyż $\gamma(G) = \gamma(G_1) + \ldots + \gamma(G_k)$.

Baza cykli

Def Niech T będzie lasem spinającym grafu G. *Fundamentalny zbiór cykli* (*baza cykli*) jest to zbiór $\gamma(G)$ różnych cykli grafu G taki, że każdy cykl do niego należący powstał przez dodanie pewnej krawędzi ze zbioru $E(G) \setminus E(T)$.

Przykład Dwa fundamentalne zbiory cykli pewnego grafu, uzyskane za pomocą różnych drzew spinających T_1 i T_2.



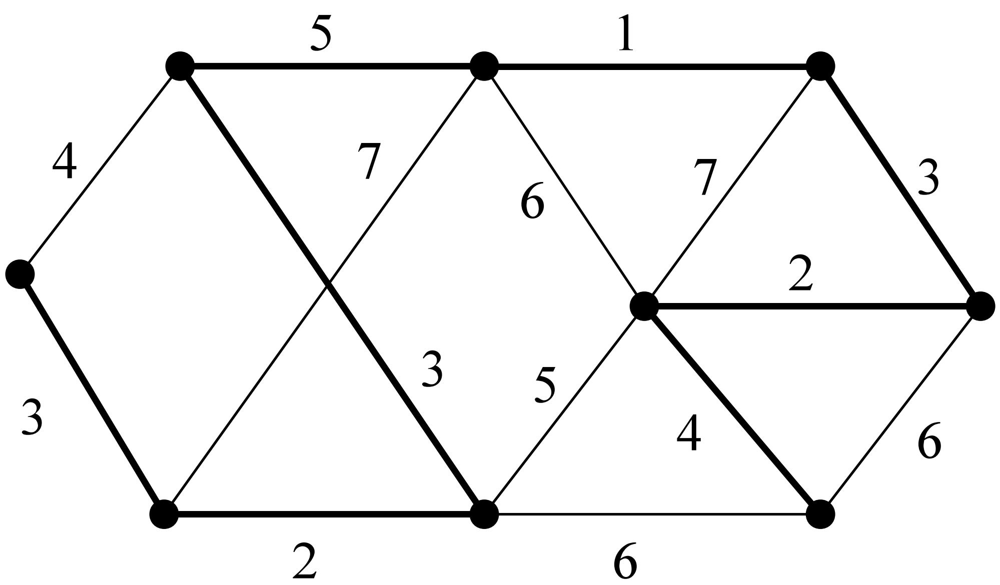
Minimalne drzewa spinające

Niech będzie dany graf G. Zakładamy, że z każdą krawędzią $e_i \in E(G)$ jest skojarzona pewna liczba nieujemna, będąca wagą tej krawędzi. Rozważamy problem szukania drzewa spinającego którego suma wag krawędzi jest minimalna. Przykładem rozwiązania dla tego problem jest algorytm Prima o złożoności $O(n^2)$.

Procedure Prim(G, s)
begin
  $T = (\{v_i, v_j\}, \{\{v_i, v_j\}\})$, gdzie $\{v_i, v_j\}$ jest krawędzią o najmniejszej wadze;
  while $n(T) < n(G)$ **do begin**
    znajdź najkrótszą krawędź $\{u,v\}$ taką, że $u \in V(T)$ oraz $v \notin V(T)$;
    $T := T \cup \{u,v\}$;
  end
end

Minimalne drzewa spinające

Procedure Prim(G, s)
begin
  $T = (\ \{ v_i,v_j\}, \{\{v_i,v_j\}\}\ )$, gdzie $\{v_i,v_j\}$ jest krawędzią o
    najmniejszej wadze;
  while $n(T) < n(G)$ **do begin**
    znajdź najkrótszą krawędź $\{u,v\}$ taką, że $u \in V(T)$ oraz $v \notin V(T)$;
    $T := T \cup \{u,v\}$;
  end
end



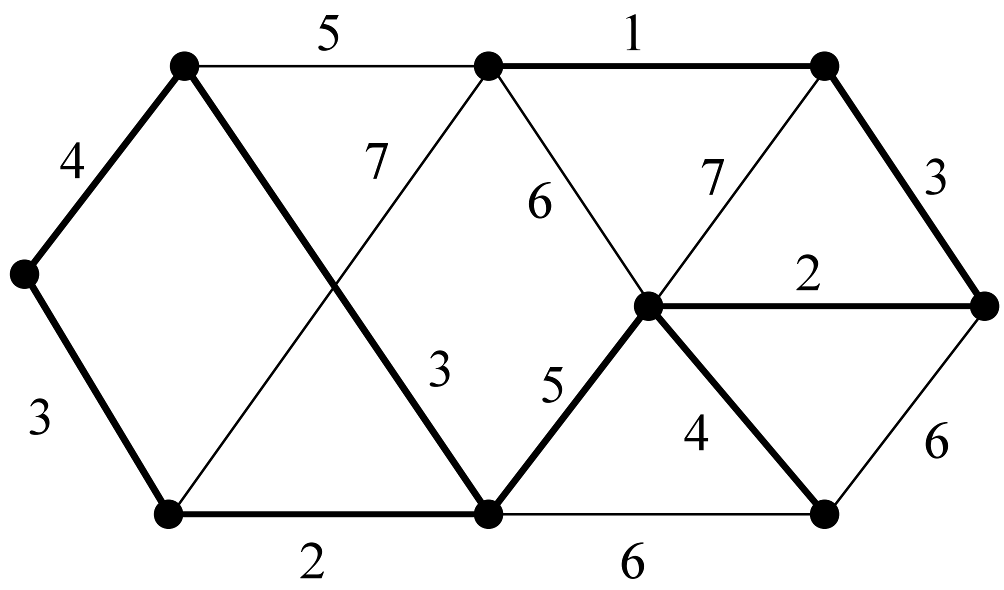
Minimalne drzewa spinające

Algorytm Kruskala jest drugim przykładem efektywnego szukania minimalnego drzewa spinającego. Złożoność tego algorytmu to $O(m \log(m))$.

Procedure Kruskal(G, s)
begin
  posortuj krawędzie $e_1, \ldots, e_m$ niemalejąco wg wag;
  $T = \varnothing$;
  $i := 1$; $j := 1$;
  repeat
    if e_i nie tworzy cyklu w T **then begin**
      $T := T \cup \{e_i\}$;
      $j := j + 1$;
    end;
    $i := i + 1$;
  until $j = n - 1$
end

Minimalne drzewa spinające

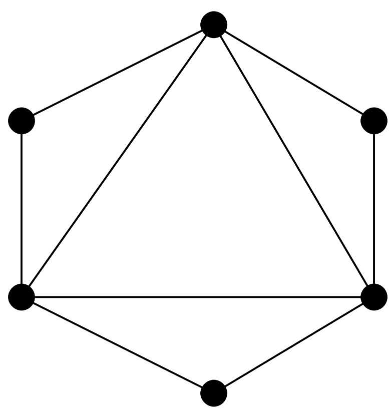
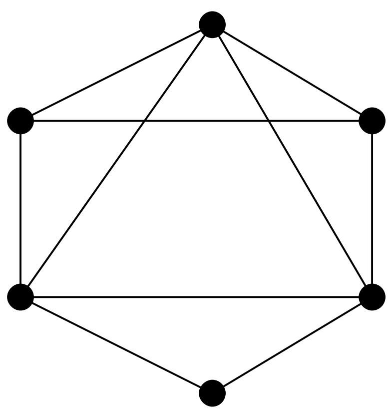
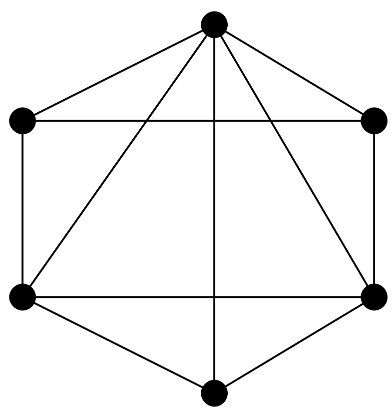
Procedure Kruskal(G, s)
begin
  posortuj krawędzie $e_1, \dots, e_m$ niemalejąco wg wag;
  $T = \emptyset$;
  $i := 1;\ j := 1$;
  repeat
    if e_i nie tworzy cyklu w T **then begin**
      $T := T \cup \{e_i\}$;
      $j := j + 1$;
    end;
    $i := i + 1$;
  until $j = n - 1$
end



Grafy eulerowskie

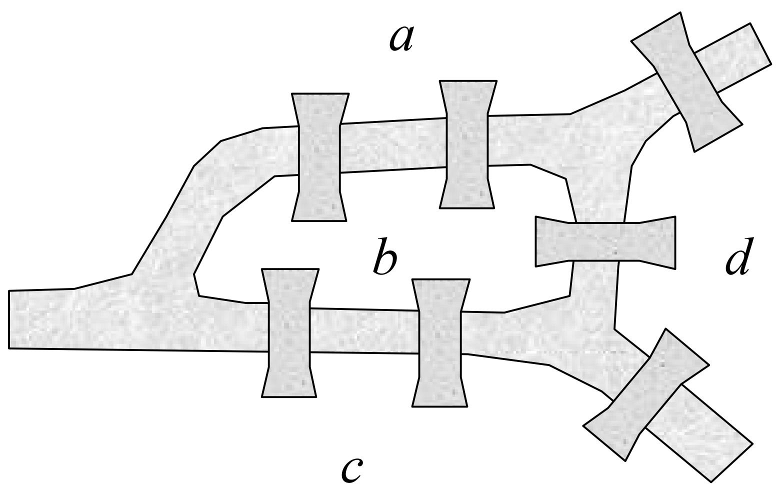
Def. Graf jest *grafem eulerowskim,* jeśli zawiera cykl zawierający wszystkie krawędzie. Graf jest *półeulerowski,* gdy posiada łańcuch o powyższej własności.

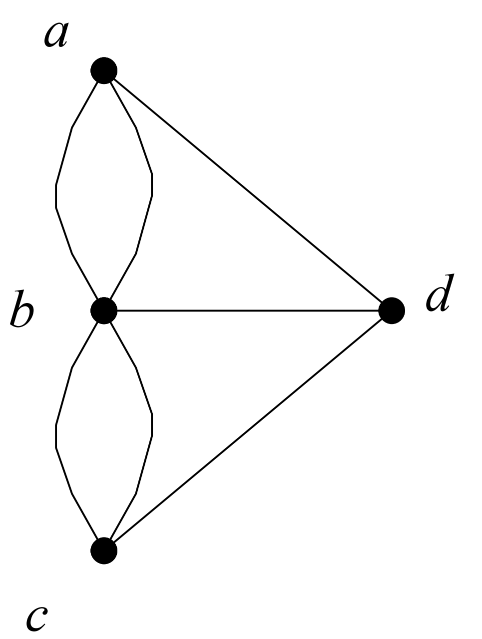
Przykład Poniżej podano reprezentacje trzech grafów, z których pierwszy jest eulerowski, drugi – półeulerowski, natomiast ostatni nie jest ani eulerowski ani półeulerowski.

Grafy eulerowskie

Nazwa „eulerowski” pochodzi stąd, iż Euler w 1736 r. rozwiązał „problem mostów królewieckich”. Pytano, czy można przejść dokładnie raz przez każdy z siedmiu mostów (rys. po lewej) tak, aby powrócić do punktu wyjścia. Zauważmy, że problem jest równoważny stwierdzeniu, czy graf pokazany na rys. po prawej stronie jest eulerowski.





Grafy eulerowskie

Aby podać warunek konieczny i dostateczny na to, aby dany graf był eulerowski będzie potrzebny poniższy lemat:

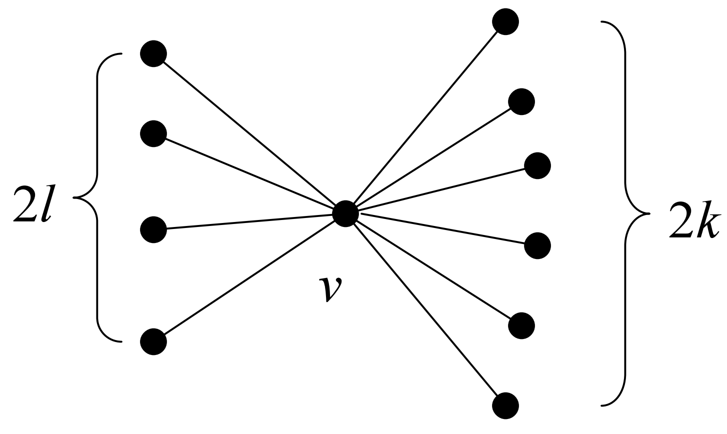
Lemat *Jeśli każdy wierzchołek grafu G ma stopień równy co najmniej* 2*, to G zawiera cykl*.

Dowód: Jeśli G zawiera pętle lub krawędzie wielokrotne, to lemat jest spełniony, więc zakładamy dalej, że G jest grafem prostym. Następnie konstruujemy ścieżkę: wybieramy dowolny wierzchołek $v_1 \in V(G)$, po czym dodajemy kolejne według zasady: jeśli v_i jest ostatnio dodanym wierzchołkiem, to jeśli v_i jest sąsiedni z v_j poprzednio dodanym do ścieżki, to otrzymujemy cykl zawierający $v_j,\ldots,v_i$ co kończy dowód. W przeciwnym wypadku wybieramy dowolnego sąsiada wierzchołka v_i (istnieje, ponieważ $\deg(v_i)>1$) i dodajemy go do ścieżki. Graf G jest skończony więc w pewnym kroku otrzymamy wierzchołek v_k, który jest sąsiedni z pewnym v_l, gdzie $l<k$.

