

Stefan Sokołowski

MATEMATYKA DYSKRETNA

wykład 9

**Inst. Informatyki Stosowanej, ANS
Elbląg, 2024/2025**

Grafy

Grafy

Definicja:

Graf (nieskierowany)

Grafy

Definicja:

Graf (nieskierowany) — para $G = \langle V, E \rangle$

- V — dowolny zbiór **wierzchołków** (*vertices*)

Grafy

Definicja:

Graf (nieskierowany) — para $G = \langle V, E \rangle$

- V — dowolny zbiór **wierzchołków** (*vertices*);
- E — zbiór **krawędzi** (*edges*)

Grafy

Definicja:

Graf (nieskierowany) — para $G = \langle V, E \rangle$

- V — dowolny zbiór **wierzchołków** (*vertices*);
- E — zbiór **krawędzi** (*edges*); pojedyncza krawędź to dwuelementowy zbiór wierzchołków.

Grafy

Definicja:

Graf (nieskierowany) — para $G = \langle V, E \rangle$

- V — dowolny zbiór **wierzchołków** (*vertices*);
- E — zbiór **krawędzi** (*edges*); pojedyncza krawędź to dwuelementowy zbiór wierzchołków.

Przykład:

$$G = \left\langle \overbrace{\{a, b, c, d\}}^V, \overbrace{\{\}}^E \right\rangle$$

Grafy

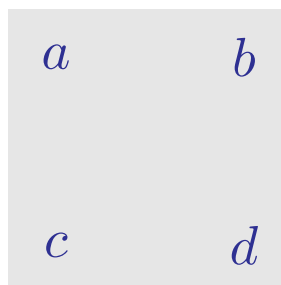
Definicja:

Graf (nieskierowany) — para $G = \langle V, E \rangle$

- V — dowolny zbiór **wierzchołków** (*vertices*);
- E — zbiór **krawędzi** (*edges*); pojedyncza krawędź to dwuelementowy zbiór wierzchołków.

Przykład:

$$G = \left\langle \overbrace{\{a, b, c, d\}}^V, \overbrace{\{\}}^E \right\rangle$$



Grafy

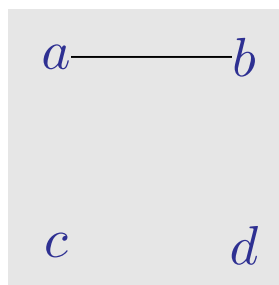
Definicja:

Graf (nieskierowany) — para $G = \langle V, E \rangle$

- V — dowolny zbiór **wierzchołków** (*vertices*);
- E — zbiór **krawędzi** (*edges*); pojedyncza krawędź to dwuelementowy zbiór wierzchołków.

Przykład:

$$G = \left\langle \overbrace{\{a, b, c, d\}}^V, \overbrace{\{\{a, b\}\}}^E \right\rangle$$



Grafy

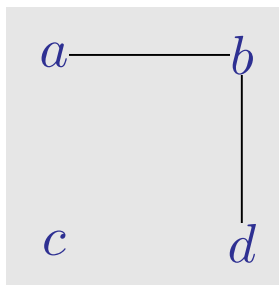
Definicja:

Graf (nieskierowany) — para $G = \langle V, E \rangle$

- V — dowolny zbiór **wierzchołków** (*vertices*);
- E — zbiór **krawędzi** (*edges*); pojedyncza krawędź to dwuelementowy zbiór wierzchołków.

Przykład:

$$G = \left\langle \overbrace{\{a, b, c, d\}}^V, \overbrace{\{\{a, b\}, \{b, d\}\}}^E \right\rangle$$



Grafy

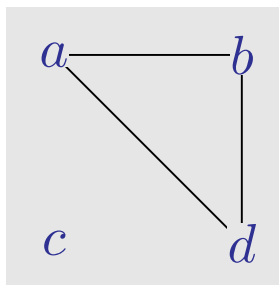
Definicja:

Graf (nieskierowany) — para $G = \langle V, E \rangle$

- V — dowolny zbiór **wierzchołków** (*vertices*);
- E — zbiór **krawędzi** (*edges*); pojedyncza krawędź to dwuelementowy zbiór wierzchołków.

Przykład:

$$G = \left\langle \overbrace{\{a, b, c, d\}}^V, \overbrace{\{\{a, b\}, \{b, d\}, \{a, d\}\}}^E \right\rangle$$



Grafy

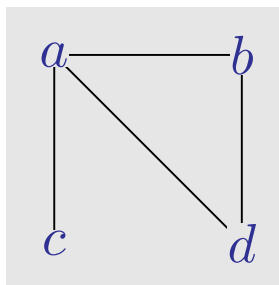
Definicja:

Graf (nieskierowany) — para $G = \langle V, E \rangle$

- V — dowolny zbiór **wierzchołków** (*vertices*);
- E — zbiór **krawędzi** (*edges*); pojedyncza krawędź to dwuelementowy zbiór wierzchołków.

Przykład:

$$G = \left\langle \overbrace{\{a, b, c, d\}}^V, \overbrace{\{\{a, b\}, \{b, d\}, \{a, d\}, \{a, c\}\}}^E \right\rangle$$



Grafy

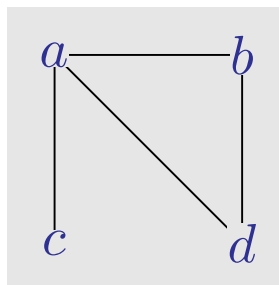
Definicja:

Graf (nieskierowany) — para $G = \langle V, E \rangle$

- V — dowolny zbiór **wierzchołków** (*vertices*);
- E — zbiór **krawędzi** (*edges*); pojedyncza krawędź to dwuelementowy zbiór wierzchołków.

Przykład:

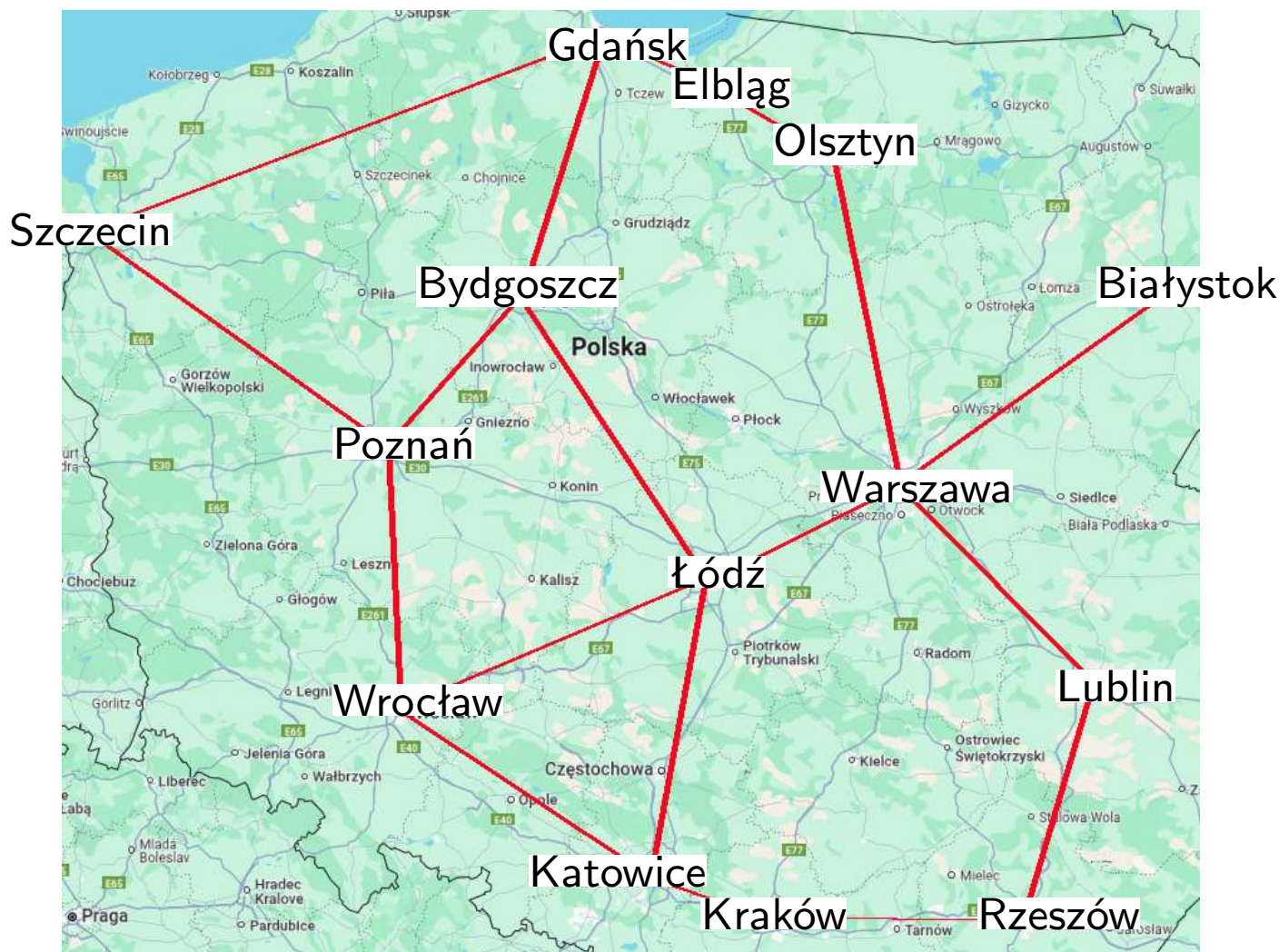
$$G = \left\langle \overbrace{\{a, b, c, d\}}^V, \overbrace{\{\{a, b\}, \{b, d\}, \{a, d\}, \{a, c\}\}}^E \right\rangle$$



Krawędzie muszą być zbiorami **naprawdę** 2-elementowymi; $\{v, v\}$ nie jest krawędzią.



Grafy



$$G = \left\langle \left\{ \text{Białystok, Bydgoszcz, Elbląg, Gdańsk, Katowice, Kraków, Lublin, Łódź, Olsztyn, Poznań, Rzeszów, Szczecin, Warszawa, Wrocław} \right\}, \left\{ \begin{array}{l} \{ \text{Elbląg, Gdańsk} \}, \{ \text{Elbląg, Olsztyn} \}, \{ \text{Olsztyn, Warszawa} \}, \\ \{ \text{Gdańsk, Szczecin} \}, \{ \text{Gdańsk, Bydgoszcz} \}, \{ \text{Bydgoszcz, Poznań} \}, \dots \end{array} \right\} \right\rangle$$

Grafy — podstawy

Rysunek jest sprawą drugorzędną...

Grafy — podstawy

Rysunek jest sprawą drugorzędną...

Nie każdy graf daje się narysować na płaszczyźnie bez przecinających się krawędzi...

Grafy — podstawy

Rysunek jest sprawą drugorzędną...

Nie każdy graf daje się narysować na płaszczyźnie bez przecinających się krawędzi...

Przykład:

$$G = \left\langle \begin{array}{l} \{a, b, c, d, e\} , \\ \{\{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}, \{c, d\}, \{c, e\}, \{d, e\}\} \end{array} \right\rangle$$

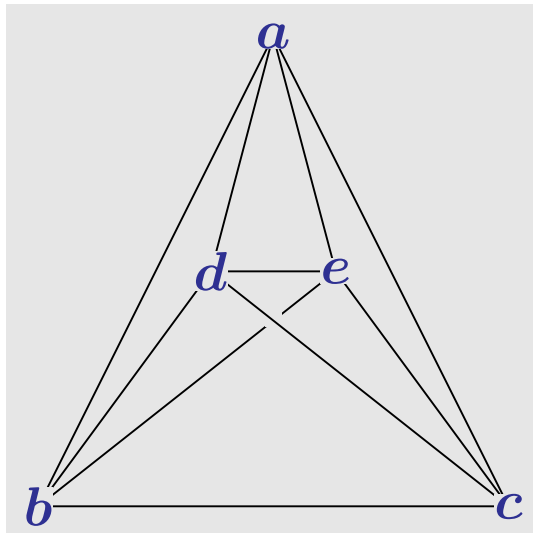
Grafy — podstawy

Rysunek jest sprawą drugorzędną...