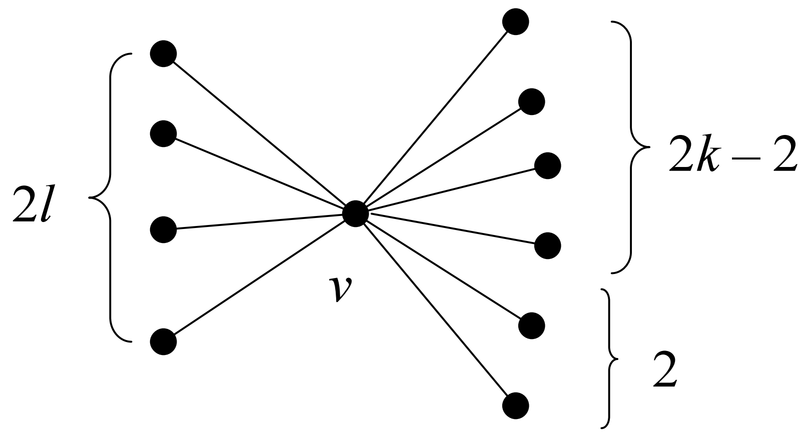
Grafy eulerowskie

Tw. (Euler, 1736) *Graf spójny jest eulerowski wtedy i tylko wtedy, gdy stopień każdego wierzchołka jest liczbą parzystą.*

Dowód: ($\Rightarrow$) Jeśli C jest cyklem Eulera w G, to trawersujemy i usuwamy krawędzie w $E(G)$ zgodnie z kolejnością zadaną przez C. Gdy przechodzimy przez dowolny wierzchołek v, to usuwamy dwie incydentne z nim krawędzie, więc stopień tego wierzchołka w tak zredukowanym grafie pozostaje parzysty. Na rys. poniżej kolorem czerwonym oznaczono krawędzie, które zostały usunięte z G.



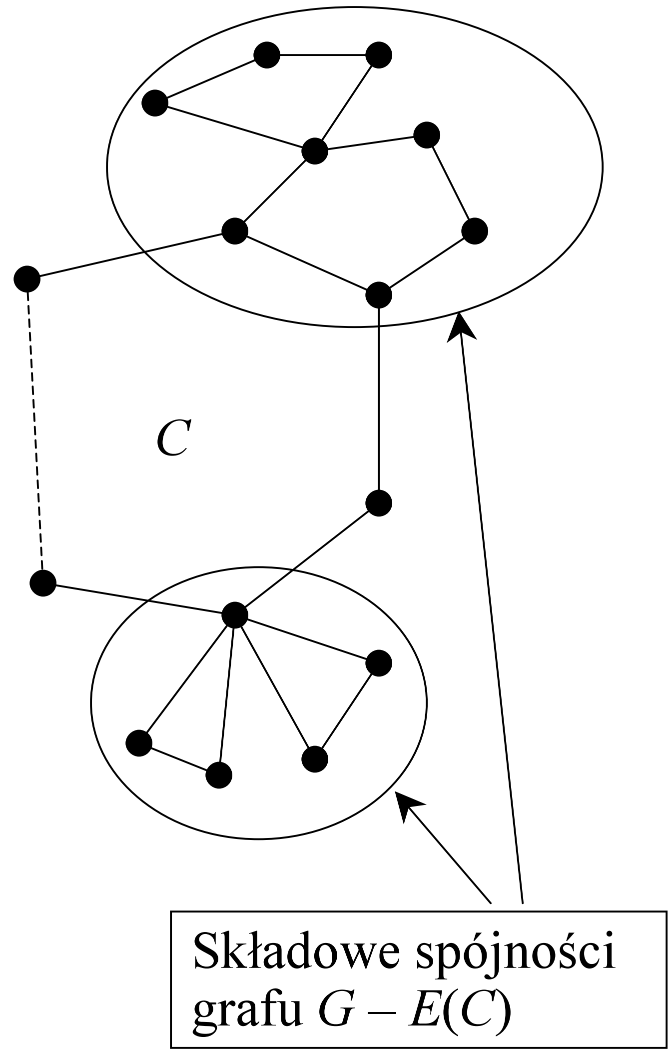
(przed przejściem przez v)



(po przejściu przez v)

Grafy eulerowskie

Dowód tw. Eulera (⇐) G jest spójny, więc dla każdego v zachodzi deg(v)>1 więc z poprzedniego lematu wiadomo, że G zawiera cykl, który oznaczmy przez C. Twierdzenie dowodzimy przez indukcję względem m. Jeśli $C = G$, to dowód jest zakończony. W przeciwnym wypadku każda składowa spójności grafu $G - E(C)$ spełnia założenie twierdzenia, więc z założenia indukcyjnego jest eulerowska. Znajdujemy cykl Eulera w G następująco: przechodzimy przez kolejne wierzchołki w C i jeśli bieżący wierzchołek należy do pewnej składowej spójności, to trawersujemy cykl Eulera w tej składowej, powracając do tego samego wierzchołka w C.



Grafy eulerowskie

Wniosek *Graf spójny jest eulerowski wtedy i tylko wtedy, gdy zbiór jego krawędzi można podzielić na rozłączne cykle.*

Wniosek *Graf spójny jest półeulerowski, gdy posiada co najwyżej dwa wierzchołki nieparzystego stopnia. Jeden z nich jest początkiem, a drugi końcem łańcucha Eulera.*

Algorytm Fleury'ego (znajdujący cykl Eulera)

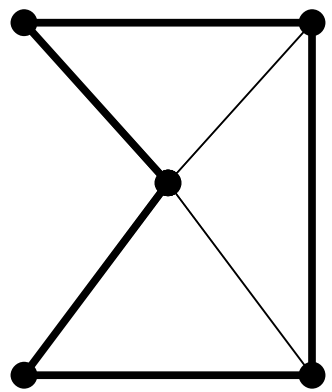
• rozpocznij wędrówkę w dowolnym wierzchołku

• usuwaj strawersowane krawędzie, przechodząc po moście jedynie w ostateczności (sytuacja, w której są do wyboru co najmniej dwa mosty oznacza, że graf nie jest eulerowski).

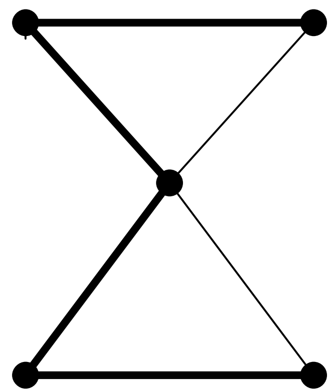
Grafy hamiltonowskie

Def. *Cykl* (*droga*) *Hamiltona* jest to cykl (droga), w którym każdy wierzchołek grafu występuje dokładnie raz. Graf jest *hamiltonowski* (*półhamiltonowski*), o ile posiada cykl (drogę) Hamiltona.

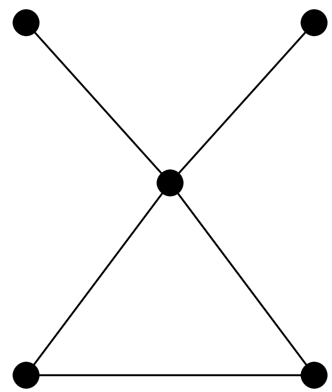
Przykład



Graf hamiltonowski



Graf półhamiltonowski

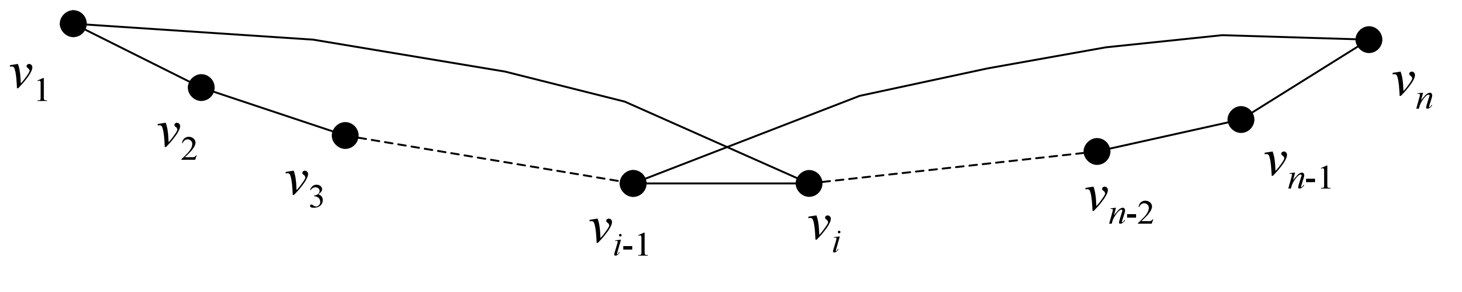


Graf nie jest ani hamiltonowski ani półhamiltonowski

Grafy hamiltonowskie

Tw. (Ore, 1960) *Jeśli G jest grafem prostym o* $n \geq 3$ *wierzchołkach i* $\deg(u) + \deg(v) \geq n$ *dla każdej pary niesąsiednich wierzchołków u i v, to graf G jest hamiltonowski*.

Dowód: Załóżmy, że istnieje graf G o podanych założeniach ale nie jest hamiltonowski. Możemy założyć, że G posiada drogę Hamiltona $v_1 \to v_2 \to \ldots \to v_n$ oraz $\{v_1, v_n\} \notin E(G)$. Stąd wynika, że $\deg(v_1) + \deg(v_n) \geq n$ a to oznacza, że istnieje indeks i taki, że $\{v_1, v_i\} \in E(G)$ oraz $\{v_{i-1}, v_n\} \in E(G)$, co pokazano na rysunku. To prowadzi do sprzeczności, gdyż $v_1 \to v_2 \to \ldots \to v_{i-1} \to v_n \to v_{n-1} \to \ldots \to v_i \to v_1$ jest cyklem Hamiltona.



Grafy hamiltonowskie

Wniosek (Dirac, 1952) *Jeśli G jest grafem prostym o* $n \geq 3$ *wierzchołkach i* $\deg(u) \geq n/2$ *dla każdego wierzchołka v, to G jest hamiltonowski*.

Dowód: Wynika z poprzedniego twierdzenia, gdyż $\deg(u) + \deg(v) \geq n$ dla każdej pary (również niesąsiednich) wierzchołków.

Uwaga Problem polegający na stwierdzeniu czy dany graf G jest hamiltonowski jest NP-zupełny. Oznacza to, że nie są znane efektywne (działające w czasie wielomianowym) algorytmy rozwiązujące ten problem. Nie jest również znane twierdzenie podające warunki konieczne i dostateczne na to, aby G był hamiltonowski.

Problem chińskiego listonosza

Dana jest sieć ulic oraz poczta. Aby listonosz dostarczył korespondencję musi przejść wzdłuż każdej ulicy co najmniej raz i powrócić do punktu wyjścia. Formułując problem w języku grafów, pytamy o najkrótszą zamkniętą marszrutę w grafie G utworzonym na podstawie sieci ulic, w którym wagi krawędzi odpowiadają długościom ulic.

Znany jest efektywny algorytm rozwiązujący ten problem, lecz my rozważymy dwa przypadki:

Przypadek 1: graf G jest eulerowski. Wówczas każdy cykl Eulera jest optymalnym rozwiązaniem, które można znaleźć korzystając np. z algorytmu Fleury'ego.

Problem chińskiego listonosza

Przypadek 2: graf *G* jest półeulerowski. Znajdujemy ścieżkę Eulera łączącą dwa wierzchołki nieparzystego stopnia *u* i *v*. Następnie szukamy najkrótszej drogi z *u* do *v*. Łącząc obie drogi otrzymujemy rozwiązanie.

Przykład

1) Droga Eulera:

$b \to a \to f \to b \to c \to$

$d \to e \to c \to f \to e$

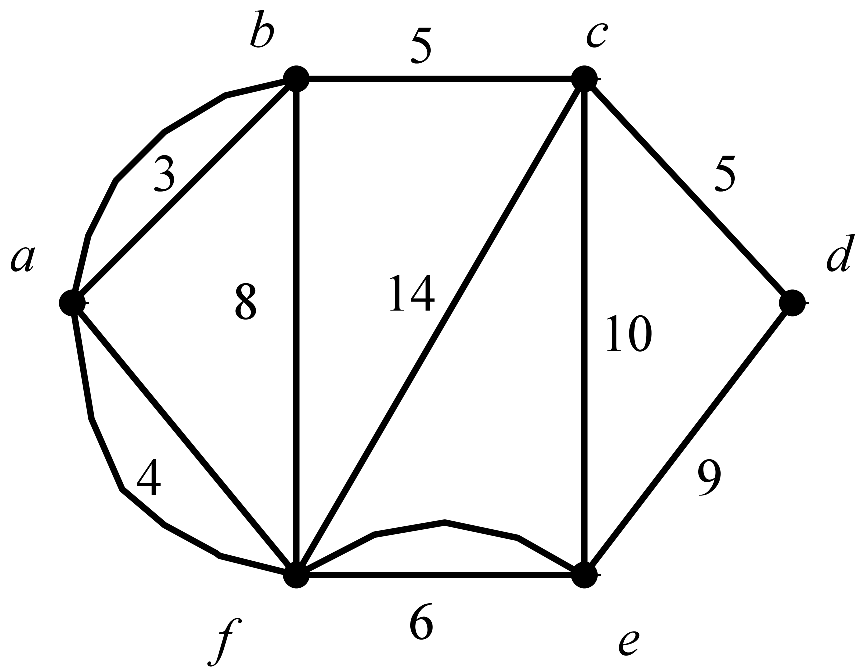
2) Najkrótsza droga z *e* do *b*

$e \to f \to a \to b$

3) „Trasa listonosza”:

$b \to a \to f \to b \to c \to$

$d \to e \to c \to f \to e \to f \to a \to b$

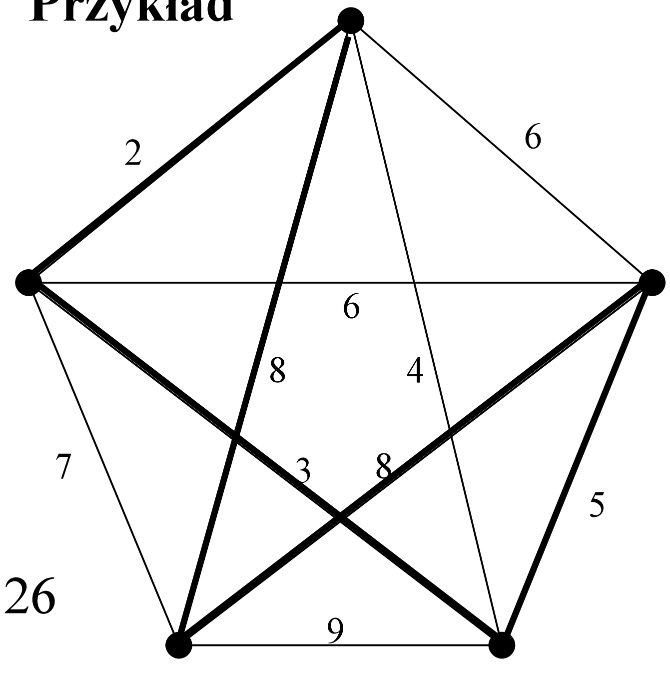


Problem komiwojażera

Dany jest zbiór miast. Komiwojażer chce odwiedzić wszystkie miasta (każde dokładnie raz) i powrócić do punktu wyjścia. Problem polega na znalezieniu najkrótszej trasy o tej własności.

Zdefiniujemy powyższy problem w języku teorii grafów. Niech będzie dany graf pełny G. Zakładamy, że z każdą krawędzią e_i jest skojarzona jej waga (długość) oznaczana dalej przez w_i. Rozwiązaniem problemu komiwojażera jest taki cykl Hamiltona, którego suma wag krawędzi jest minimalna.

Przykład



$\Sigma = 26$

Problem komiwojażera

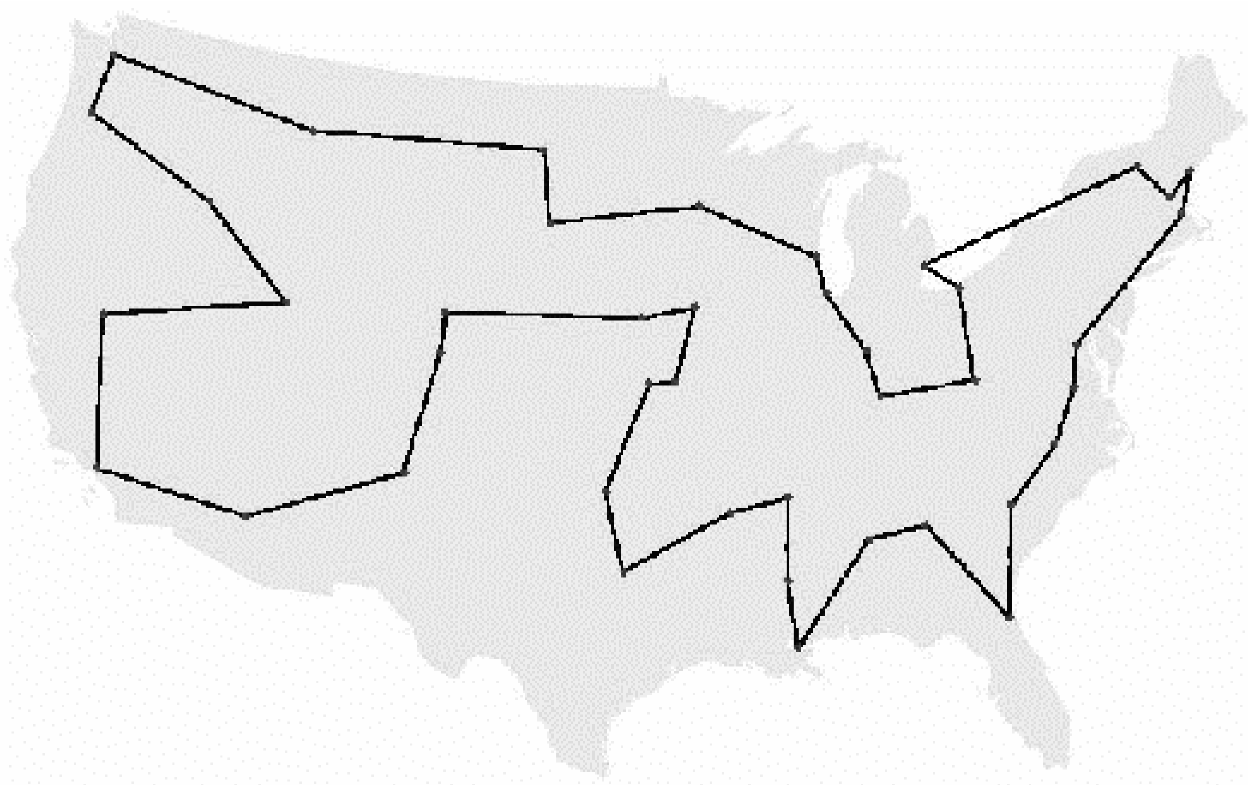
Uwagi

- problem komiwojażera jest NP-trudny, co oznacza, że nie są znane algorytmy o wielomianowej złożoności obliczeniowej rozwiązujące ten problem (przypuszczalnie takie nie istnieją)
- w praktyce jesteśmy zmuszeni posługiwać się wielomianowymi algorytmami przybliżonymi, tzn. takimi, które szybko znajdują rozwiązanie, które jest w przybliżeniu równe optymalnemu

Przykład Jednym z możliwych algorytmów dokładnych jest sprawdzenie wszystkich możliwych cykli Hamiltona i wybranie najkrótszego. Wadą takiego podejścia jest to, że liczba cykli jest zbyt duża, gdyż dla n-wierzchołkowego grafu wynosi $(n!)/2$. Stąd, jeśli dysponujemy komputerem sprawdzającym milion permutacji na sekundę, to:

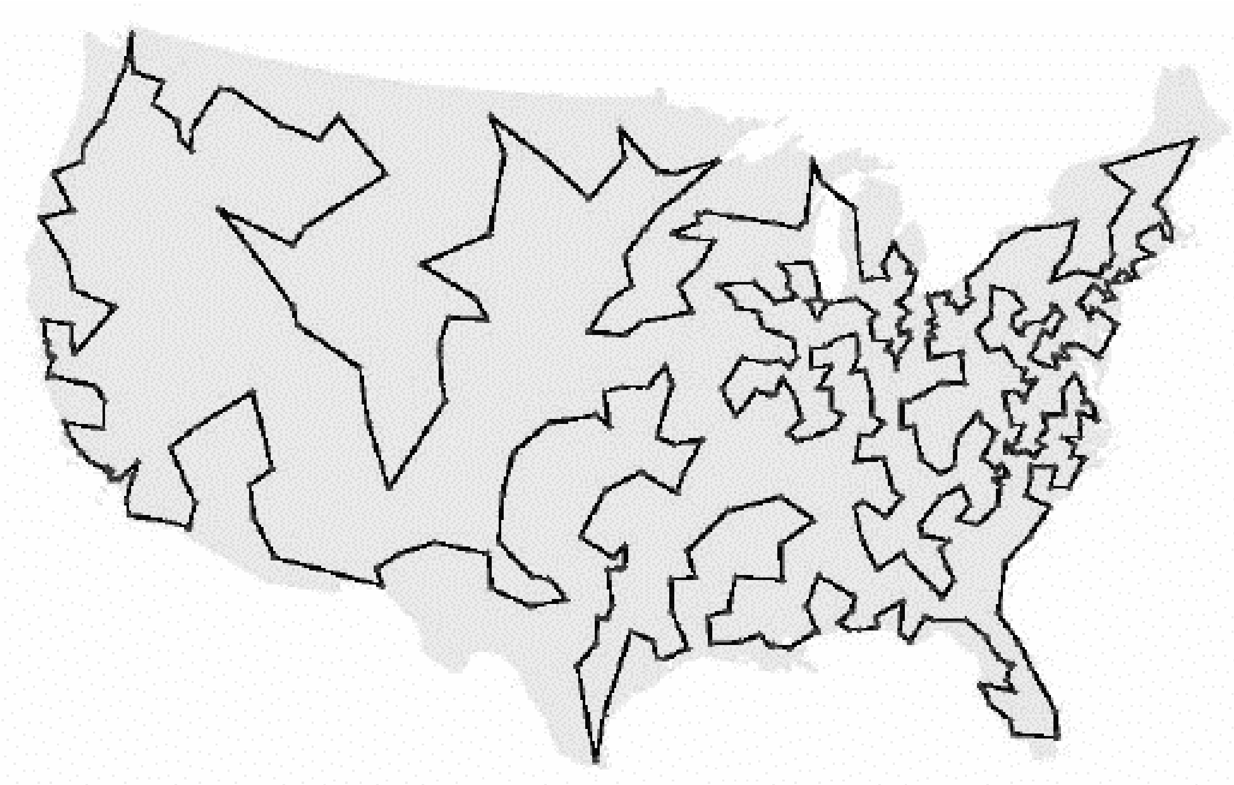
- $n = 10 \quad \Rightarrow \quad$ ilość cykli $= (10!)/2 = 1814400 \quad \Rightarrow \quad$ czas obliczeń $= 1.8$ s
- $n = 20 \quad \Rightarrow \quad$ ilość cykli $= (20!)/2 \approx 10^{18} \quad \Rightarrow \quad$ czas obliczeń ≈ 40 tys. lat

Historia problemu komiwojażera



George Dantzig, Ray Fulkerson i Selmer Johnson (1954) zaprezentowali optymalne rozwiązanie problemu komiwojażera dla 49 amerykańskich miast.

Historia problemu komiwojażera



Padberg i Rinaldi (1987) obliczyli optymalne rozwiązanie dla 532 punktów

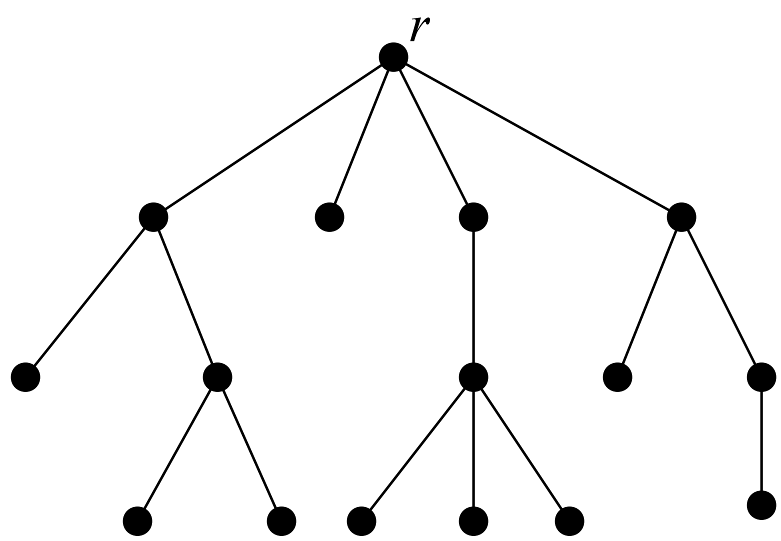
Historia problemu komiwojażera

Rozwiązanie obejmujące 13549 miast amerykańskich, uzyskane w 1998 roku.

Algorytm *DFS*

Procedure *DFS*(*G*, *v*)
begin
  odwiedź wierzchołek *v*;
  for każdy nieodwiedzony sąsiad *u* wierzchołka *v* **do**
    DFS(*G,u*);
end

Przykład Działanie procedury w przypadku drzewa (wywołanie *DFS*(*T,r*))



Algorytm 2-przyblizony

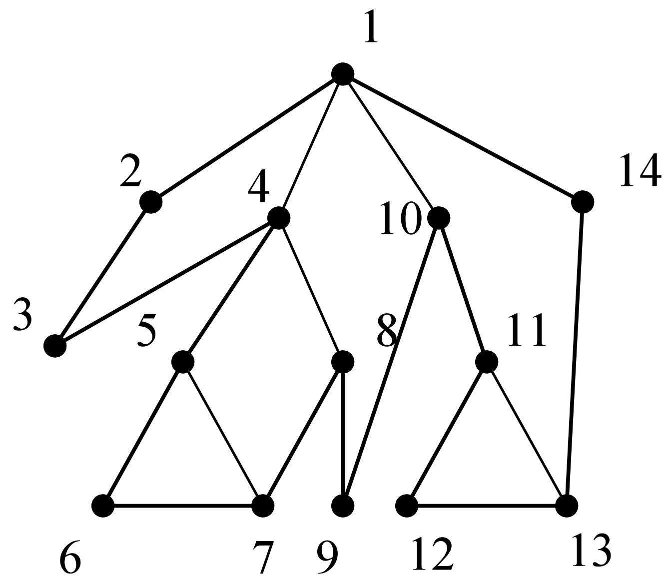
Procedure $A(\ G\ )$
begin
znajdź minimalne drzewo spinające T grafu G;
$DFS(T,v)$, gdzie v jest dowolnym wierzchołkiem drzewa $T;$
jeśli $v_1,\ldots,v_n$ są kolejnymi odwiedzonymi wierzchołkami, to utwórz cykl Hamiltona następująco: $v_1 \to v_2 \to \ldots \to v_{n-1} \to v_\mathrm{n} \to v_1$.
return C;
end

Przykład

1) Załóżmy, że znaleziono następujące drzewo spinające pewnego grafu pełnego G

2) kolejność odwiedzania wierzchołków przez DFS

3) wyznaczamy cykl Hamiltona



Algorytm 2-przyblizony

Oznaczenia: $W(H) = \Sigma_{e \in E(H)}\, w(e)$, gdzie H jest podgrafem G.
C_o – najkrótszy cykl Hamiltona w grafie G.
T_o – najlżejsze drzewo spinające grafu G.