Nie każdy graf daje się narysować na płaszczyźnie bez przecinających się krawędzi...

Przykład:

$$G = \left\langle \begin{array}{l} \{a, b, c, d, e\} , \\ \{\{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}, \{c, d\}, \{c, e\}, \{d, e\}\} \end{array} \right\rangle$$



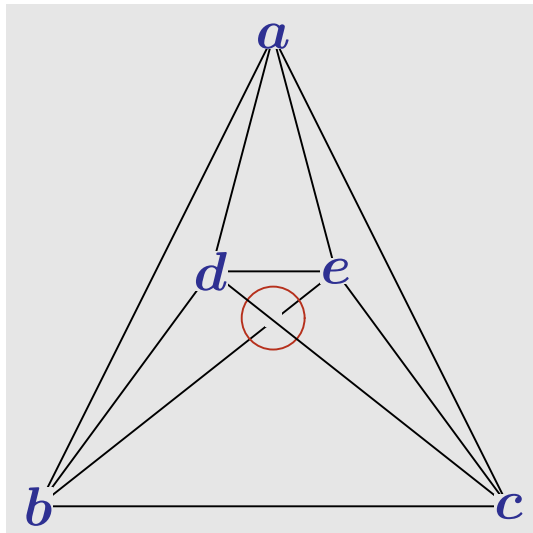
Grafy — podstawy

Rysunek jest sprawą drugorzędną...

Nie każdy graf daje się narysować na płaszczyźnie bez przecinających się krawędzi...

Przykład:

$$G = \left\langle \begin{array}{l} \{a, b, c, d, e\} , \\ \{\{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}, \{c, d\}, \{c, e\}, \{d, e\}\} \end{array} \right\rangle$$



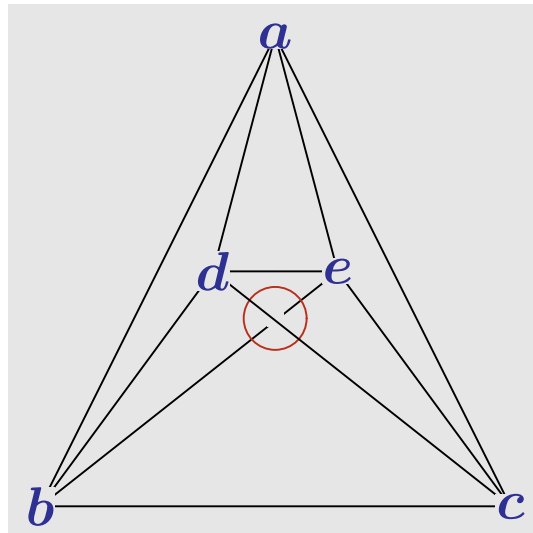
Grafy — podstawy

Rysunek jest sprawą drugorzędną...

Nie każdy graf daje się narysować na płaszczyźnie bez przecinających się krawędzi...

Przykład:

$$G = \left\langle \begin{array}{l} \{a, b, c, d, e\} , \\ \{\{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}, \{c, d\}, \{c, e\}, \{d, e\}\} \end{array} \right\rangle$$



Takie grafy są pełnoprawne; rysunek jest tylko nieistotną poglądową ilustracją.

Grafy — podstawy

Definicja:

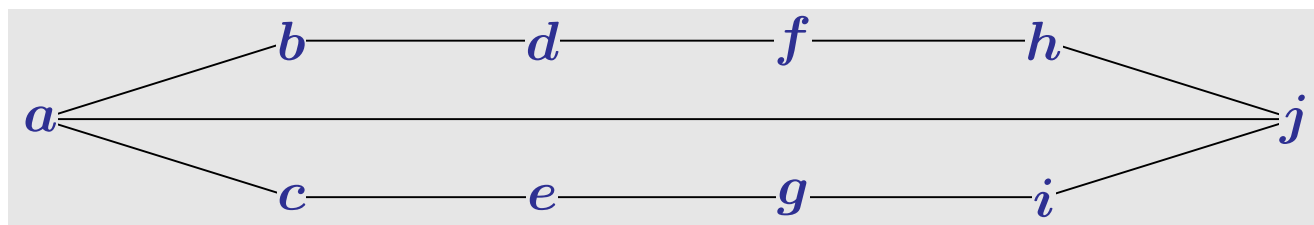
Dwa wierzchołki w grafie są **sąsiednie**, jeśli łączy je krawędź.

Grafy — podstawy

Definicja:

Dwa wierzchołki w grafie są **sąsiednie**, jeśli łączy je krawędź.

Przykład:

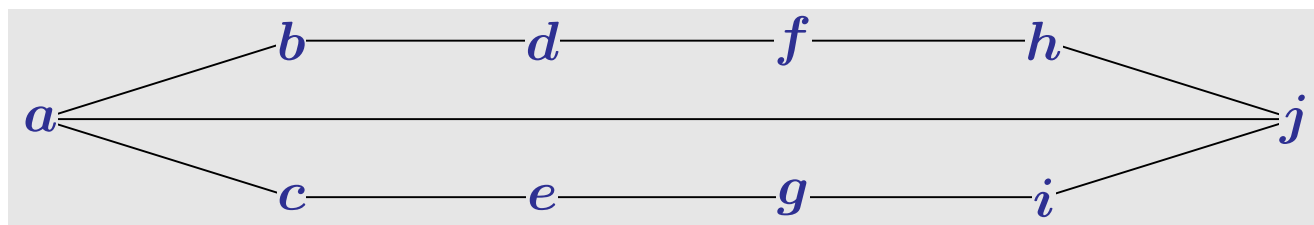


Grafy — podstawy

Definicja:

Dwa wierzchołki w grafie są **sąsiednie**, jeśli łączy je krawędź.

Przykład:



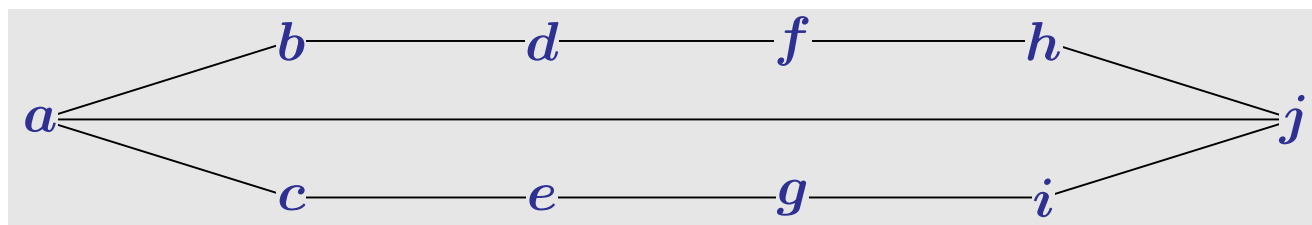
a i j — sąsiednie

Grafy — podstawy

Definicja:

Dwa wierzchołki w grafie są **sąsiednie**, jeśli łączy je krawędź.

Przykład:



a i j — sąsiednie

Definicja:

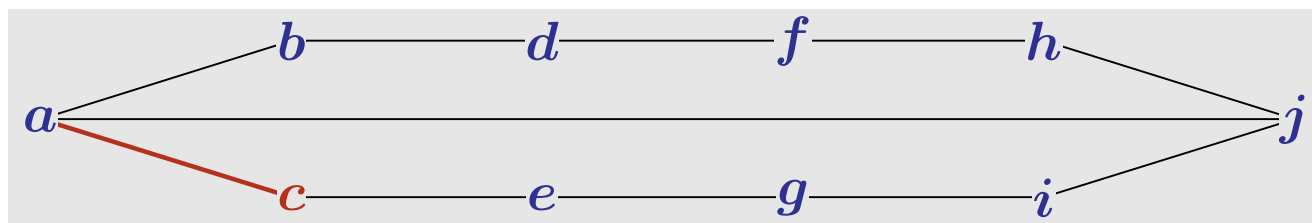
Wierzchołek $v \in V$ jest **incydentny** z krawędzią $e \in E$, jeśli $v \in e$.

Grafy — podstawy

Definicja:

Dwa wierzchołki w grafie są **sąsiednie**, jeśli łączy je krawędź.

Przykład:



a i j — sąsiednie

Definicja:

Wierzchołek $v \in V$ jest **incydentny** z krawędzią $e \in E$, jeśli $v \in e$.

Przykład:

W powyższym grafie

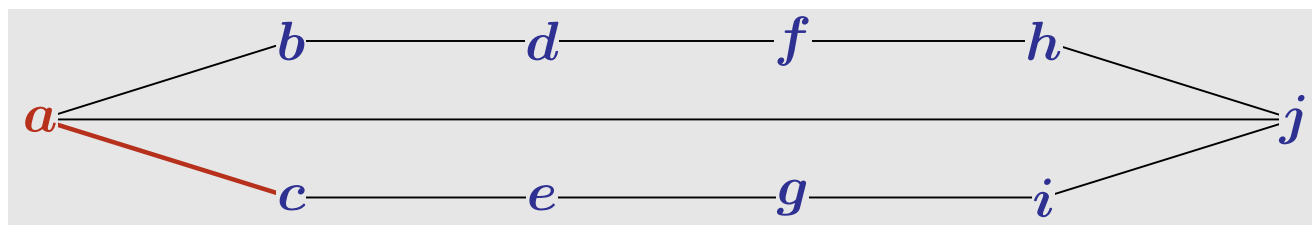
c jest incydentny z $\{a, c\}$

Grafy — podstawy

Definicja:

Dwa wierzchołki w grafie są **sąsiednie**, jeśli łączy je krawędź.

Przykład:



a i j — sąsiednie

Definicja:

Wierzchołek $v \in V$ jest **incydentny** z krawędzią $e \in E$, jeśli $v \in e$.

Przykład:

W powyższym grafie

c jest incydentny z $\{a, c\}$

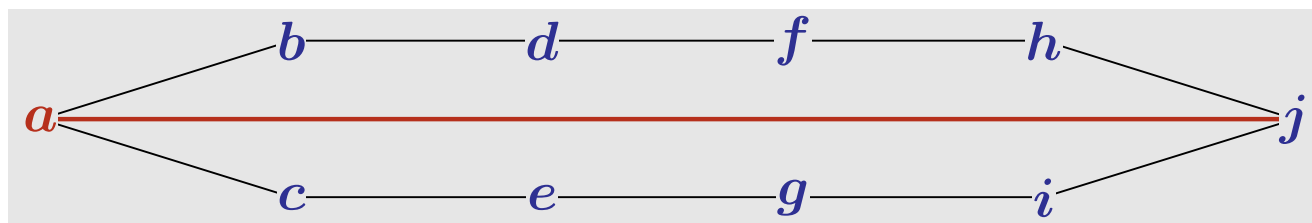
a jest incydentny z $\{a, c\}$

Grafy — podstawy

Definicja:

Dwa wierzchołki w grafie są **sąsiednie**, jeśli łączy je krawędź.

Przykład:



a i j — sąsiednie

Definicja:

Wierzchołek $v \in V$ jest **incydentny** z krawędzią $e \in E$, jeśli $v \in e$.