Tw. Dla algorytmu A zachodzi $W(A(C))/W(C_o)<2$.

Dowód: Mamy oszacowanie: $W(T_o) < W(C_o)$.

Z nierówności trójkąta wynika, że $W(A(C)) \leq 2W(T_o)$.

Stąd

$$\frac{W(A(C))}{W(C_o)} \leq \frac{2W(T_o)}{W(C_o)} < \frac{2W(C_o)}{W(C_o)} = 2$$

Algorytm Christofidesa

Jest to przykład suboptymalnego algorytmu dla problemu komiwojażera.

- jak poprzednio, G jest grafem pełnym z obciążonymi krawędziami
- czas działania algorytmu to $O(n^3)$
- długość znalezionego cyklu (suma wag jego krawędzi) jest co najwyżej 1.5 razy dłuższa od długości najkrótszego cyklu

Procedure Christofides(G)
begin
  znajdź minimalne drzewo spinające T_o grafu G;
  znajdź zbiór V^{odd} węzłów nieparzystego stopnia w drzewie T_o;
  znajdź w V^{odd} minimalne skojarzenia dokładne M_o^{odd};
  znajdź cykl Eulera w podgrafie indukowanym przez $T_o + M_o^{odd}$;
  przekształć cykl Eulera w cykl Hamiltona C_{ch} w grafie pełnym;
end

Oszacowanie dolne

Poniżej podamy pewne oszacowania dotyczące długości optymalnej trasy komiwojażera, które będą potrzebne podczas analizy algorytmu.

Oznaczenia: C_o – najkrótszy cykl Hamiltona w obciążonym grafie G
T_o – minimalne drzewo spinające grafu G
$W(H) = \Sigma_{e \in E(H)} w(e)$, gdzie H jest podgrafem G.

Lemat $W(T_o) < W(C_o)$

Dowód: Niech e będzie dowolną krawędzią cyklu C_o. Podgraf $C_o - e$ jest ścieżką oraz jest również pewnym drzewem spinającym grafu G. Stąd otrzymujemy :

$$W(C_o) > W(C_o - e) \geq W(T_o).$$

Oszacowanie dolne

Oznaczenia: C_o – najkrótszy cykl Hamiltona w obciążonym grafie G
M_o – minimalne skojarzenia dokładne grafu G
$W(H) = \Sigma_{e \in E(H)}\, w(e)$, gdzie H jest podgrafem G.

Lemat $W(M_o) \leq W(C_o)/2$

Dowód: Niech M_1 oraz M_2 będą podzbiorami krawędzi cyklu C_o trawersowanymi w odpowiednio parzystych i nieparzystych krokach. Bez utraty ogólności możemy założyć, że $W(M_1) \leq W(M_2)$. Z równości

$$W(M_1) + W(M_2) = W(C_o)$$

$$2W(M_1) \leq W(M_1) + W(M_2)$$

wynika, że

$$W(M_1) \leq W(C_o)/2.$$

Po uwzględnieniu, że $W(M_o) \leq W(M_1)$ otrzymujemy tezę.

Algorytm Christofidesa

Oznaczenia: T_o – najlżejsze drzewo spinające grafu G
V_o^{odd} – wierzchołki o nieparzystym stopniu w T
M_o^{odd} – minimalne skojarzenia dokładne w V_o^{odd}
C_o^{odd} – najkrótszy cykl Hamiltona w grafie induk. przez V_o^{odd}
C_o – najkrótszy cykl Hamiltona w obciążonym grafie G
C_{ch} – cykl znaleziony przez alg. Christofidesa
C_E – cykl Eulera w grafie ($V(G)$, $E(T_o) \cup E(M_o^{odd})$)

Lemat *Jeśli G spełnia nierówność trójkąta, to* $W(C_{ch}) \leq 3W(C_o)/2$

Dowód: W poprzednich lematach wykazaliśmy, że

$W(T_o) < W(C_o)$ oraz $W(M_o^{odd}) \leq W(C_o^{odd})/2$.

Z nierówności $W(C_o^{odd}) \leq W(C_o)$ wynika, że $W(M_o^{odd}) \leq W(C_o)/2$.

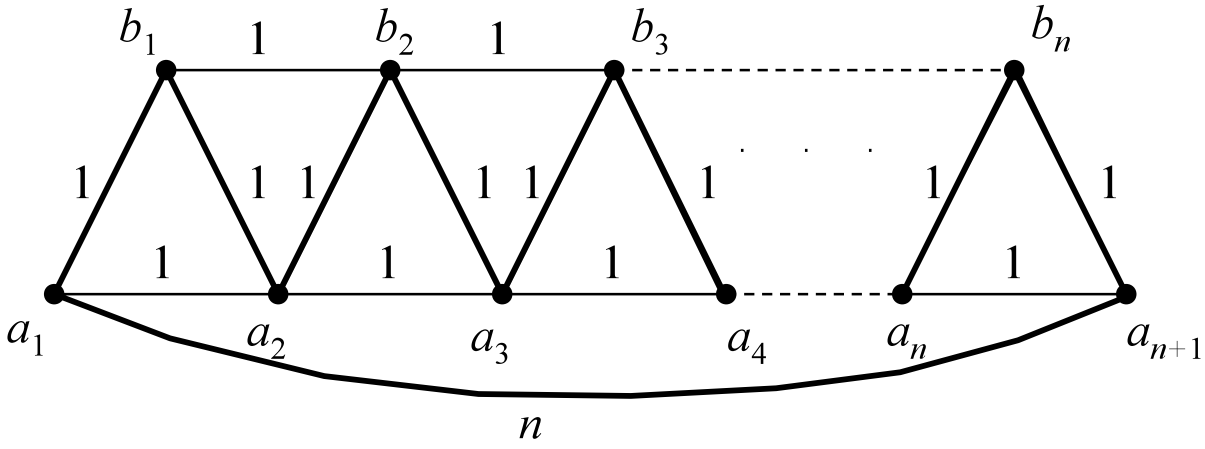
Z faktu, że G spełnia nierówność trójkąta wynika, że $W(C_{ch}) \leq W(C_E)$.

Łącząc powyższe nierówności otrzymujemy:

$$W(C_{ch}) \leq W(C_E) = W(T_o) + W(M_o^{odd}) < W(C_o) + W(C_o)/2 = 3W(C_o)/2.$$

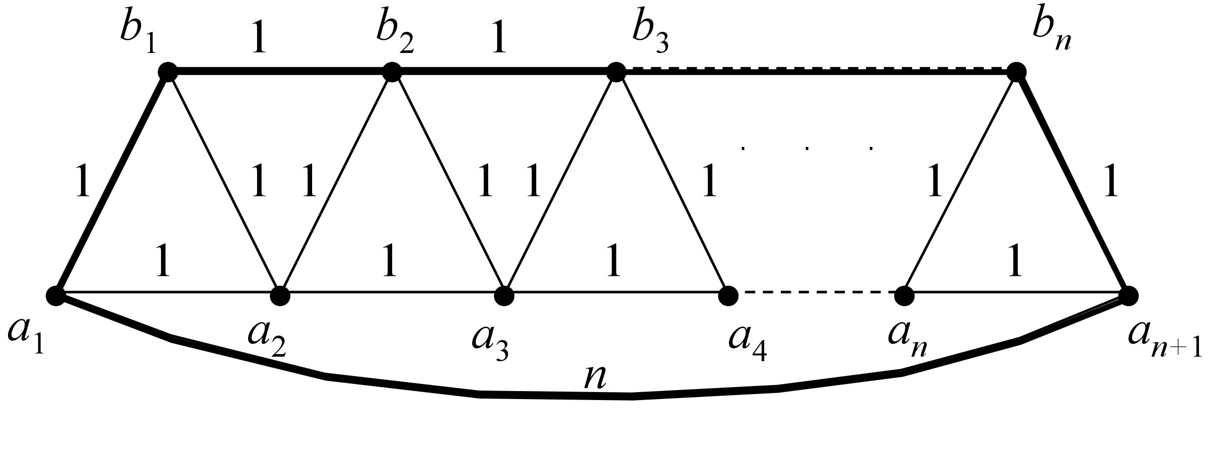
Algorytm Christofidesa

Uwaga Istnieją grafy, dla których współczynnik $W(C_{ch})/(3W(C_o))$ może przyjmować wartość dowolnie bliską 3/2.



1) minimalne drzewo spinające

2) $W(C_{ch}) = 3n$



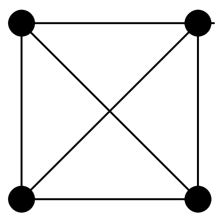
3) rozwiązanie optymalne $3W(C_o) = 2n + 1$

Planarność

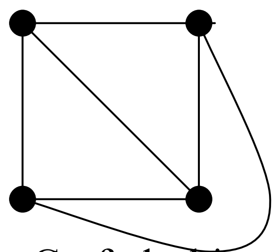
Def. *Graf płaski* – dowolny graf narysowany na płaszczyźnie w taki sposób, że jego krawędzie się nie przecinają.

Def. *Graf planarny* – graf izomorficzny z pewnym grafem płaskim.

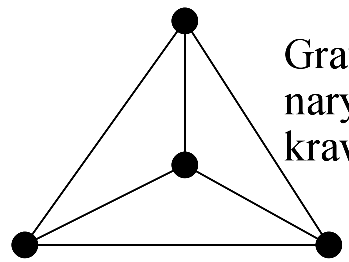
Przykład Graf K_4 jest planarny.



Graf K_4



Graf płaski izomorficzny z K_4

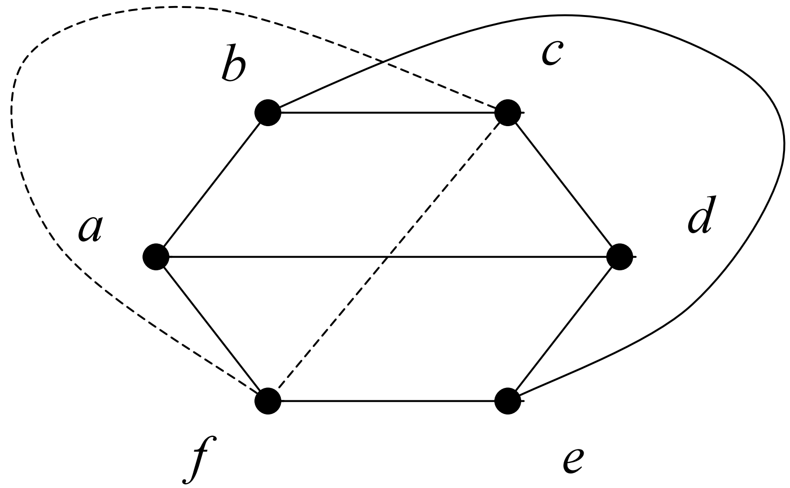


Graf płaski izomorficzny z K_4 narysowany tak, aby wszystkie krawędzie były odcinkami

Tw. *Każdy planarny graf prosty można narysować za pomocą odcinków*.

Planarność

Tw. *Graf $K_{3,3}$ jest nieplanarny.*



Dowód:

1) $K_{3,3}$ zawiera cykl C_6

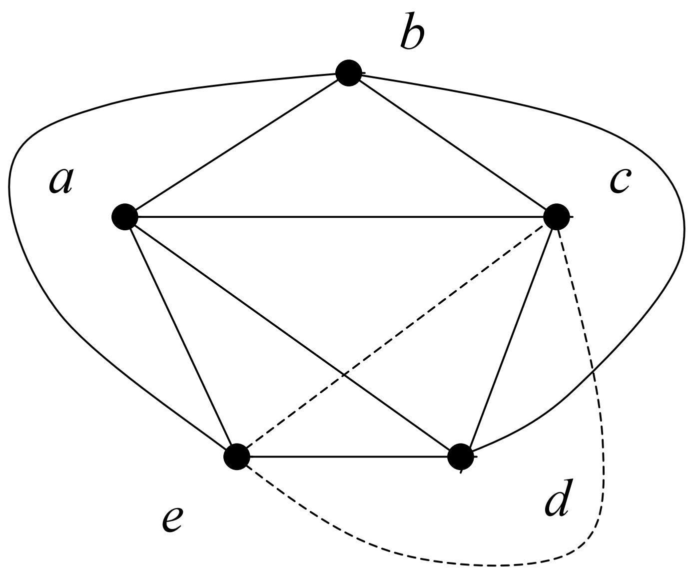
2) $\{a,d\} \in E(K_{3,3})$ (przypadek, gdy krawędź jest na zewnątrz cyklu jest analogiczny)

3) $\{b,e\} \in E(K_{3,3})$

4) $\{c,f\} \in E(K_{3,3})$ – każde narysowanie tej krawędzi prowadzi do przecięcia (z krawędzią $\{a,d\}$ lub $\{b,e\}$)

Planarność

Tw. *Graf K_5 jest nieplanarny*.



Dowód:

1) K_5 zawiera cykl C_5

2) $\{a,c\} \in E(K_5)$ (przypadek, gdy krawędź jest na zewnątrz cyklu jest analogiczny)

3) $\{b,e\}$, $\{b,d\}$ $\in E(K_5)$ i leżą w całości na zewnątrz cyklu C_5

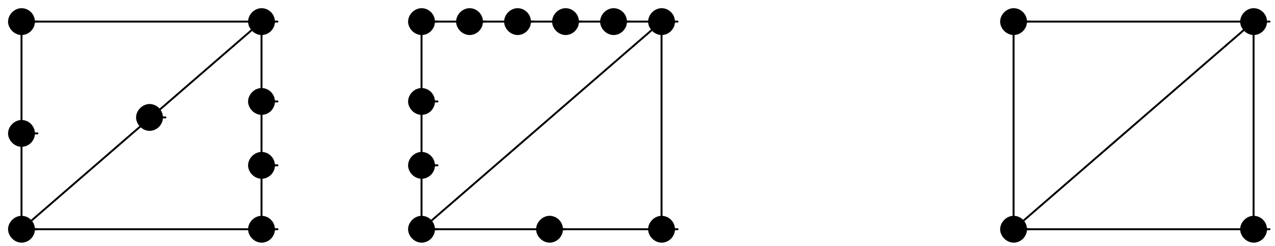
4) $\{a,d\} \in E(K_5)$ – krawędź znajduje się w całości wewnątrz cyklu

5) $\{c,e\} \in E(K_5)$ – każde narysowanie tej krawędzi prowadzi do przecięcia – z krawędzią $\{a,d\}$ lub $\{b,d\}$

Planarność

Def. Dwa grafy nazywamy *homeomorficznymi*, jeśli można je otrzymać z tego samego grafu poprzez zastępowanie jego krawędzi ścieżkami.

Przykład Podane dwa grafy są homeomorficzne (otrzymano je z grafu po prawej stronie):

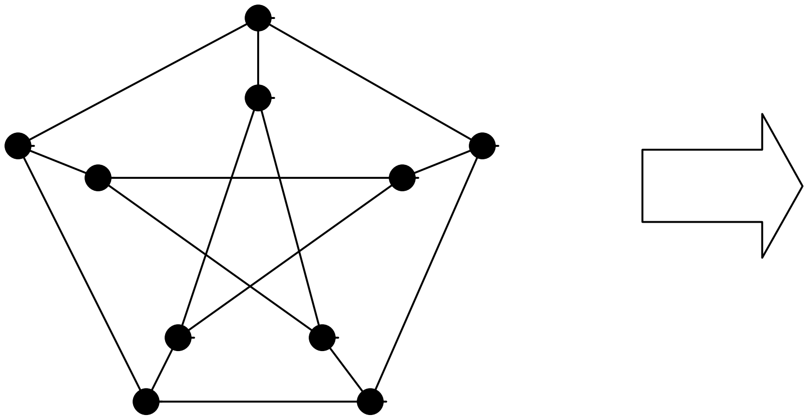


Tw. (Kuratowski, 1930) *Dany graf jest planarny wtedy i tylko wtedy, gdy nie zawiera podgrafu homeomorficznego z grafem K_5 lub z grafem $K_{3,3}$.*

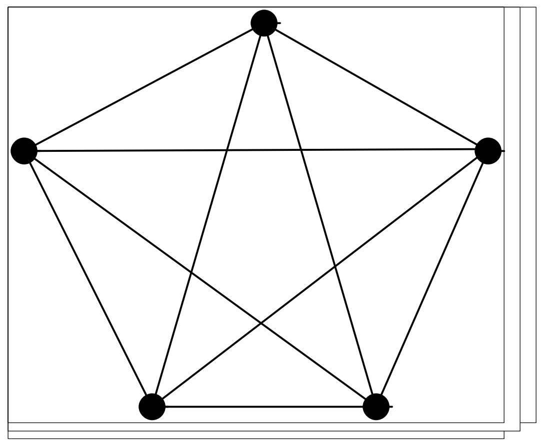
Planarność

Def. Graf G jest *ściągalny* do grafu H, gdy H można otrzymać ściągając kolejno krawędzie grafu G (ściąganie krawędzi $\{u,v\}$ polega na usunięciu jej z grafu i zastąpieniu wierzchołków u i v nowym wierzchołkiem o zbiorze sąsiadów $\{w \in V(G) : \{w,v\} \in E(G)$ lub $\{w,v\} \in E(G)\}$).

Przykład Graf Petersena jest ściągalny do K_5.



Graf Petersena



Graf K_5

Umieszczanie grafów

Def. *Umieszczanie grafu* to takie przedstawienie grafu w przestrzeni, że wierzchołkom odpowiadają punkty, a krawędziom linie, które się nie przecinają.

Tw. *Każdy graf może zostać umieszczony w* 3*-wymiarowej przestrzeni euklidesowej*.

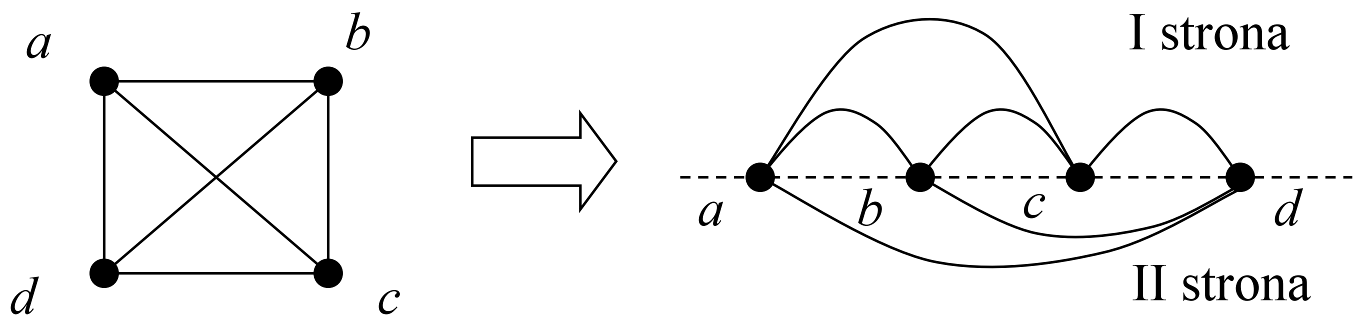
Tw. *Graf jest planarny wtedy i tylko wtedy, gdy może być umieszczony na powierzchni kuli*.

Umieszczanie grafów w książkach: Jest to funkcja odwzorowująca zbiór wierzchołków w zbiór punktów leżących na jednej prostej (grzbiecie) oraz zbiór krawędzi w zbiór płaszczyzn zawierających powyższą prostą (strony).

Def. *Grubość książkowa* $\beta(G)$ grafu G jest to najmniejsza liczbą stron, na której można umieścić graf.

Parametry planarności grafu

Przykład Umieszczenie grafu K_4 w książce o 2 stronach.



Tw. *Grubości książkowe cykli, hiperkostek i grafów pełnych dane są wzorami*:

$$\beta(C_n) = 1,$$

$$\beta(Q_n) = n - 1,$$

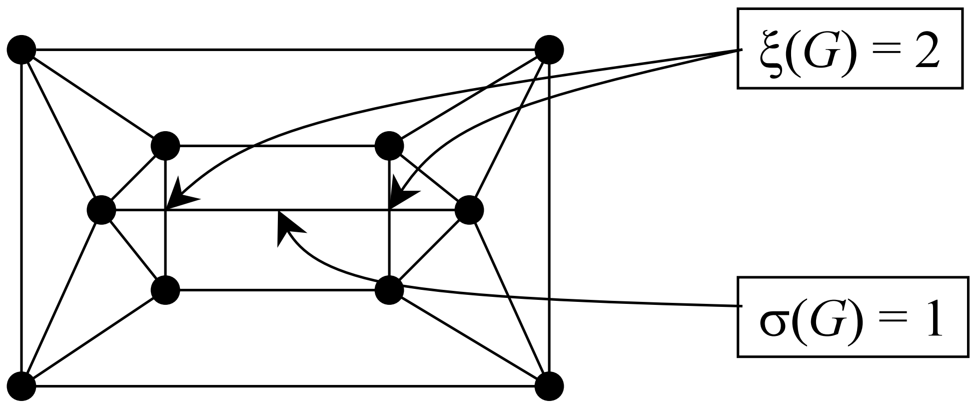
$$\beta(K_n) = \left\lfloor \frac{n}{2} \right\rfloor,$$

Parametry planarności grafu

Def. *Liczba przecięć* $\xi(G)$ grafu G jest najmniejszą liczbą przecięć, która musi wystąpić przy narysowaniu grafu na płaszczyźnie (zakładamy, że w jednym punkcie przecinają się co najwyżej dwie krawędzie).

Def. *Stopień nieplanarności* $\sigma(G)$ to minimalna liczba krawędzi, których usunięcie prowadzi do uzyskania grafu planarnego.

Przykład



Parametry planarności grafu

Tw. *Dla dowolnego grafu G* $\sigma(G) \le \xi(G)$.

Tw. *Oszacowania i wartości dokładne parametru* $\xi(G)$ *dla cykli, hiperkostek, grafów pełnych i pełnych dwudzielnych*:

$$\xi(C_n) = 0,$$

$$\xi(Q_n) = \frac{1}{6}2^{2n},$$

$$\frac{2}{7}\binom{n}{4} \le \xi(K_n) \le \frac{1}{4}\left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor,$$

$$\xi(K_{p,q}) \le \left\lfloor \frac{p}{2} \right\rfloor \left\lfloor \frac{p-1}{2} \right\rfloor \left\lfloor \frac{q}{2} \right\rfloor \left\lfloor \frac{q-1}{2} \right\rfloor.$$

Parametry planarności grafu

Tw. *Dla dowolnego grafu* G *i jego podgrafu* H *zachodzi*

$$\sigma(H) \le \sigma(G) \quad oraz \quad \xi(H) \le \xi(G).$$

Tw. *Wartości dokładne parametru* $\sigma(G)$ *dla cykli, hiperkostek, grafów pełnych i pełnych dwudzielnych*:

$$\sigma(C_n) = 0,$$

$$\sigma(Q_n) = 2^2(n-2) - n2^{n-1} + 4,$$

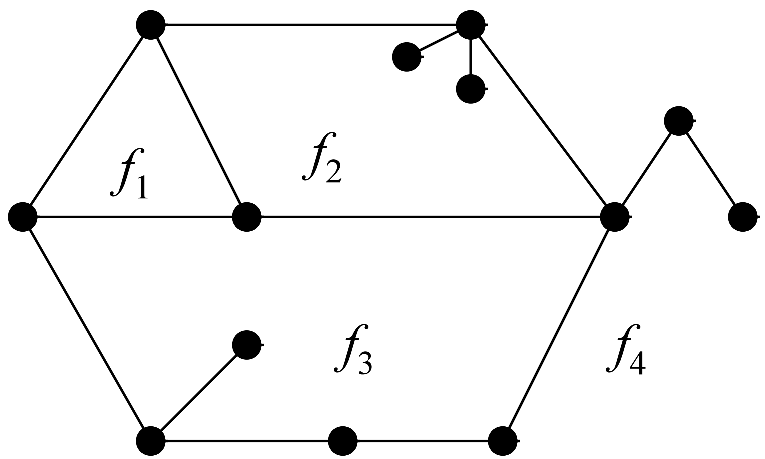
$$\sigma(K_n) = \frac{(n-3)(n-4)}{2},$$

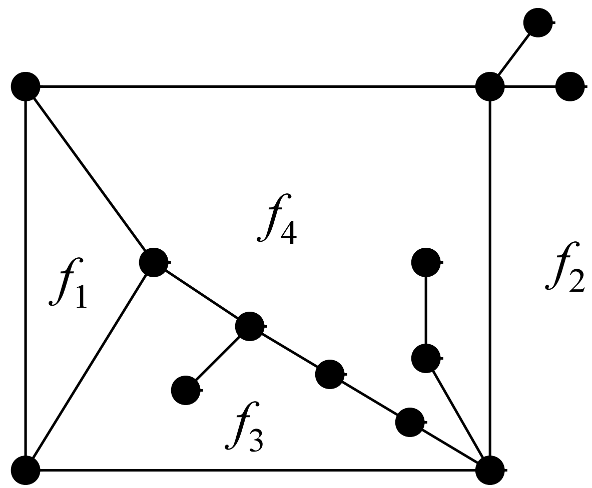
$$\sigma(K_{p,q}) = pq - 2(p+q) + 4.$$

Ściany w grafach

Def. *Ściana* w grafie płaskim zawierająca punkt x jest to zbiór wszystkich punktów y takich, że x i y można połączyć krzywą, która nie przecina żadnej krawędzi grafu. Ściana, która jest nieograniczona nazywana jest *ścianą nieskończoną*.

Przykład Poniższy graf zawiera cztery ściany – f_1, f_2, f_3 i f_4 (nieskończona). Narysujemy ten graf tak, aby f_2 była ścianą nieskończoną.