Przykład:

W powyższym grafie

c jest incydentny z $\{a, c\}$

a jest incydentny z $\{a, c\}$

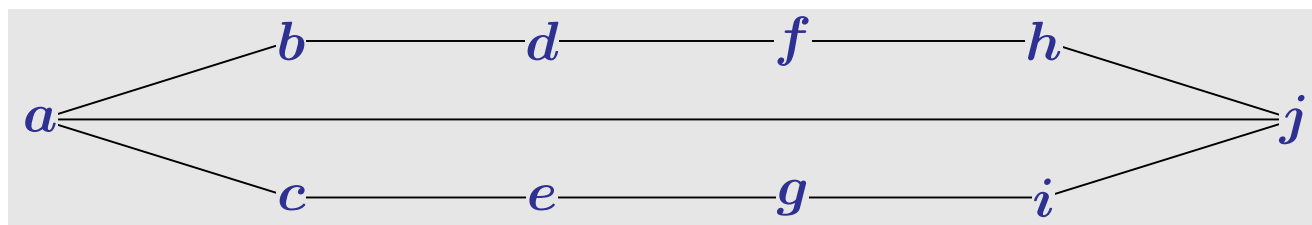
a jest incydentny z $\{j, a\}$

Grafy — podstawy

Definicja:

Dwa wierzchołki w grafie są **sąsiednie**, jeśli łączy je krawędź.

Przykład:



a i j — sąsiednie

Definicja:

Wierzchołek $v \in V$ jest **incydentny** z krawędzią $e \in E$, jeśli $v \in e$.

Przykład:

W powyższym grafie

c jest incydentny z $\{a, c\}$

a jest incydentny z $\{a, c\}$

a jest incydentny z $\{j, a\}$

.....

Grafy — podstawy

Definicja:

Stopień wierzchołka $v \in V$ w grafie $G = \langle V, E \rangle$ to liczba jego sąsiadów

Grafy — podstawy

Definicja:

Stopień wierzchołka $v \in V$ w grafie $G = \langle V, E \rangle$ to liczba jego sąsiadów:

$$d_G(v) \stackrel{\text{def}}{=} |N_G(v)| \quad \text{gdzie}$$
$$N_G(v) \stackrel{\text{def}}{=} \left\{ u \in V \mid \{v, u\} \in E \right\}$$

Grafy — podstawy

Definicja:

Stopień wierzchołka $v \in V$ w grafie $G = \langle V, E \rangle$ to liczba jego sąsiadów:

$$d_G(v) \stackrel{\text{def}}{=} |N_G(v)| \quad \text{gdzie} \\ N_G(v) \stackrel{\text{def}}{=} \left\{ u \in V \mid \{v, u\} \in E \right\}$$

Definicja:

Wierzchołek stopnia 0 (nie mający sąsiadów) nazywamy **izolowanym**.

Grafy — podstawy

Definicja:

Stopień wierzchołka $v \in V$ w grafie $G = \langle V, E \rangle$ to liczba jego sąsiadów:

$$d_G(v) \stackrel{\text{def}}{=} |N_G(v)| \quad \text{gdzie}$$
$$N_G(v) \stackrel{\text{def}}{=} \left\{ u \in V \mid \{v, u\} \in E \right\}$$

Definicja:

Wierzchołek stopnia 0 (nie mający sąsiadów) nazywamy **izolowanym**.

Twierdzenie: (lemat o uściskach dłoni)

$$\sum_{v \in V} d_G(v) = 2 \cdot |E|$$

Grafy — podstawy

Definicja:

Stopień wierzchołka $v \in V$ w grafie $G = \langle V, E \rangle$ to liczba jego sąsiadów:

$$d_G(v) \stackrel{\text{def}}{=} |N_G(v)| \quad \text{gdzie}$$
$$N_G(v) \stackrel{\text{def}}{=} \{u \in V \mid \{v, u\} \in E\}$$

Definicja:

Wierzchołek stopnia 0 (nie mający sąsiadów) nazywamy **izolowanym**.

Twierdzenie: (lemat o uściskach dłoni)

$$\sum_{v \in V} d_G(v) = 2 \cdot |E|$$

— czyli suma stopni wszystkich wierzchołków to 2-krotność liczby krawędzi.

Grafy — podstawy

Wniosek:

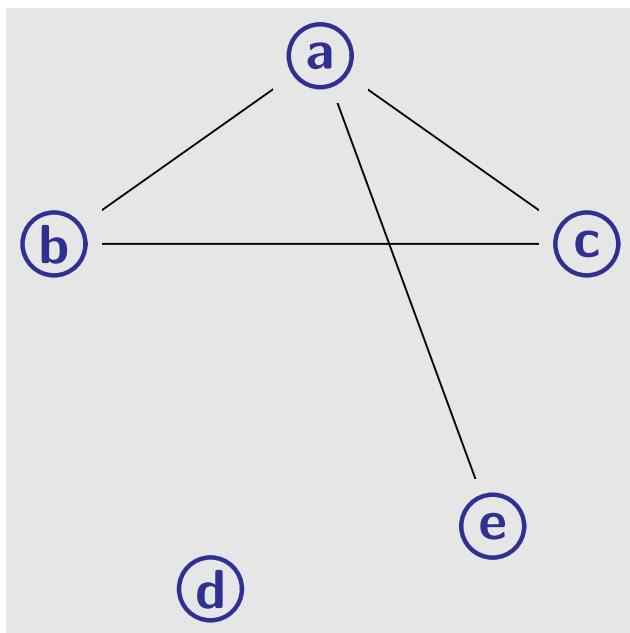
Wierzchołków o nieparzystym stopniu jest parzysta ilość.

Grafy — podstawy

Wniosek:

Wierzchołków o nieparzystym stopniu jest parzysta ilość.

Przykład:



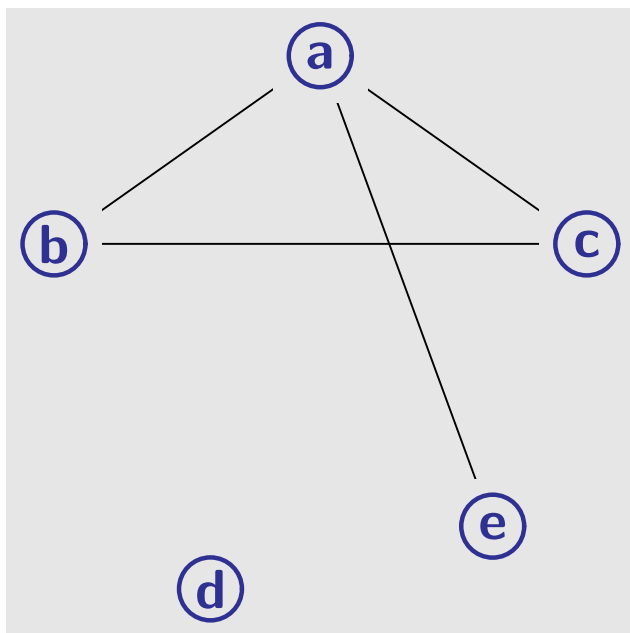
	stopień	parzysty?
<i>a</i>	3	

Grafy — podstawy

Wniosek:

Wierzchołków o nieparzystym stopniu jest parzysta ilość.

Przykład:



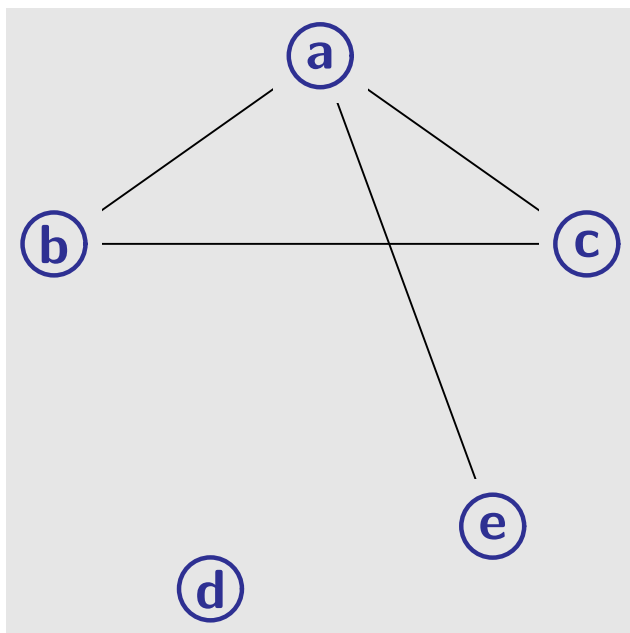
	stopień	parzysty?
<i>a</i>	3	nie

Grafy — podstawy

Wniosek:

Wierzchołków o nieparzystym stopniu jest parzysta ilość.

Przykład:



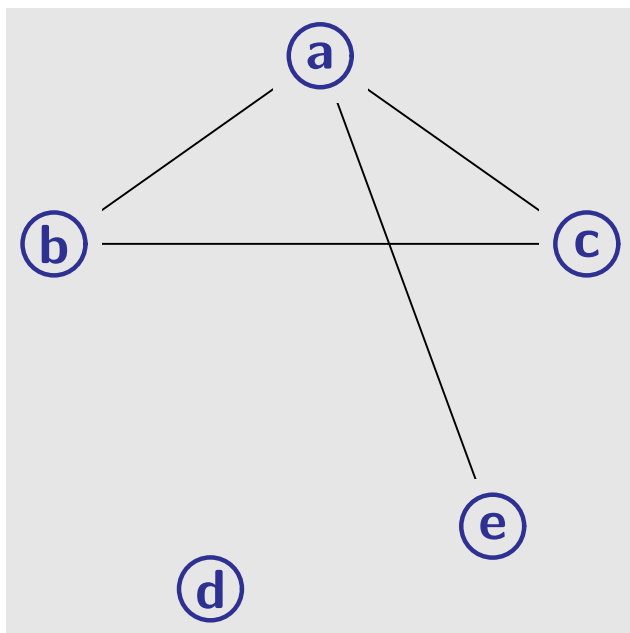
	stopień	parzysty?
<i>a</i>	3	nie
<i>b</i>	2	tak

Grafy — podstawy

Wniosek:

Wierzchołków o nieparzystym stopniu jest parzysta ilość.

Przykład:



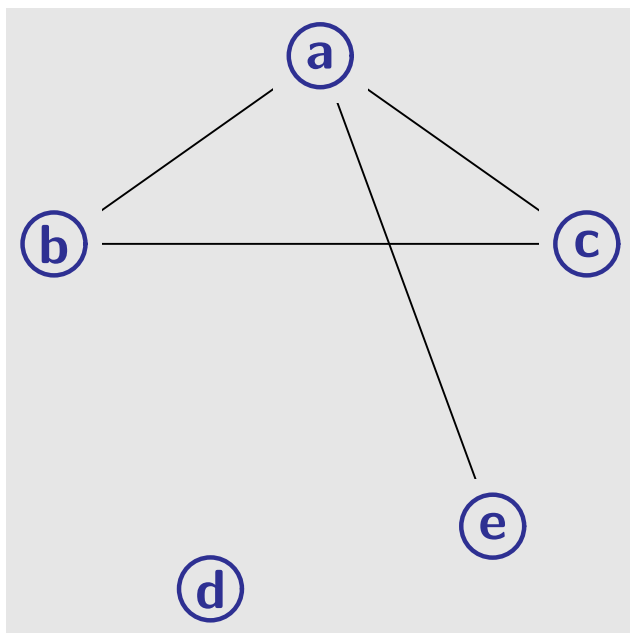
	stopień	parzysty?
<i>a</i>	3	nie
<i>b</i>	2	tak
<i>c</i>	2	tak

Grafy — podstawy

Wniosek:

Wierzchołków o nieparzystym stopniu jest parzysta ilość.

Przykład:



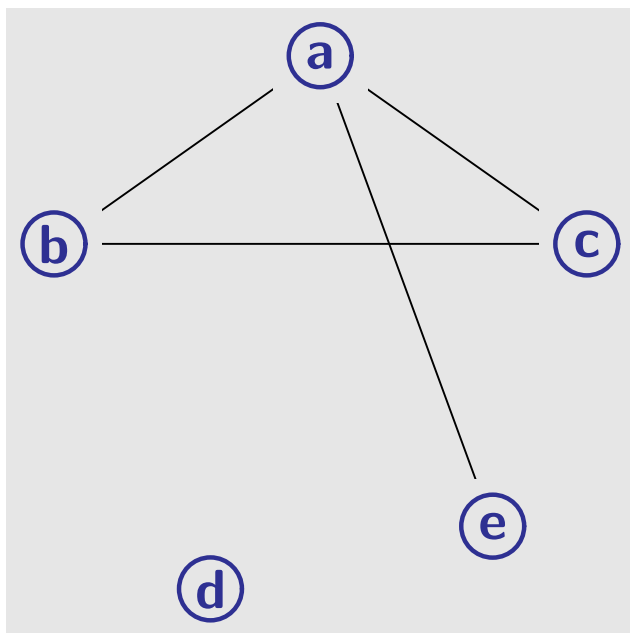
	stopień	parzysty?
<i>a</i>	3	nie
<i>b</i>	2	tak
<i>c</i>	2	tak
<i>d</i>	0	tak

Grafy — podstawy

Wniosek:

Wierzchołków o nieparzystym stopniu jest parzysta ilość.

Przykład:



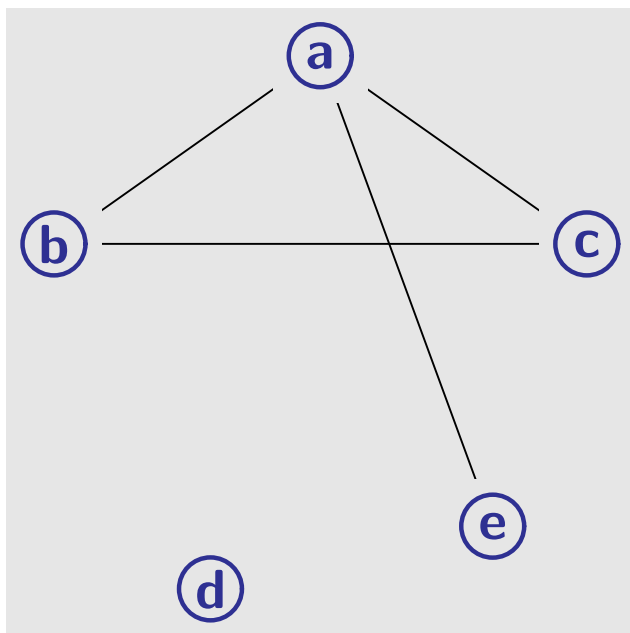
	stopień	parzysty?
<i>a</i>	3	nie
<i>b</i>	2	tak
<i>c</i>	2	tak
<i>d</i>	0	tak
<i>e</i>	1	nie

Grafy — podstawy

Wniosek:

Wierzchołków o nieparzystym stopniu jest parzysta ilość.

Przykład:



	stopień	parzysty?
<i>a</i>	3	nie
<i>b</i>	2	tak
<i>c</i>	2	tak
<i>d</i>	0	tak
<i>e</i>	1	nie
	suma: 8	2 nieparzyste

Grafy — podstawy

Definicja:

Graf $G = \langle V, E \rangle$ jest r -regularny, jeśli

$$\forall_{v \in V} d_G(v) = r$$

Grafy — podstawy

Definicja:

Graf $G = \langle V, E \rangle$ jest r -regularny, jeśli

$$\forall_{v \in V} d_G(v) = r$$

czyli jeśli wszystkie jego wierzchołki mają stopień r .

Grafy — podstawy

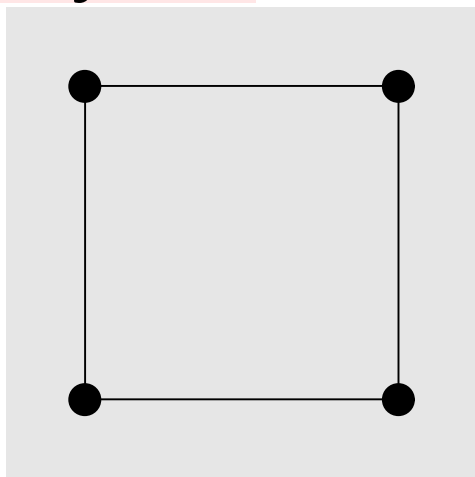
Definicja:

Graf $G = \langle V, E \rangle$ jest r -regularny, jeśli

$$\forall v \in V \ d_G(v) = r$$

czyli jeśli wszystkie jego wierzchołki mają stopień r .

Przykład:



graf 2-regularny

Grafy — podstawy

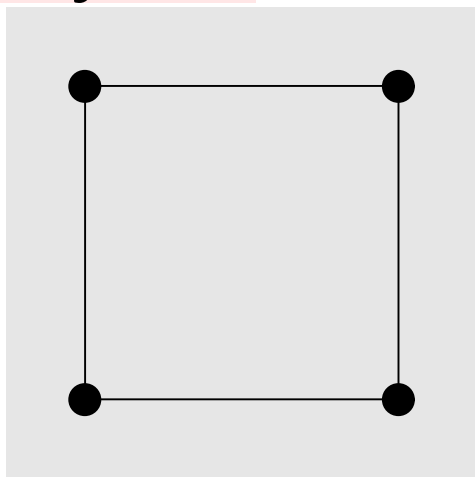
Definicja:

Graf $G = \langle V, E \rangle$ jest r -regularny, jeśli

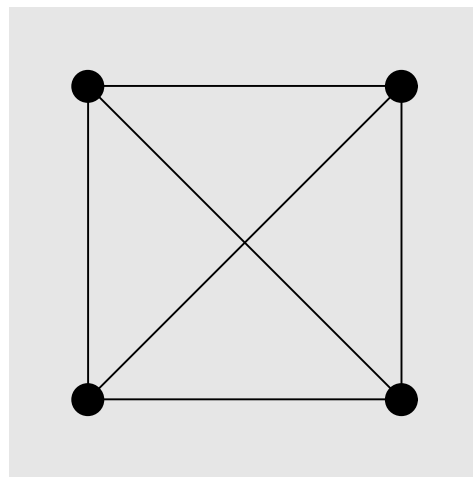
$$\forall v \in V \ d_G(v) = r$$

czyli jeśli wszystkie jego wierzchołki mają stopień r .

Przykład:



graf 2-regularny



graf 3-regularny

Grafy — podstawy

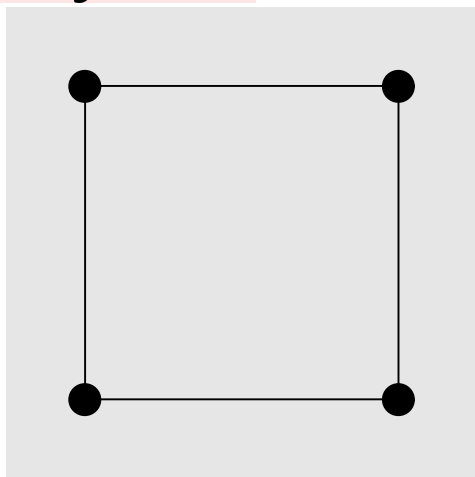
Definicja:

Graf $G = \langle V, E \rangle$ jest r -regularny, jeśli

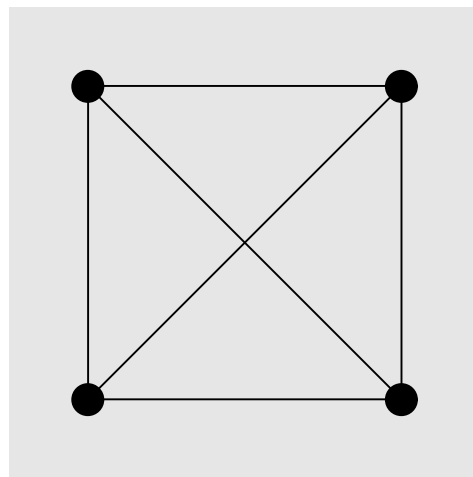
$$\forall v \in V \ d_G(v) = r$$

czyli jeśli wszystkie jego wierzchołki mają stopień r .

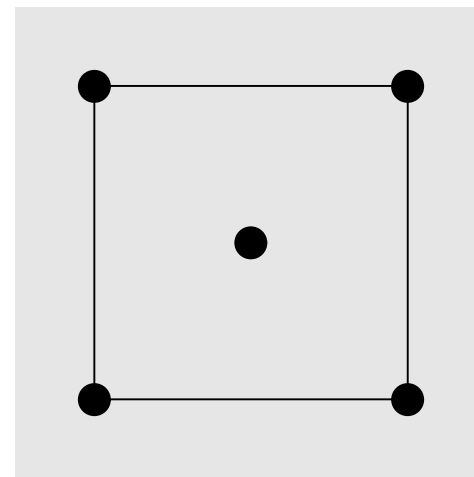
Przykład:



graf 2-regularny



graf 3-regularny



graf nie r -regularny
dla żadnego r

Grafy — podstawy

Definicja:

Graf $G = \langle V, E \rangle$ jest **pełny**, jeśli

$$\forall_{v,u \in V} \{v, u\} \in E$$

Grafy — podstawy

Definicja:

Graf $G = \langle V, E \rangle$ jest **pełny**, jeśli

$$\forall_{v,u \in V} \{v, u\} \in E$$

czyli jeśli każde jego dwa różne wierzchołki łączy krawędź.

Grafy — podstawy

Definicja:

Graf $G = \langle V, E \rangle$ jest **pełny**, jeśli

$$\forall_{v,u \in V} \{v, u\} \in E$$

czyli jeśli każde jego dwa różne wierzchołki łączy krawędź.

Oznaczenie: K_n graf pełny o n wierzchołkach.

Grafy — podstawy

Definicja:

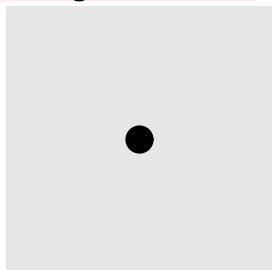
Graf $G = \langle V, E \rangle$ jest **pełny**, jeśli

$$\forall_{v,u \in V} \{v, u\} \in E$$

czyli jeśli każde jego dwa różne wierzchołki łączy krawędź.

Oznaczenie: K_n graf pełny o n wierzchołkach.

Przykład:



K_1

Grafy — podstawy

Definicja:

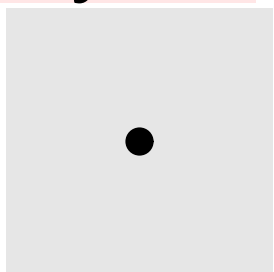
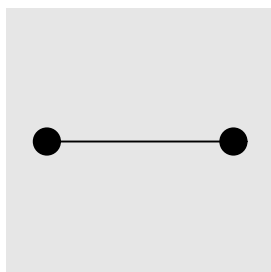
Graf $G = \langle V, E \rangle$ jest **pełny**, jeśli

$$\forall v, u \in V \quad \{v, u\} \in E$$

czyli jeśli każde jego dwa różne wierzchołki łączy krawędź.