Ściany w grafach

Tw. (Euler, 1750) *Jeśli G jest spójnym grafem płaskim o n wierzchołkach, m krawędziach i f ścianach, to $n - m + f = 2$.*

Dowód: Przeprowadzimy indukcję względem m.

Jeśli $m = 0$, to $n = 1$ oraz $f = 1$, więc twierdzenie zachodzi.

Załóżmy, że twierdzenie jest prawdziwe dla wszystkich grafów o co najwyżej $m - 1$ krawędziach.

Rozważmy m-krawędziowy graf G.

Jeśli ten graf jest drzewem, to $m=n-1$ oraz $f = 1$, stąd tw. zachodzi.

Jeśli G zawiera cykl, to usuńmy pewną krawędź e, należącą do cyklu. Graf $G - e$ posiada $m - 1$ krawędzi i $f - 1$ ścian, więc korzystając z założenia indukcyjnego otrzymujemy:

$$n - (m - 1) + (f - 1) = 2,$$

co oznacza, że

$$n - m + f = 2.$$

Lemat o pocałunkach

Tw. *Dla każdego spójnego, prostego i płaskiego grafu G o $n \geq 3$ wierzchołkach zachodzi*: $2m \geq 3f$.

Dowód: Aby wykazać nierówność rozważmy dwa przypadki:

1) G jest drzewem. Prawdą jest, że $m \geq 2$, co oznacza, że:

$$2m \geq 4 > 3 = 3f.$$

2) G zawiera cykl. Usuwamy sukcesywnie wierzchołki stopnia 1, otrzymując *rdzeń* $R(G)$ grafu. Liczbę wierzchołków i krawędzi rdzenia oznaczmy odpowiednio przez n' i m'. Każda ściana rdzenia jest otoczona przez co najmniej 3 krawędzie (wynika to z założenia, że graf G jest prosty) oraz liczba ścian w G i w $R(G)$ są sobie równe. Zatem

$$2m' \geq 3f.$$

Z nierówności $m \geq m'$ otrzymujemy tezę.

Własności grafów planarnych

Tw. *Jeśli G jest prostym, planarnym grafem spójnym o $n \geq 3$ wierzchołkach, to $m \leq 3n - 6$.*

Dowód:

- Możemy założyć, że G jest grafem płaskim.
- Z lematu o pocałunkach mamy: $2m \geq 3f$.
- Ze wzoru Eulera: $n - m + f = 2$.
- Mnożymy ostatnią równość stronami przez 3 i przekształcamy:

$$3n - 3m + 3f = 6$$
$$3f = 6 - 3n + 3m$$
$$2m \geq 6 - 3n + 3m$$
$$m \leq 3n - 6.$$

Wniosek *Graf K_5 jest nieplanarny*.

Dowód: Wynika z powyższego twierdzenia, gdyż
gdyż $m(K_5) = 10$ oraz $3n(K_5) - 6 = 9$.

Własności grafów planarnych

Tw. *Jeśli G jest prostym, planarnym grafem spójnym o* $n \geq 3$ *wierzchołkach, nie zawierającym trójkątów, to* $m \leq 2n - 4$.

Dowód:

- Lemat o pocałunkach dla grafu bez trójkątów brzmi: $2m \geq 4f$, co zapisujemy jako $f \leq m/2$.
- Ze wzoru Eulera: $n - m + f = 2$.
- Łącząc dwa powyższe fakty otrzymujemy:

$$f = 2 - n + m$$

$$2 - n + m \leq m/2$$

$$m \leq 2n - 4.$$

Wniosek *Graf* $K_{3,3}$ *jest nieplanarny*.

Dowód: Wynika z powyższego twierdzenia, gdyż

$m(K_{3,3}) = 9$ oraz $2n(K_{3,3}) - 4 = 8$.

Własności grafów planarnych

Def. *Pąkiem* nazywamy wierzchołek grafu, którego stopień wynosi co najwyżej 5.

Tw. *Każdy planarny, prosty graf G zawiera co najmniej* 3 *pąki.*

Dowód:

- przypuśćmy, że G zawiera co najwyżej dwa pąki v_1 i v_2
- mamy: $2 + 6(n-2) \leq \deg(v_1) + \deg(v_2) + 6(n-2)$
- z założenia, że wszystkie wierzchołki oprócz v_1 i v_2 mają stopnie co najmniej 6 wynika: $\deg(v_1) + \deg(v_2) + 6(n-2) \leq 2m$
- udowodniliśmy wcześniej, że $m \leq 3n - 6$, co zapisujemy w postaci $2m \leq 6n - 12$.
- Łącząc powyższe nierówności:

$2 + 6n - 12 \leq 6n - 12$, co prowadzi do sprzeczności.

Grubość grafu

Def. *Grubość* $\tau(G)$ grafu to najmniejsza liczba grafów planarnych, które można złożyć, aby otrzymać graf G. Przez złożenie grafów G_1 i G_2, o tych samych zbiorach wierzchołków, rozumiemy taki graf G, że $V(G) = V(G_1)$ oraz $E(G) = \{e : e \in E(G_1) \text{ lub } e \in E(G_2)\}$.

Tw. *Niech będzie dany graf G o $n \geq 3$ wierzchołkach. Wtedy*

$$\tau(G) \geq \left\lceil \frac{m}{3n-6} \right\rceil, \tau(G) \geq \left\lfloor \frac{m+3n-7}{3n-6} \right\rfloor.$$

Dowód:

•Pierwsza nierówność wynika z udowodnionego wcześniej faktu dla grafów planarnych: $m \leq 3n - 6$. Parametr $\tau(G)$ jest liczbą całkowitą, więc bierzemy „sufit” z otrzymanego wyrażenia.

•Drugą równość otrzymujemy z pierwszej poprzez zastosowanie wzoru $\lfloor a/b \rfloor = \lceil (a+b-1)/b \rceil$.

Grubość grafu

Tw. *Dla grafów pełnych oraz hiperkostek zachodzą wzory*:

$$\tau(K_n) = \begin{cases} \left\lfloor \dfrac{n+7}{6} \right\rfloor & n \neq 9,10 \\ 3 & n = 9,10 \end{cases}, \quad \tau(Q_n) = \left\lceil \dfrac{n+1}{4} \right\rceil$$

Uwaga Graf G jest planarny wtedy i tylko wtedy, gdy $\tau(G) = 1$.
Oznacza to, że $\tau(K_5)$, $\tau(K_{3,3}) > 1$.

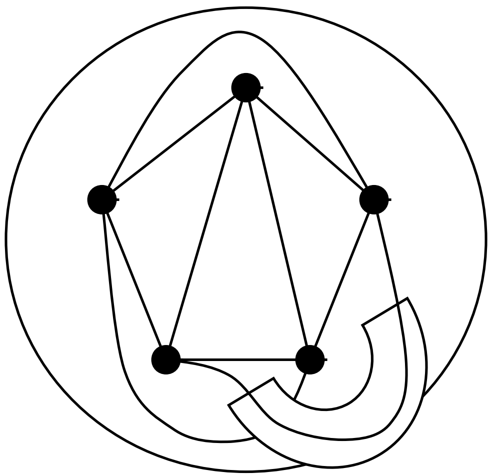
Tw. *Pomiędzy grubością grafu a stopniem nieplanarności i grubością książkową zachodzą nierówności*:

$$\tau(G) - 1 \leq \sigma(G),$$
$$\tau(G) \leq \lceil \beta(G)/2 \rceil.$$

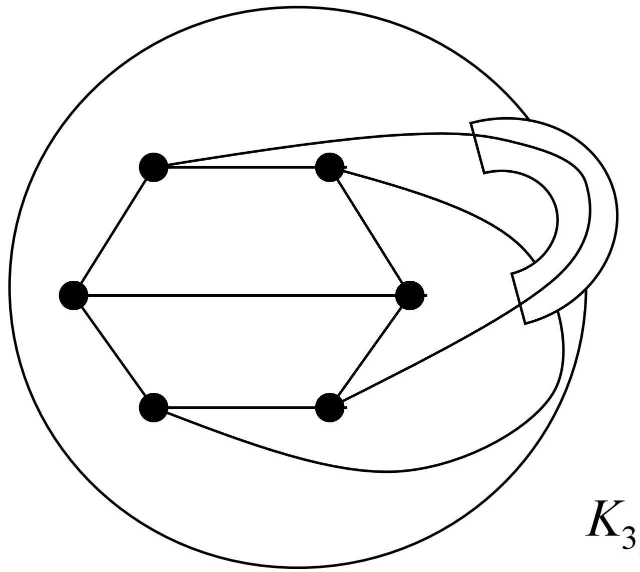
Genus grafu

Def. Mówimy, że powierzchnia jest *rodzaju* (*genusu*) g, jeśli jest ona „równoważna” kuli o g rączkach. Graf jest *rodzaju* (*genusu*) g, jeśli może być narysowany bez przecięć na powierzchni rodzaju g i jednocześnie nie można go narysować bez przecięć na powierzchni rodzaju $g - 1$.

Przykład Grafy K_5 i $K_{3,3}$ są grafami rodzaju 1.



K_5



$K_{3,3}$

Genus grafu

Tw. *Dla każdego grafu G zachodzi $g \leq \sigma(G)$, gdzie $\sigma(G)$ – stopień nieplanarności grafu.*

Dowód: Niech $S \subset E$ będzie zbiorem minimalnej mocy takim, że $G - S$ jest planarny. Oczywiście $|S| = \sigma(G)$. Jeśli graf $G - S$ narysujemy bez przecięć na sferze oraz dla każdej krawędzi z S wprowadzimy rączkę, umożliwiającą narysowanie tej krawędzi tak, aby nie przecinała innych, to otrzymamy rysunek grafu G bez przecięć na powierzchni rodzaju $|S|$, co kończy dowód.

Uogólnienie twierdzenia Eulera dla grafów genusu g ma postać:

Tw. *Jeśli genus grafu G o n wierzchołkach, m krawędziach i f ścianach jest równy g, to* $n - m + f = 2 - 2g$.

Genus grafu

Tw. *Jeśli G jest grafem genusu g o n≥4 wierzchołkach, m krawędziach, to*

$$g(G) \geq \left\lceil \frac{m-3n}{6} + 1 \right\rceil.$$

Dowód:

- Na mocy lematu o pocałunkach: $2m \geq 3f$ lub równoważnie $-2m \leq -3f$.
- Z uogólnionego wzoru Eulera wynika:

$$n - m + f = 2 - 2g$$

co zapisujemy w postaci

$$g = 1 + (n - m + f)/2.$$

- Stąd

$$g = 1 + (2m - 3n - 3f)/6 \geq 1 + (3m - 3n - 2m)/6 = 1 + (m - 3n)/6.$$

- Po uwzględnieniu, że g jest liczbą całkowitą, przyjmujemy jako oszacowanie dolne „sufit” z otrzymanego wyrażenia.

Genus grafu pełnego

Po zastosowaniu poprzedniego oszacowania dla grafu pełnego otrzymujemy:

$$g(K_n) \geq \left\lceil \frac{\frac{n(n-1)}{2} - 3n}{6} + 1 \right\rceil = \left\lceil \frac{(n-3)(n-4)}{12} \right\rceil.$$

Tw. (Ringel i Youngs, 1968)

$$g(K_n) = \left\lceil \frac{(n-3)(n-4)}{12} \right\rceil.$$

Wniosek *Dowolny grafy G o* $n \leq 7$ *wierzchołkach jest planarny lub toroidalny* (*genusu* 1).

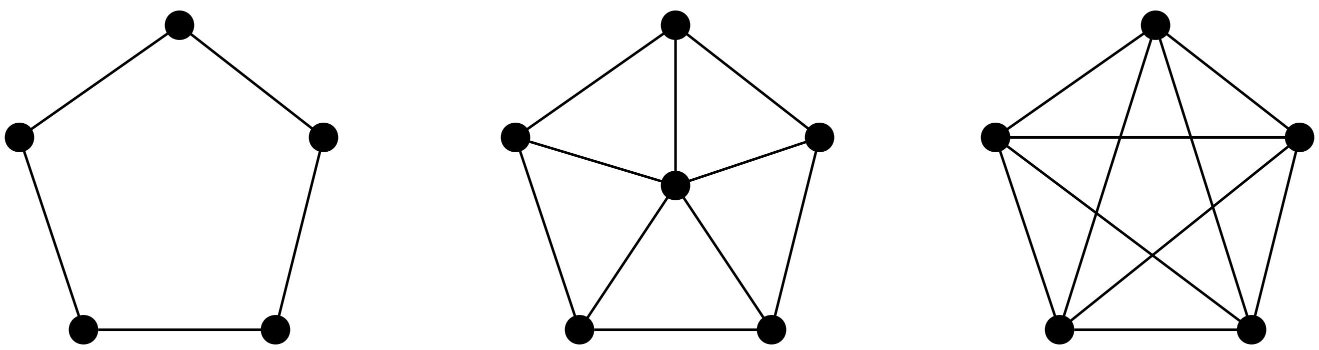
Dowód: $g(G) \leq g(K_7) = \lceil (7-3)(7-4)/12 \rceil = 1.$

Kolorowanie wierzchołków

Def. Niech G będzie grafem prostym. Przez *kolorowanie wierzchołków* rozumiemy takie etykietowanie elementów $V(G)$ liczbami naturalnymi, że sąsiednie wierzchołki otrzymują różne liczby (kolory, etykiety).

Def. *Liczba chromatyczna* grafu G jest to najmniejsza liczba k taka, że istnieje pokolorowanie G za pomocą k kolorów i jest oznaczana symbolem $\chi(G)$.

Przykład Optymalne (zużywające minimalną liczbę kolorów) pokolorowania grafów C_5, W_6, K_5.