Oznaczenie: K_n graf pełny o n wierzchołkach.

Przykład:

 K_1  K_2

Grafy — podstawy

Definicja:

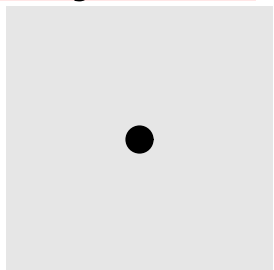
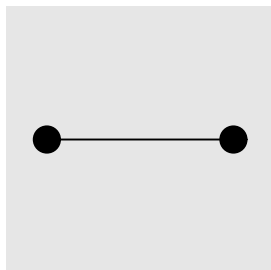
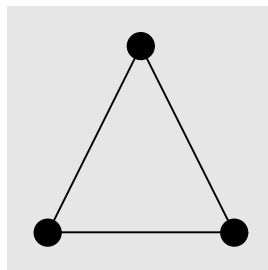
Graf $G = \langle V, E \rangle$ jest **pełny**, jeśli

$$\forall_{v,u \in V} \{v, u\} \in E$$

czyli jeśli każde jego dwa różne wierzchołki łączy krawędź.

Oznaczenie: K_n graf pełny o n wierzchołkach.

Przykład:

 K_1  K_2  K_3

Grafy — podstawy

Definicja:

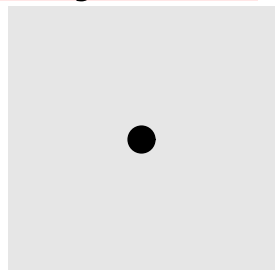
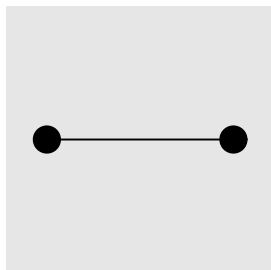
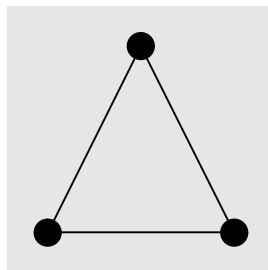
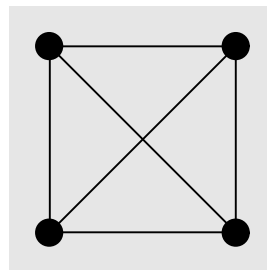
Graf $G = \langle V, E \rangle$ jest **pełny**, jeśli

$$\forall_{v,u \in V} \{v,u\} \in E$$

czyli jeśli każde jego dwa różne wierzchołki łączy krawędź.

Oznaczenie: K_n graf pełny o n wierzchołkach.

Przykład:

 K_1  K_2  K_3  K_4

Grafy — podstawy

Definicja:

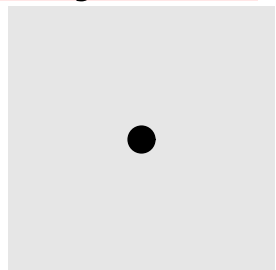
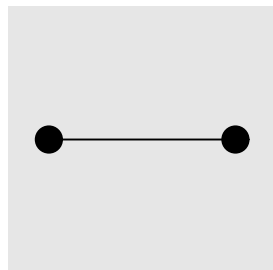
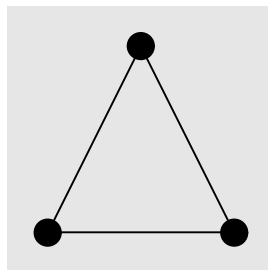
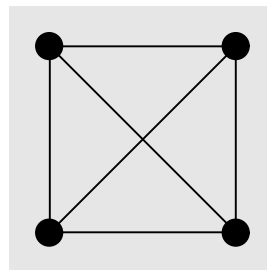
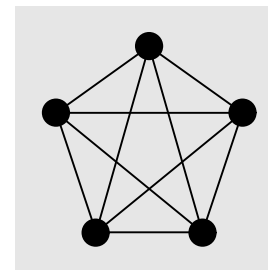
Graf $G = \langle V, E \rangle$ jest **pełny**, jeśli

$$\forall_{v,u \in V} \{v,u\} \in E$$

czyli jeśli każde jego dwa różne wierzchołki łączy krawędź.

Oznaczenie: K_n graf pełny o n wierzchołkach.

Przykład:

 K_1  K_2  K_3  K_4  K_5

Grafy — podstawy

Definicja:

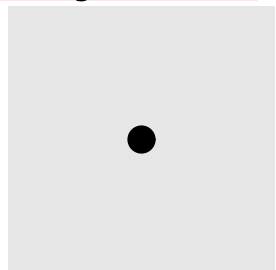
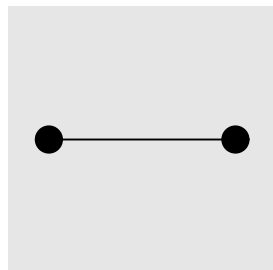
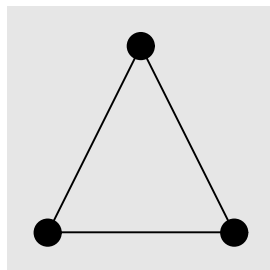
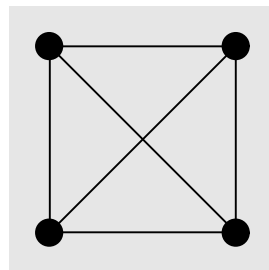
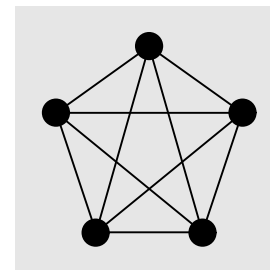
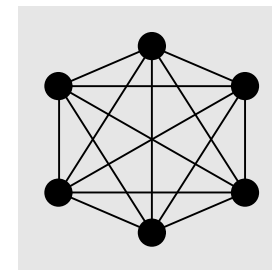
Graf $G = \langle V, E \rangle$ jest **pełny**, jeśli

$$\forall_{v,u \in V} \{v,u\} \in E$$

czyli jeśli każde jego dwa różne wierzchołki łączy krawędź.

Oznaczenie: K_n graf pełny o n wierzchołkach.

Przykład:

 K_1  K_2  K_3  K_4  K_5  K_6

Grafy — podstawy

Definicja:

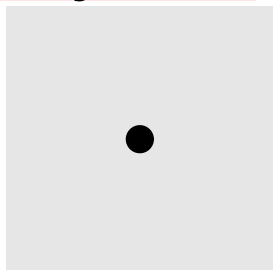
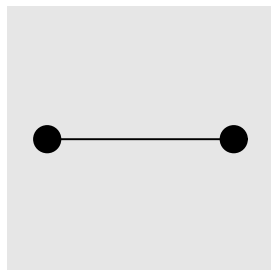
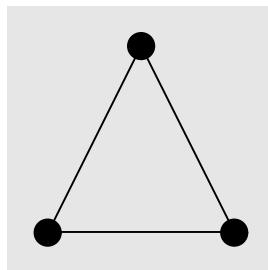
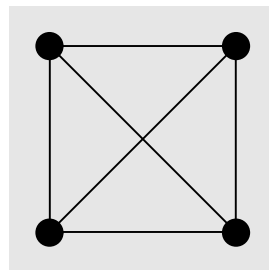
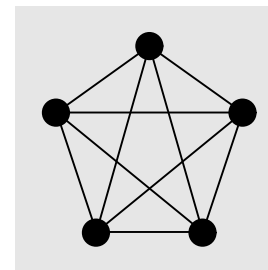
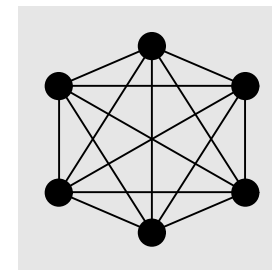
Graf $G = \langle V, E \rangle$ jest **pełny**, jeśli

$$\forall_{v,u \in V} \{v,u\} \in E$$

czyli jeśli każde jego dwa różne wierzchołki łączy krawędź.

Oznaczenie: K_n graf pełny o n wierzchołkach.

Przykład:


 K_1

 K_2

 K_3

 K_4

 K_5

 K_6

Twierdzenie:

Liczba krawędzi grafu pełnego K_n wynosi

$$\binom{n}{2}$$

Grafy — podstawy

Definicja:

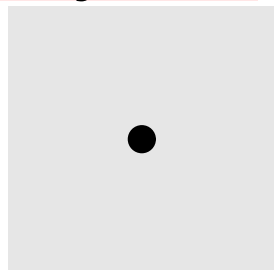
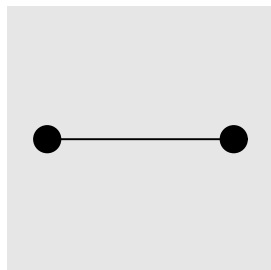
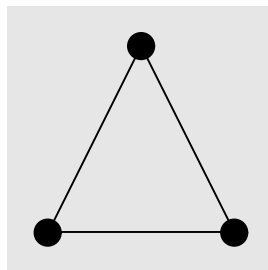
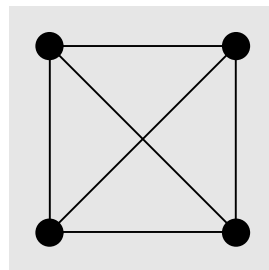
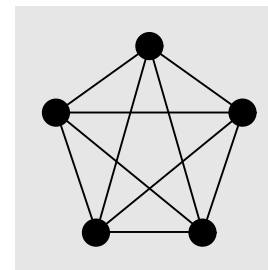
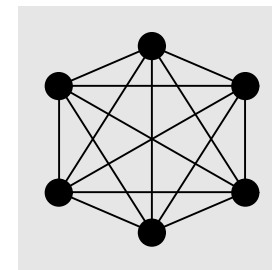
Graf $G = \langle V, E \rangle$ jest **pełny**, jeśli

$$\forall_{v,u \in V} \{v,u\} \in E$$

czyli jeśli każde jego dwa różne wierzchołki łączy krawędź.

Oznaczenie: K_n graf pełny o n wierzchołkach.

Przykład:


 K_1

 K_2

 K_3

 K_4

 K_5

 K_6

Twierdzenie:

Liczba krawędzi grafu pełnego K_n wynosi

$$\binom{n}{2} = \frac{n(n-1)}{2}$$

Grafy — podstawy

Definicja:

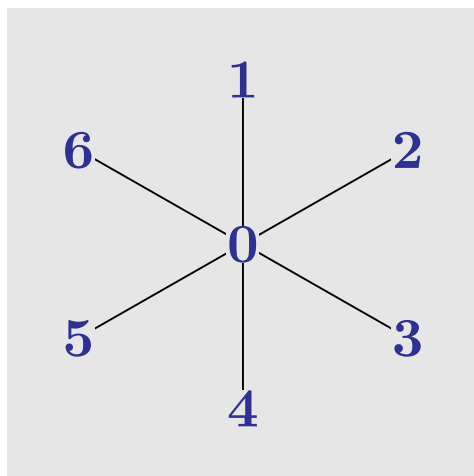
Graf $G = \langle V, E \rangle$ jest **dwudzielny**, jeśli jego zbiór wierzchołków V można podzielić na dwa zbiory rozłączne V_1 i V_2 ($V_1 \cup V_2 = V$ i $V_1 \cap V_2 = \emptyset$),
takie że $\left\{ \begin{array}{l} \text{żadne dwa wierzchołki w } V_1 \\ \text{ani} \\ \text{żadne dwa wierzchołki w } V_2 \end{array} \right\}$ nie są sąsiednie

Grafy — podstawy

Definicja:

Graf $G = \langle V, E \rangle$ jest **dwudzielny**, jeśli jego zbiór wierzchołków V można podzielić na dwa zbiory rozłączne V_1 i V_2 ($V_1 \cup V_2 = V$ i $V_1 \cap V_2 = \emptyset$), takie że $\left\{ \begin{array}{l} \text{żadne dwa wierzchołki w } V_1 \\ \text{ani} \\ \text{żadne dwa wierzchołki w } V_2 \end{array} \right\}$ nie są sąsiednie
czyli jeśli da się pomalować wierzchołki dwoma kolorami tak, że każda krawędź łączy wierzchołki różnych kolorów.

Przykład:



Grafy — podstawy

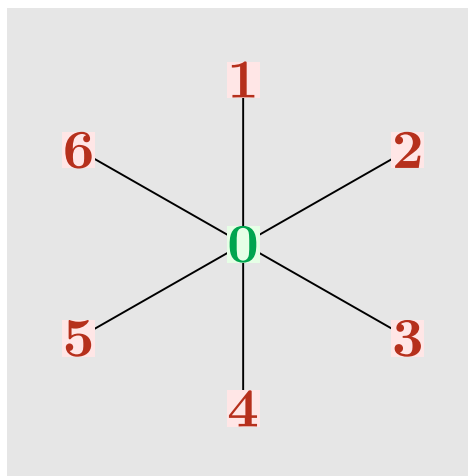
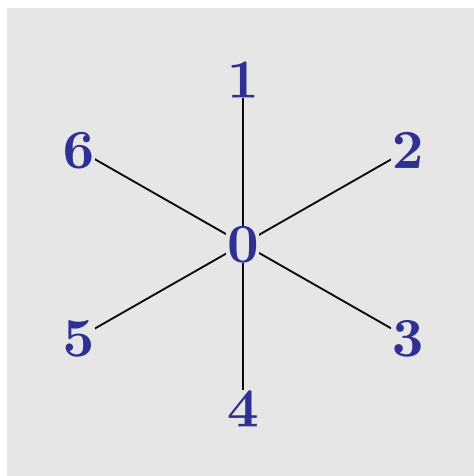
Definicja:

Graf $G = \langle V, E \rangle$ jest **dwudzielny**, jeśli jego zbiór wierzchołków V można podzielić na dwa zbiory rozłączne V_1 i V_2 ($V_1 \cup V_2 = V$ i $V_1 \cap V_2 = \emptyset$), takie że

$$\left\{ \begin{array}{l} \text{żadne dwa wierzchołki w } V_1 \\ \text{ani} \\ \text{żadne dwa wierzchołki w } V_2 \end{array} \right\} \text{ nie są sąsiednie}$$

czyli jeśli da się pomalować wierzchołki dwoma kolorami tak, że każda krawędź łączy wierzchołki różnych kolorów.

Przykład:



Grafy — podstawy

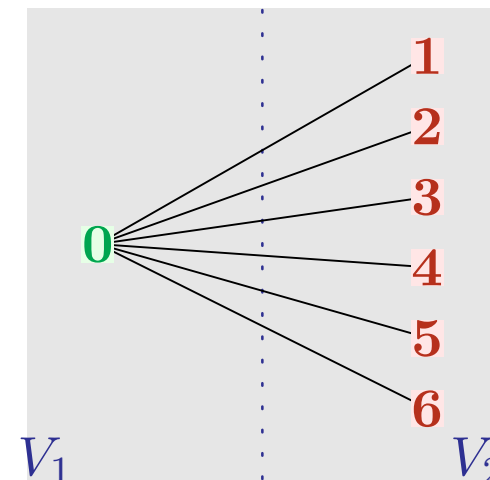
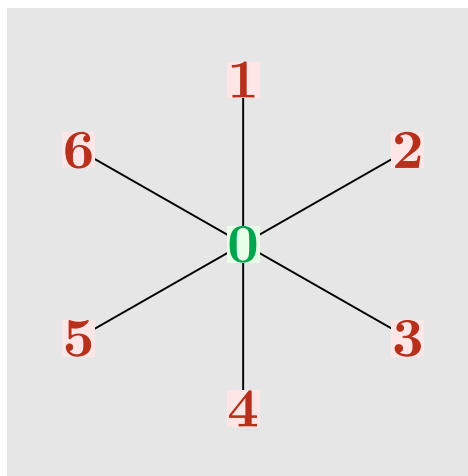
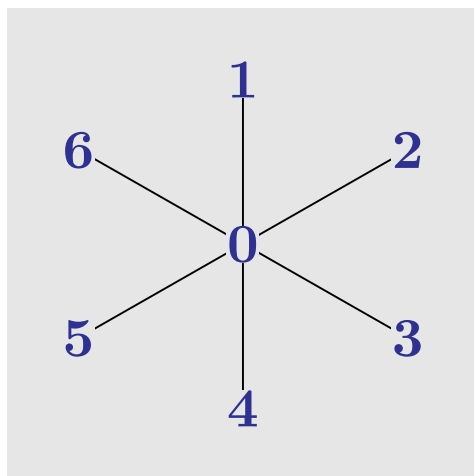
Definicja:

Graf $G = \langle V, E \rangle$ jest **dwudzielny**, jeśli jego zbiór wierzchołków V można podzielić na dwa zbiory rozłączne V_1 i V_2 ($V_1 \cup V_2 = V$ i $V_1 \cap V_2 = \emptyset$), takie że

$$\left\{ \begin{array}{l} \text{żadne dwa wierzchołki w } V_1 \\ \text{ani} \\ \text{żadne dwa wierzchołki w } V_2 \end{array} \right\} \text{ nie są sąsiednie}$$

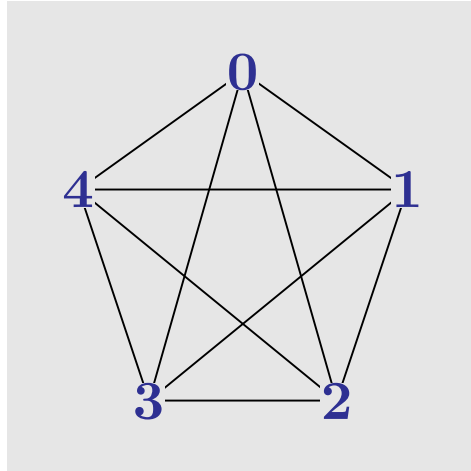
czyli jeśli da się pomalować wierzchołki dwoma kolorami tak, że każda krawędź łączy wierzchołki różnych kolorów.

Przykład:



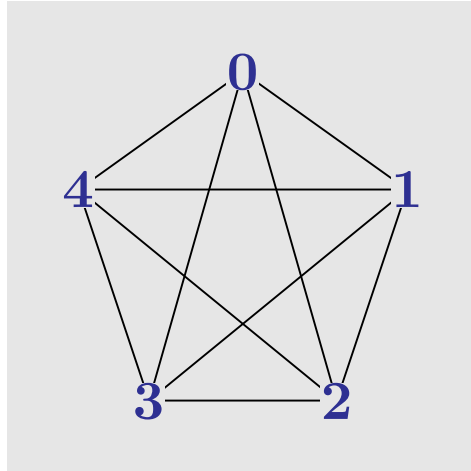
Grafy — podstawy

Przykład:



Grafy — podstawy

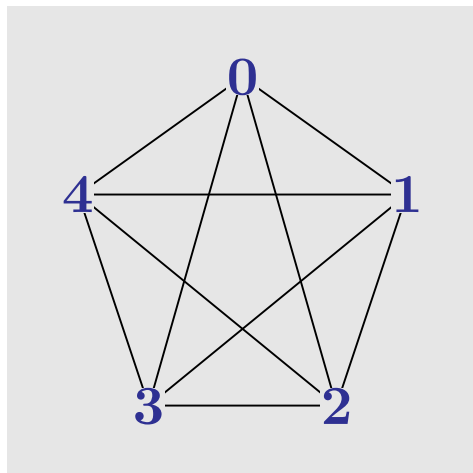
Przykład:



nie dwudzielny

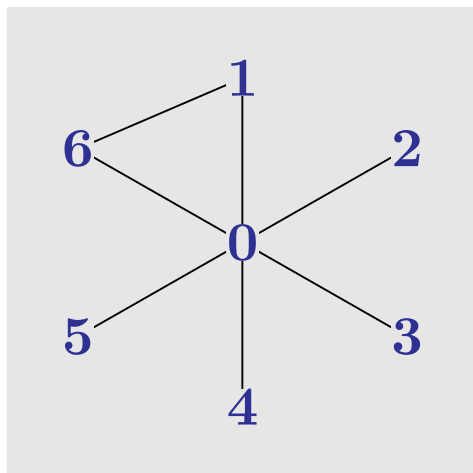
Grafy — podstawy

Przykład:



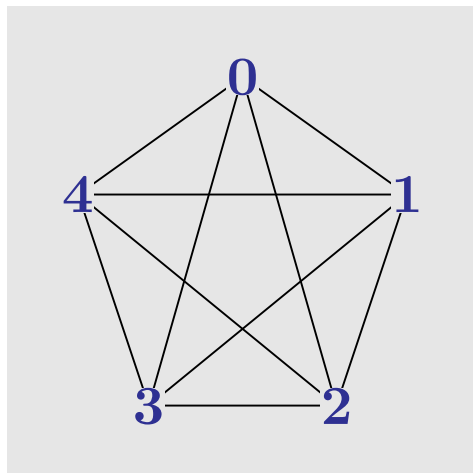
nie dwudzielny

Przykład:



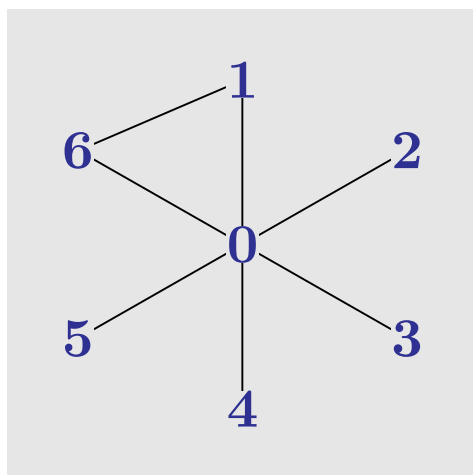
Grafy — podstawy

Przykład:



nie dwudzielny

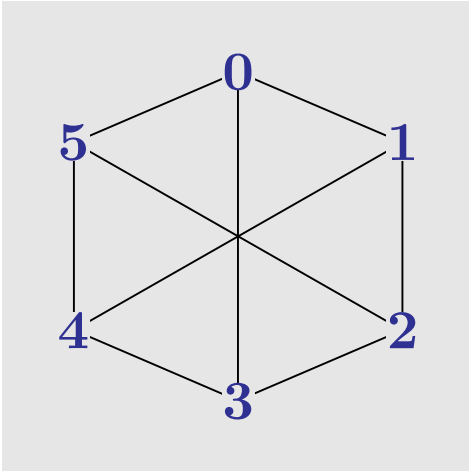
Przykład:



nie dwudzielny

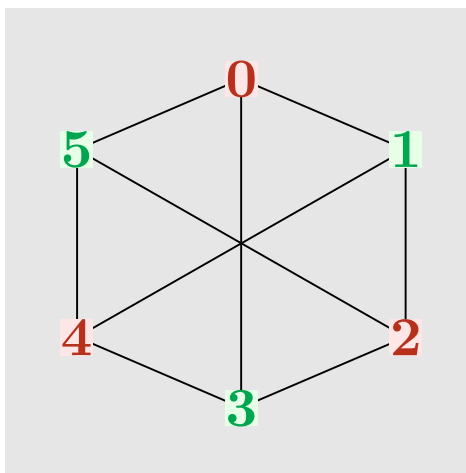
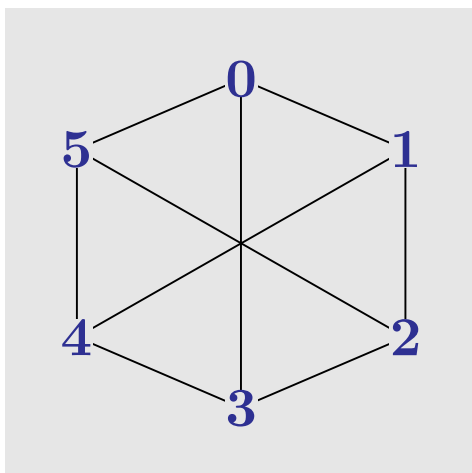
Grafy — podstawy