Kolorowanie wierzchołków

Uwaga Problem wyznaczania liczby chromatycznej jest w ogólności NP-trudny. Zatem, w praktyce użyteczne są oszacowania.

Def. *Kliką* grafu G nazywamy jego podgraf pełny.

Lemat *Prawdziwe jest następujące oszacowanie dolne*:

$$\chi(G) \geq \omega,$$

gdzie ω jest rozmiarem maksymalnej kliki grafu G.

Uwaga Powyższe oszacowanie ma dwie wady:

•ω jest parametrem trudnym do wyliczenia. Ze związku $\frac{n^2}{n^2 - 2m} \leq \omega$ otrzymujemy oszacowanie mniej dokładne, lecz łatwe do obliczenia.

•różnica pomiędzy $\chi(G)$ a ω może być dowolnie duża, na co przykładem są grafy Mycielskiego.

Kolorowanie wierzchołków

Tw. *Dla dowolnego grafu o maksymalnym stopniu wierzchołka* Δ *zachodzi oszacowanie* $\chi(G) \leq \Delta + 1$.

Dowód: Przeprowadzimy indukcję względem n.

- Jeśli $n = 1$, to nierówność oczywiście zachodzi.
- Zakładamy, że twierdzenie jest prawdziwe dla pewnego $n > 0$.
- Dowodzimy przypadek, gdy graf G ma $n + 1$ wierzchołków.

Usuńmy z G dowolny wierzchołek v. Dla grafu $G - v$ z założenia indukcyjnego mamy: $\chi(G - v) \leq \Delta + 1$. Wierzchołek v ma w grafie G co najwyżej Δ sąsiadów, więc jeden spośród $\Delta+1$ kolorów jest dla v dostępny, co pozwala uzyskać pokolorowanie G za pomocą co najwyżej $\Delta+1$ barw.

Tw (Brooks, 1941) *Istnieją dwie klasy grafów, dla których* $\chi(G) = \Delta + 1$: *grafy pełne oraz cykle o nieparzystej liczbie wierzchołków*.

Wniosek *Jeśli* $G \neq K_n$ *oraz* $\Delta \geq 3$, *to* $\chi(G) \leq \Delta$.

Kolorowanie wierzchołków

Uwaga Oszacowanie $\chi(G) \leq \Delta$ może być bardzo niedokładne, zwłaszcza dla gwiazd, dla których $\chi(K_{1,s}) = 2$ oraz $\Delta(K_{1,s}) = s$.

Tw. *Dla grafu G o m krawędziach zachodzą oszacowania*:

$$\chi(G) \leq \sqrt{2m} + 1,$$

$$\chi(G) \leq \lambda + 1,$$

gdzie λ jest długością najdłuższej drogi w grafie G.

Uwaga Pierwsze z powyższych oszacowań może być niedokładne, gdyż dla grafów pełnych dwudzielnych $K_{k,k}$ różnica

$$\sqrt{2k^2} + 1 - \chi(G) = \sqrt{2}k - 1$$

może przyjmować dowolnie dużą wartość.

Uwaga Drugie z oszacowań jest niedokładne dla ścieżki P_n, dla której $\chi(P_n) = 2$ oraz $\lambda(P_n) = n - 1$.

Kolorowanie grafów planarnych

Tw. *Każdy graf planarny jest* 6-*barwny*.

Dowód: Zastosujemy indukcję względem liczby wierzchołków grafu.

•Jeśli $n = 1$, to twierdzenie jest oczywiście prawdziwe.

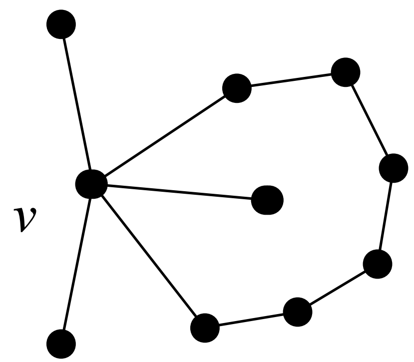
•Zakładamy, że własność zachodzi dla wszystkich (n–1)-wierzchołkowych grafów planarnych.

•Niech G będzie grafem planarnym o n wierzchołkach. Wiemy, że G posiada co najmniej jeden pąk v (wierzchołek o stopniu mniejszym lub równym 5). Po usunięciu v mamy $(n - 1)$-wierzchołkowy graf planarny $G - v$ do którego stosujemy założenie indukcyjne otrzymując jego 6-pokolorowanie. Wierzchołek v ma w G co najwyżej 5 sąsiadów, więc jeden z sześciu kolorów będzie dla v dostępny. Stąd G jest 6-barwny.

Kolorowanie grafów planarnych

Tw. (Heawood, 1890) *Każdy graf planarny jest* 5-*barwny*.

Dowód: Podobnie jak w poprzednim twierdzeniu stosujemy indukcję względem *n*. Jeśli wyznaczymy (z zał. ind.) 5-pokolorowanie *G* – *v* (gdzie *v* jest pąkiem) i wierzchołek *v* jest incydentny z co najwyżej 4 kolorami, to twierdzenie zachodzi. W przeciwnym wypadku rozważamy przypadki (kolory 1 – czerwony, 2 – zielony, 3 – niebieski, 4 – fioletowy):



Przypadek 2:

- Wierzchołki o kolorach 1,3 (sąsiedzi *v*) wraz z *v* tworzą cykl w *G*
- Wtedy w składowej spójności zawierającej sąsiada *v* o kolorze 2 możemy zamienić kolory 2 i 4
- *v* otrzymuje kolor 2

Tw. (Appel, Haken + komputer, 1976) *Każdy graf planarny jest* 4-*barwny*.

Algorytmy przybliżone

Przez $A(G)$ oznaczmy liczbę kolorów, którą algorytm A używa podczas kolorowania grafu G. Wyróżniamy następujące parametry, uwzględniane podczas opisu algorytmu przybliżonego A:

1) Złożoność obliczeniowa.

2) *Funkcja dobroci* zdefiniowana jako:

$$A(n) = \max\{\, A(G)/\chi(G) : G \text{ ma } n \text{ wierzchołków} \,\}.$$

Najgorszą możliwą funkcją dobroci jest $A(n)=n$, najlepszą zaś $A(n)=1$.

3) Najmniejszy *dość trudny* graf – najmniejszy graf G, dla którego algorytm <u>może</u> użyć więcej kolorów niż $\chi(G)$.

4) Najmniejszy *trudny* graf – najmniejszy graf G, dla którego algorytm <u>musi</u> użyć więcej kolorów niż $\chi(G)$.

Algorytm sekwencyjny

Algorytm sekwencyjny S można opisać następująco:

- Uporządkuj w dowolny sposób wierzchołki grafu G $v_1,\ldots,v_n$.
- Koloruj wierzchołki zachłannie zgodnie z przyjętą permutacją

Własności:

1) algorytm statyczny – kolejność wierzchołków ustalona na początku nie zmienia się podczas realizacji algorytmu

2) Ścieżka P_4 jest najmniejszym dość trudnym grafem

3) Graf trudny nie istnieje

4) Funkcja dobroci jest liniowa. Jej oczekiwana wartość wynika z oszacowania: $S(G) \leq (2 + \varepsilon)\ \chi(G)$

5) Złożoność $O(n + m)$

Algorytm *LF*

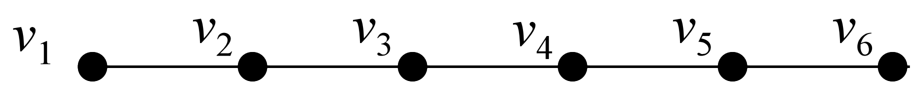
Algorytm *LF* (largest first) można opisać następująco:

- Uporządkuj wierzchołki grafu G malejąco według stopni $v_1,\ldots,v_n$.
- Koloruj wierzchołki zachłannie zgodnie z przyjętą permutacją

Własności:

1) algorytm statyczny – kolejność wierzchołków ustalona na początku nie zmienia się podczas realizacji algorytmu

2) Ścieżka P_6 jest najmniejszym dość trudnym grafem:

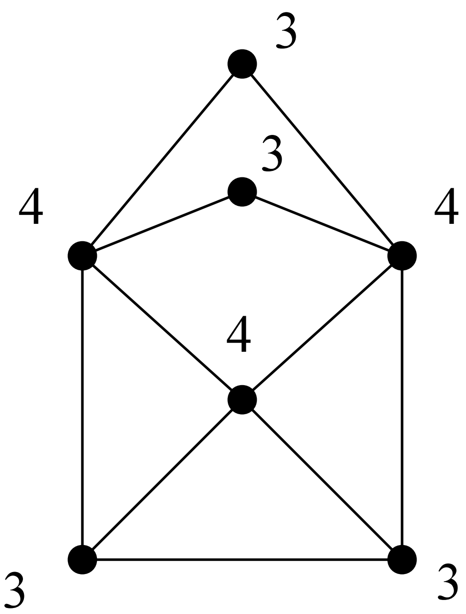


Kolejność wierzchołków: v_2, v_5, v_3, v_4, v_1, v_6.

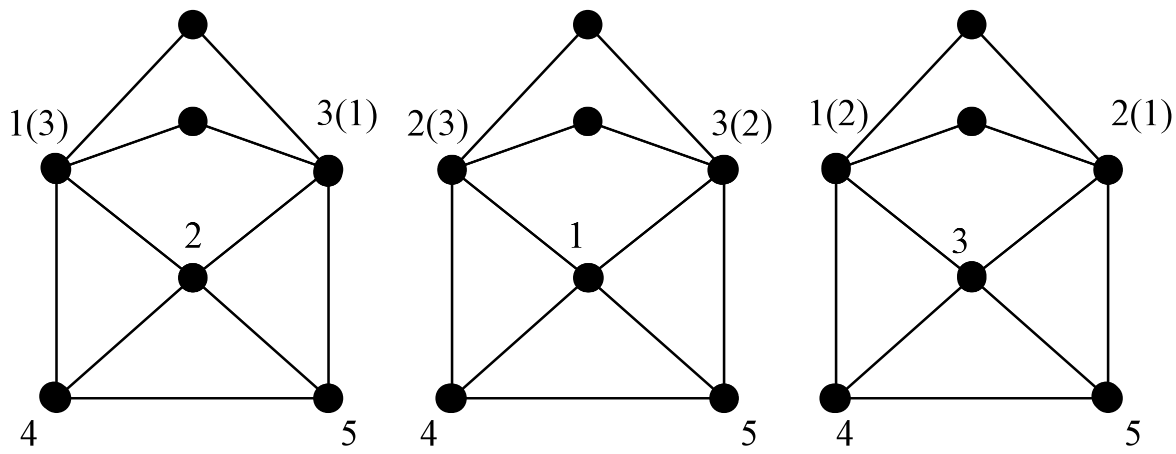
Algorytm *LF*

Własności algorytm *LF* (c.d.):

3) najmniejszym trudnym grafem do kolorowania jest „koperta”, która jest grafem 3-barwnym, natomiast *LF* zużywa czterech kolorów:



„Koperta” wraz z oznaczonymi stopniami wierzchołków



a) Szeregujemy wierzchołki stopnia 4

b) Kolorujemy te wierzchołki

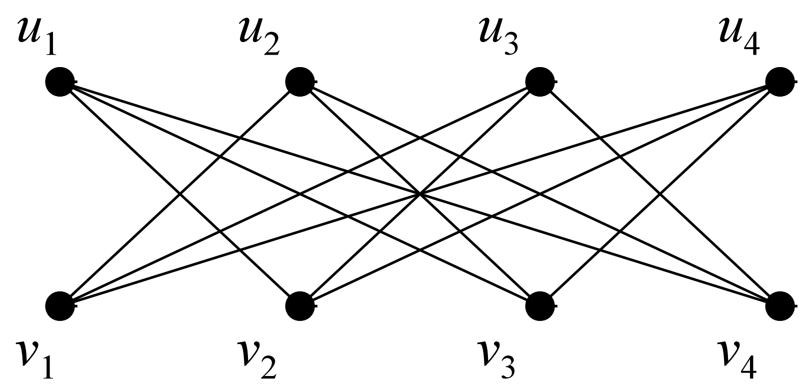
c) W każdym przypadku wybór wierzch. stopnia 3 jest symetryczny

d) We wszystkich przyp. wymagany kolor nr 5

Algorytm *LF*

Własności algorytm *LF* (cd.):

4) funkcja dobroci to $O(n)$. Zdefiniujmy k-ty graf Johnsona J_k jako $K_{k,k} - M$, gdzie $M = \{\{u_i, v_i\} : u_i \in V_1(K_{k,k}), v_i \in V_2(K_{k,k})\}$. Przykład grafu J_4 pokazuje rysunek (wraz z pokolorowaniem utworzonym przez algorytm *LF*).



$\chi(J_k) = 2$, gdyż grafy Johnsona są dwudzielne. Dla permutacji wierzchołków $u_1, v_1, u_2, v_2, \ldots, u_k, u_k$ algorytm *LF* używa k kolorów.

Stąd
$$\frac{LF(J_k)}{\chi(J_k)} = \frac{n/2}{2} = \frac{n}{4}$$

Algorytm *SL*

Algorytm *SL* (smallest last) składa się z dwóch etapów:

1) faza redukcji grafu: usuwamy wierzchołek o minimalnym stopniu i usuwamy go z grafu (powtarzamy dopóki graf nie jest pusty).
2) kolorujemy wierzchołki zachłannie w kolejności ustalonej w poprzednim kroku, zaczynając od wierzchołków usuwanych później.

Własności:

- Algorytm statyczny
- Złożoność algorytmu: $O(n+m)$
- Funkcja dobroci jest liniowa
- Przypadki pozytywne: drzewa, cykle, grafy jednocykliczne, kola, grafy Mycielskiego, grafy Johnsona, grafy planarne

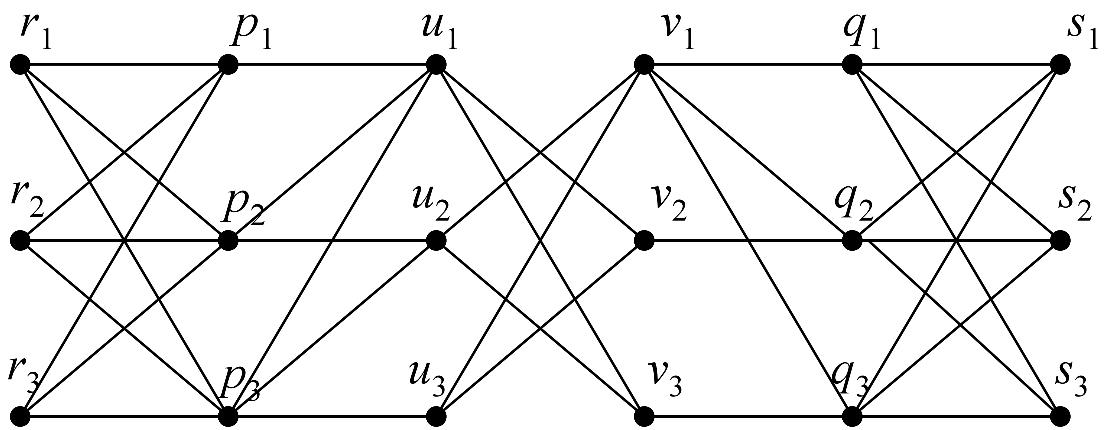
Algorytm *SL*

Własności (cd.) :

• Przypadki półpozytywne: grafy Mycielskiego, grafy Johnsona, grafy planarne (za pomocą sześciu kolorów w czasie $O(n)$)

• Przypadki negatywne: grafy dwudzielne, grafy Colemena-Moore'a

permutacja *SL* to: $q_1, s_1, \ldots, q_k, s_k, p_1, r_1, \ldots, p_k, r_k, u_1, v_1, \ldots, u_k, v_k$



$SL(CM_3) = 4$

Ogólnie:

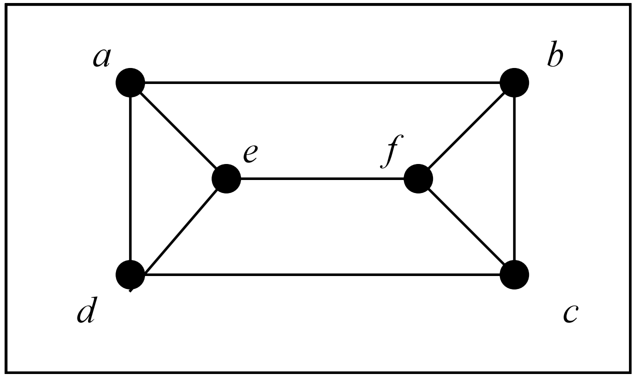
$SL(CM_k) = k$

$\chi(CM_k) = 2$

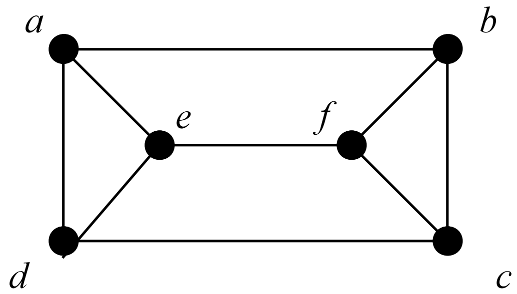
Algorytm *SL*

Własności algorytmu (cd.):

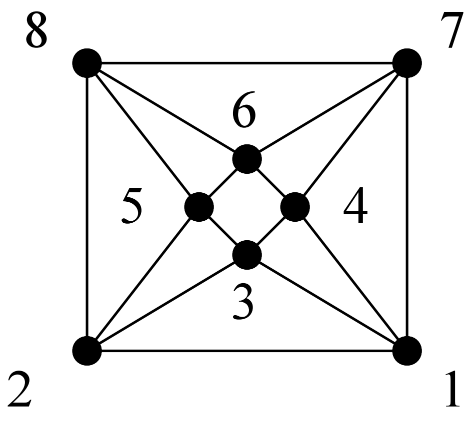
- Najmniejszym dość trudnym grafem jest „pryzma”



Permutacja: $a \quad b \quad f \quad e \quad c \quad d$



- Najmniejszym trudnym grafem jest „pryzmatoid”:



Algorytm *SLF*

Algorytm *SLF* (saturacyjny LF) można opisać następująco:

while istnieją nie pokolorowane wierzchołki **do begin**
  znajdź wierzchołek o maksymalnym stopniu spośród
   wierzchołków o maksymalnym stopniu nasycenia;
  pokoloruj znaleziony wierzchołek zachłannie;
end

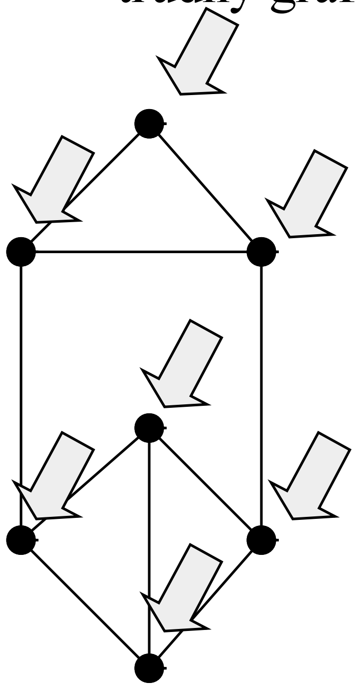
Uwaga *Stopień nasycenia* wierzchołka to ilość różnych kolorów incydentnych z tym wierzchołkiem.

- Przypadki pozytywne: grafy dwudzielne (w tym drzewa i grafy Johnsona), cykle, koła, kaktusy
- Przypadki negatywne: grafy trójdzielne

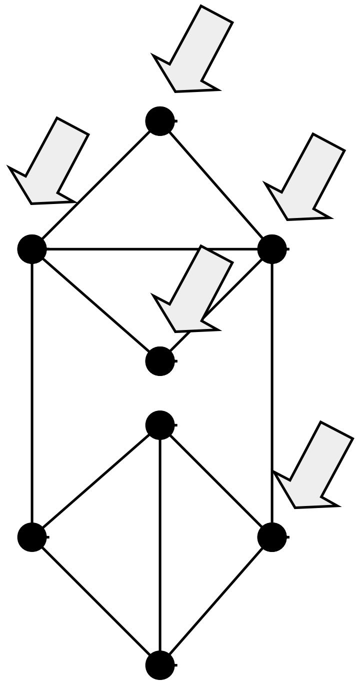
Algorytm *SLF*

Własności algorytmu:

- Złożoność: $O(m\log n)$
- Najmniejszy dość trudny graf:



- Najmniejszy trudny graf:



Pozostałe wierzchołki mogą być kolorowane w dowolnej kolejności, co zawsze prowadzi do użycia czwartego koloru.

Kolorowanie krawędzi

Def. Funkcja $c{:}E(G)\rightarrow\{1,...,k\}$ jest k-pokolorowaniem krawędziowym grafu G, o ile dla każdej pary sąsiednich krawędzi e i e' zachodzi $c(e)\neq c(e')$. Najmniejsze k, dla którego istnieje krawędziowe k-pokolorowanie nazywamy *indeksem chromatycznym* grafu G i oznaczamy symbolem $\chi'(G)$

Uwaga Pokolorowanie wierzchołków oznaczało rozbicie V na zbiory niezależne, natomiast pokolorowanie krawędzi k kolorami jest rozbiciem grafu na k skojarzeń.

Uwaga Problem kolorowania krawędzi jest równoważny kolorowaniu wierzchołków grafu krawędziowego.

Przykład $\chi'(G) = 2$ dla ścieżek i cykli parzystych

$\chi'(G) = 3$ dla drzew binarnych i cykli nieparzystych

Oszacowania dolne

Tw. *Zachodzi oszacowanie* $\Delta \leq \chi'(G)$

Dowód: Wynika stąd, iż wszystkie krawędzie incydentne z tym samym wierzchołkiem muszą otrzymać parami różne kolory.

Tw. *Zachodzi oszacowanie* $\lceil m/t \rceil \leq \chi'(G)$, *gdzie t jest rozmiarem maksymalnego skojarzenia*.

Dowód:

- Niech będzie dane pewne pokolorowanie grafu G.
- Zauważmy, że każdy z k kolorów jest przydzielony co najwyżej t krawędziom grafu G.
- Redukujemy graf, usuwając krawędzie o pewnym ustalonym kolorze.
- W każdym takim kroku usuniemy co najwyżej t krawędzi, co oznacza, że musimy wykonać co najmniej $\lceil m/t \rceil$ powyższych kroków, aby zredukować graf do grafu pustego.
- Ilość kroków jest równa ilości kolorów, co kończy dowód.

Oszacowania górne

Tw. $\chi'(G) \le \max\{\Delta, \lfloor \deg(u) + \deg(v) + \deg(w) \rfloor/2\}$, *gdzie maksimum jest obliczane względem wszystkich dróg elementarnych długości* 2.

Tw (Shannon, 1949). $\chi'(G) \le 3\Delta/2$

Tw (Vizing, 1964). $\chi'(G) \le \Delta + \mu$, *gdzie* μ *jest maksymalnym zwielokrotnieniem krawędzi w grafie, tzn.* μ *jest największą liczbą k taką, że występuje para wierzchołków połączonych k krawędziami*.

Uwaga

- Dla dużych wartości parametru μ i specyficznych grafów (np. dwuwierzchołkowych), oszacowanie Vizing'a jest słabsze od oszacowania Ore'go.
- Dla $\mu = 1$ oszacowanie Vizing'a jest bardzo dokładne jako, że $\chi'(G) \ge \Delta$.

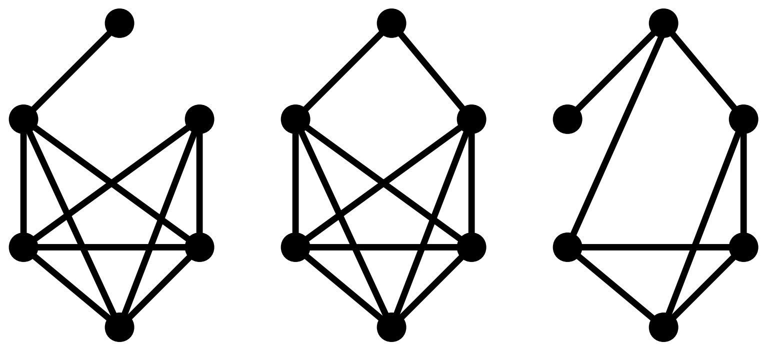
Tw. Vizinga

Wniosek *Dla grafów prostych G zachodzi* $\Delta \leq \chi'(G) \leq \Delta + 1$.

Uwaga

- Grafy, dla których $\chi'(G) = \Delta$ nazywamy grafami *klasy* 1. Przykłady to grafy dwudzielne, pełne o parzystej liczbie wierzchołków, planarne o $\Delta \geq 8$, nieparzystego rzędu z gwiazdą spinającą.
- Grafy *klasy* 2, to takie, dla których $\chi'(G) = \Delta + 1$. Przykładami są nieparzyste cykle, pełne nieparzystego rzędu, regularne nieparzystego rzędu.

Uwaga Grafów klasy 1 jest znacznie więcej. Np. spośród 112 grafów rzędu 6, tylko 3 są klasy 2.



Algorytm *NC*

Algorytm w każdym kroku wybiera dowolną krawędź i przydziela jej najniższy kolor, spośród kolorów, które nie zostały użyte do pokolorowania krawędzi sąsiednich.

Własności:

- Złożoność algorytmu to $O(m\Delta)$
- Najmniejszym dość trudnym grafem jest ścieżka P_4
- Najmniejszy trudny graf nie istnieje
- Algorytm jest 2-przybliżony, tzn. $NC(G) < 2\chi'(G)$.

  Uzasadnienie: Jeśli algorytm *NC* koloruje pewną krawędź $e = \{u,v\}$, to w najgorszym przypadku są $\deg(u) + \deg(v) - 2$ zabronione kolory dla e. Oznacza to, że kolor przydzielony e jest nie większy niż $\deg(u) + \deg(v) - 1$.

  Stąd:
  $$NC(G) \le \max\{ \deg(u) + \deg(u) - 1 : \{u,v\} \in E(G) \}$$
  $$\le 2\,\Delta - 1 < 2\,\Delta \le 2\chi'(G).$$

Algorytm *NTL*

Nazwa metody pochodzi od pierwszych liter nazwisk jej twórców (Nishizeki, Terada, Leven)

```
Procedure AlgorytmNTL( G )
begin
   if Δ(G) ≤ 2 then koloruj optymalnie trawersując ścieżki i cykle;
   else begin
      q := Δ(G) + 1; G' := (V(G), ∅);
      for  każda e ∈ E(G) do begin
          G' := G' + e;
          if  e = {u,v} nie może otrzymać wspólnego koloru brakującego
              w u i v  then
              Recolor( u, v );
          koloruj e;
      end
   end
end
```