Przykład:



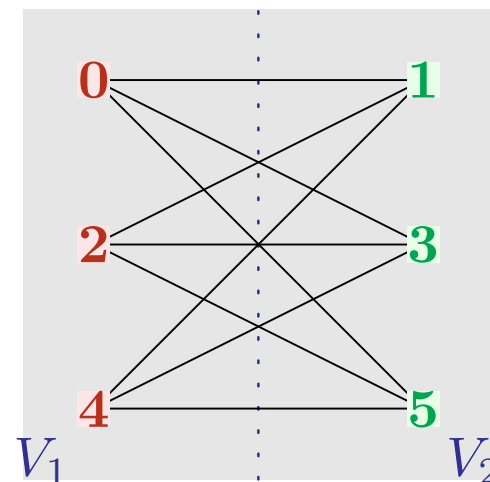
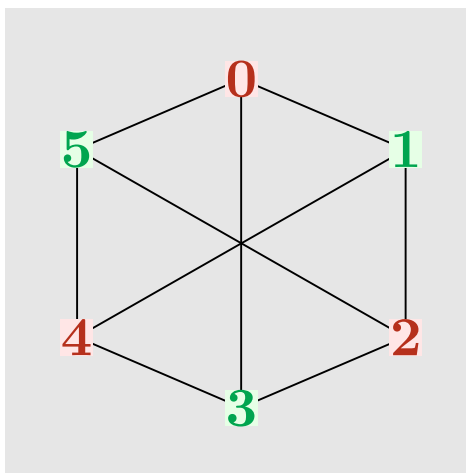
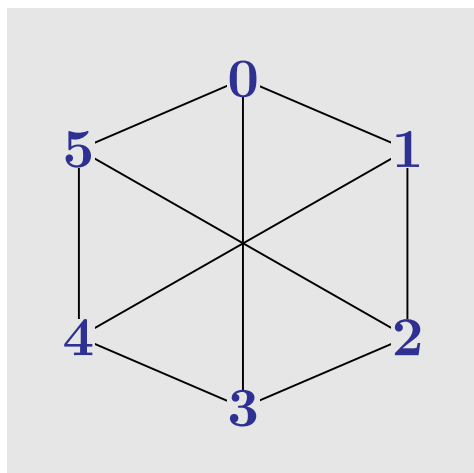
Grafy — podstawy

Przykład:



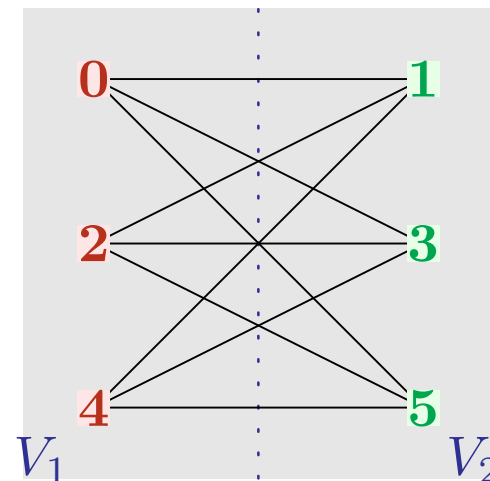
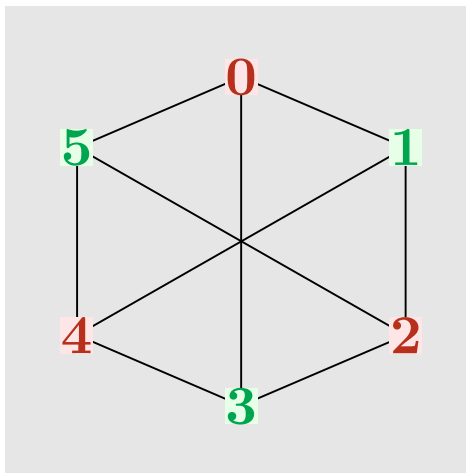
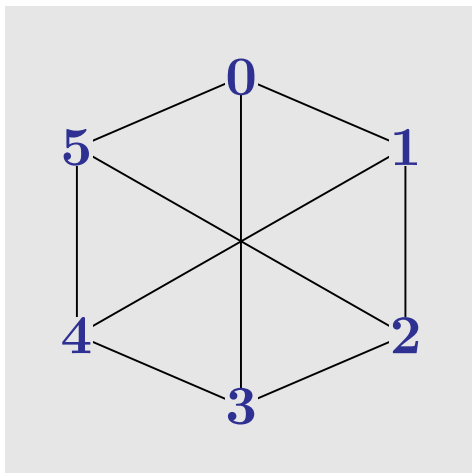
Grafy — podstawy

Przykład:

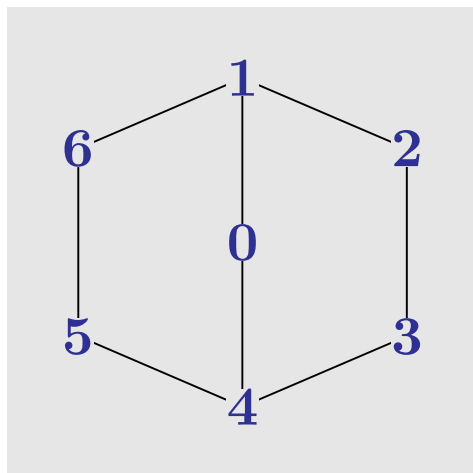


Grafy — podstawy

Przykład:

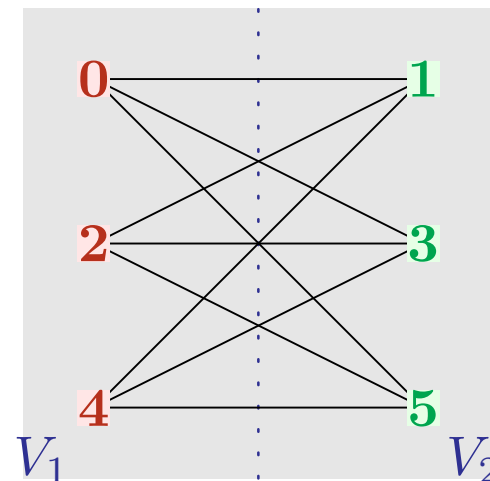
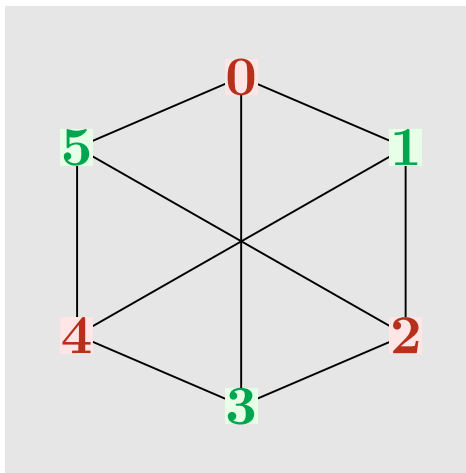
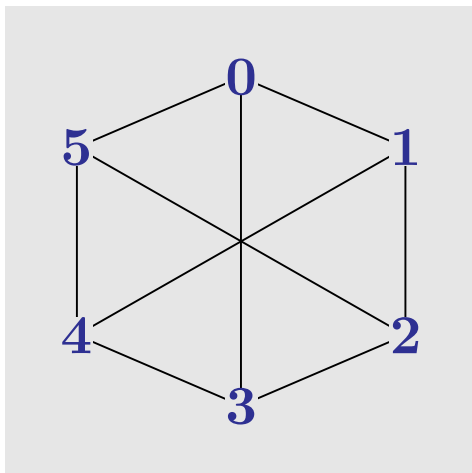


Przykład:

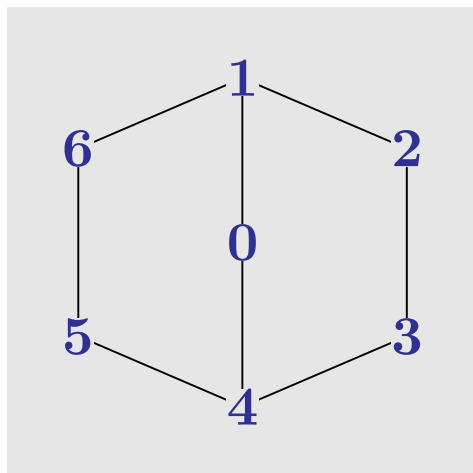


Grafy — podstawy

Przykład:



Przykład:



nie dwudzielny

Grafy — podstawy

Definicja:

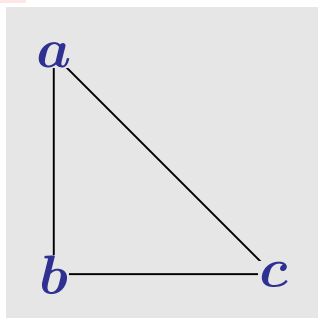
Podgraf grafu $G = \langle V, E \rangle$ to dowolny taki graf $G' = \langle V', E' \rangle$, że
 $V' \subseteq V$ oraz $E' \subseteq E$

Grafy — podstawy

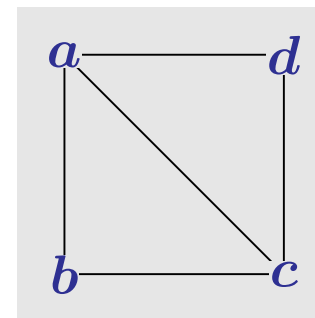
Definicja:

Podgraf grafu $G = \langle V, E \rangle$ to dowolny taki graf $G' = \langle V', E' \rangle$, że
 $V' \subseteq V$ oraz $E' \subseteq E$

Przykład:



jest podgrafem grafu

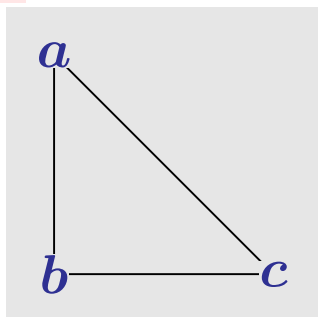


Grafy — podstawy

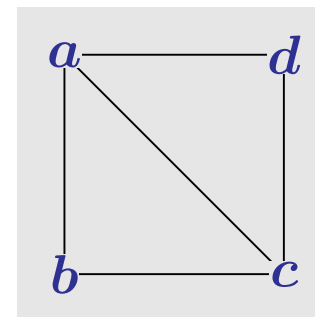
Definicja:

Podgraf grafu $G = \langle V, E \rangle$ to dowolny taki graf $G' = \langle V', E' \rangle$, że
 $V' \subseteq V$ oraz $E' \subseteq E$

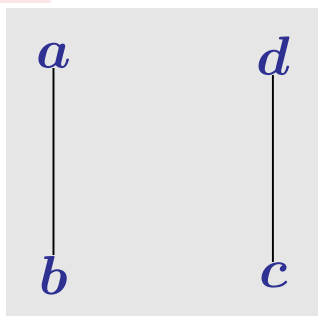
Przykład:



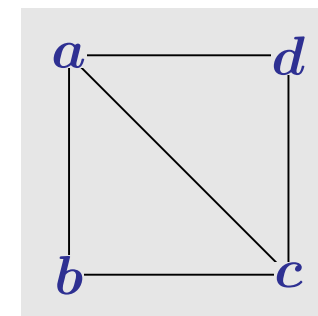
jest podgrafem grafu



Przykład:



też jest podgrafem grafu



Grafy — podstawy

Definicja:

Graf $G = \langle V, E \rangle$ jest **izomorficzny** z grafem $G' = \langle V', E' \rangle$ jeśli istnieje bijekcja (czyli funkcja różnowartościowa i „na”) $\varphi : V \rightarrow V'$ zachowująca krawędzie:

$$\forall_{v,u \in V} (\{v, u\} \in E \iff \{\varphi(v), \varphi(u)\} \in E')$$

Grafy — podstawy

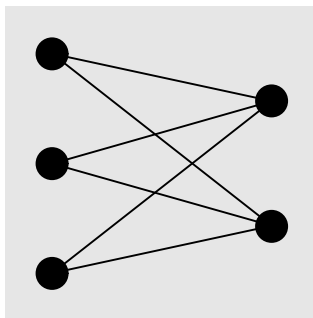
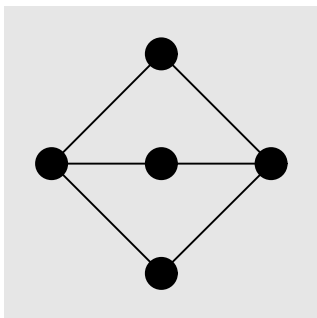
Definicja:

Graf $G = \langle V, E \rangle$ jest **izomorficzny** z grafem $G' = \langle V', E' \rangle$ jeśli istnieje bijekcja (czyli funkcja różnowartościowa i „na”) $\varphi : V \rightarrow V'$ zachowująca krawędzie:

$$\forall_{v,u \in V} (\{v, u\} \in E \iff \{\varphi(v), \varphi(u)\} \in E')$$

Przykład:

Izomorficzne:



Grafy — podstawy

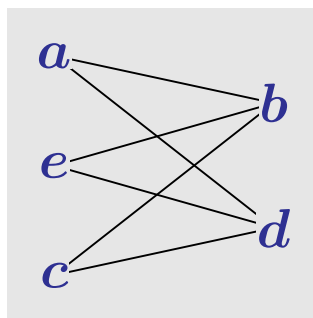
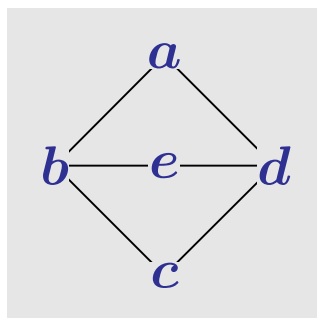
Definicja:

Graf $G = \langle V, E \rangle$ jest **izomorficzny** z grafem $G' = \langle V', E' \rangle$ jeśli istnieje bijekcja (czyli funkcja różnowartościowa i „na”) $\varphi : V \rightarrow V'$ zachowująca krawędzie:

$$\forall_{v,u \in V} (\{v, u\} \in E \iff \{\varphi(v), \varphi(u)\} \in E')$$

Przykład:

Izomorficzne:



Grafy — podstawy

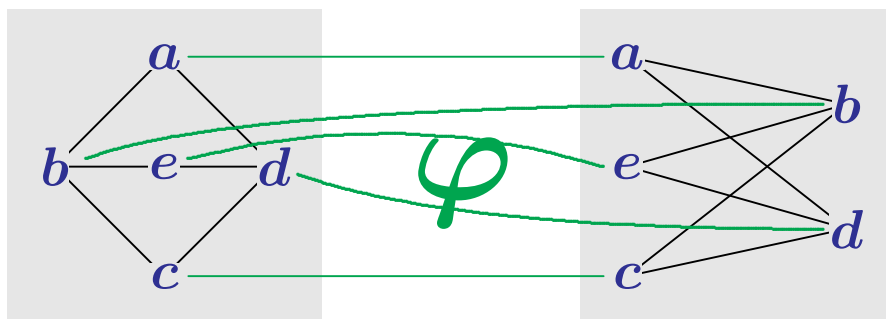
Definicja:

Graf $G = \langle V, E \rangle$ jest **izomorficzny** z grafem $G' = \langle V', E' \rangle$ jeśli istnieje bijekcja (czyli funkcja różnowartościowa i „na”) $\varphi : V \rightarrow V'$ zachowująca krawędzie:

$$\forall_{v,u \in V} (\{v, u\} \in E \iff \{\varphi(v), \varphi(u)\} \in E')$$

Przykład:

Izomorficzne:



Grafy — podstawy

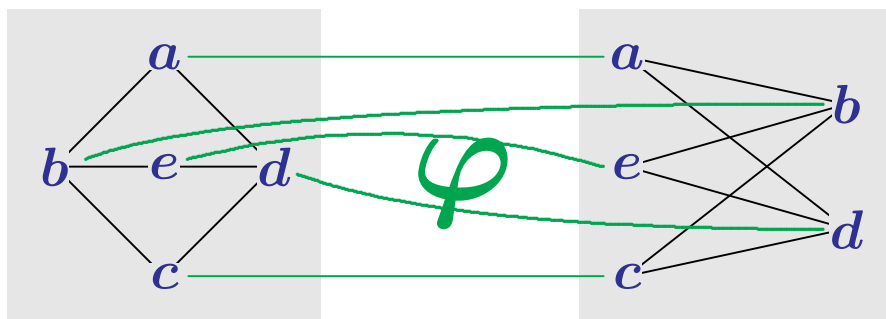
Definicja:

Graf $G = \langle V, E \rangle$ jest **izomorficzny** z grafem $G' = \langle V', E' \rangle$ jeśli istnieje bijekcja (czyli funkcja różnowartościowa i „na”) $\varphi : V \rightarrow V'$ zachowująca krawędzie:

$$\forall_{v,u \in V} (\{v, u\} \in E \iff \{\varphi(v), \varphi(u)\} \in E')$$

Przykład:

Izomorficzne:



Definicja:

ℓ_n — liczba nieizomorficznych grafów o n wierzchołkach

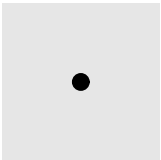
Grafy — podstawy

grafy o n wierzchołkach i m krawędziach							
n				$m =$			ℓ_n
	0	1	2	3	4	5	6

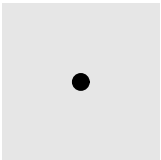
Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							l_n
	0	1	2	3	4	5	6	
1								

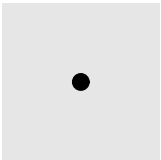
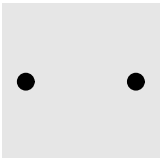
Grafy — podstawy

grafy o n wierzchołkach i m krawędziach								
n	0	1	2	3	4	5	6	ℓ_n
1								

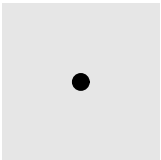
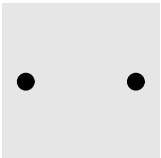
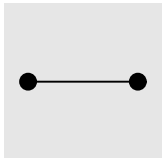
Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								

Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								

Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								

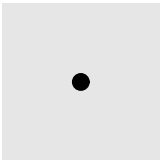
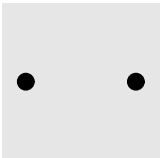
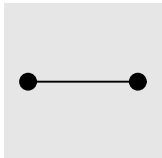
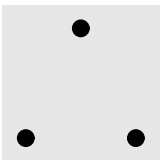
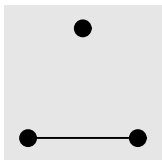
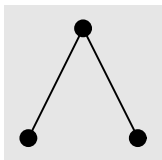
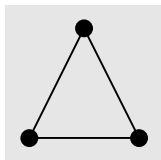
[illegible]

[illegible]

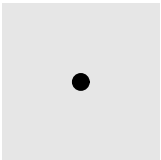
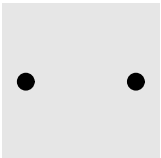
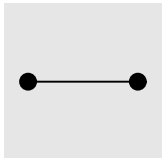
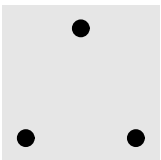
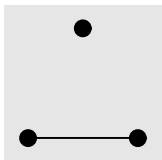
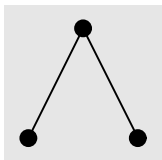
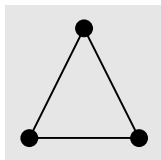
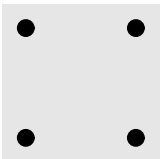
[illegible]

[illegible]

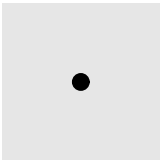
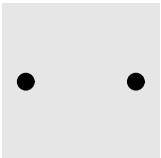
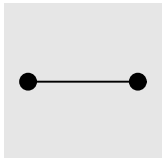
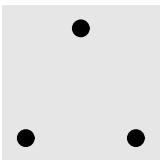
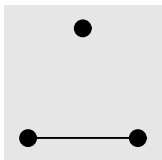
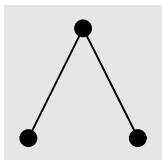
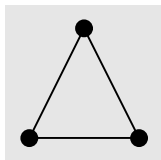
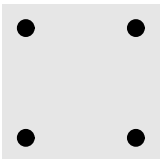
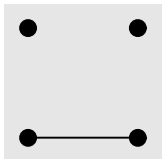
Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								2
3								4
4								

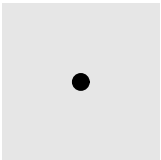
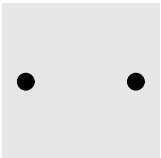
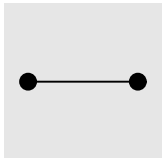
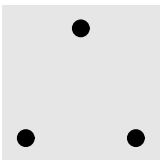
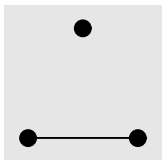
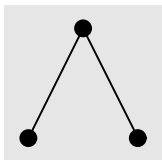
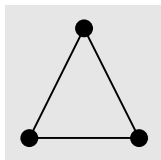
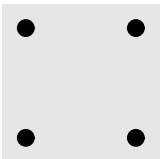
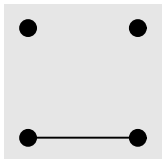
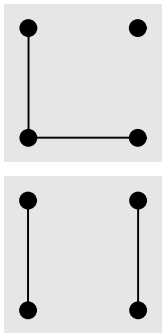
Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								2
3								4
4								

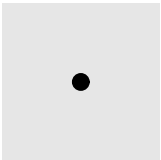
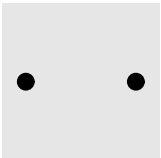
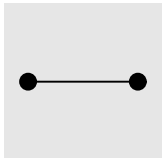
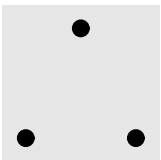
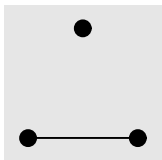
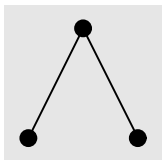
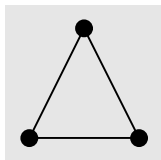
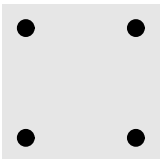
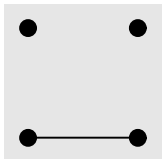
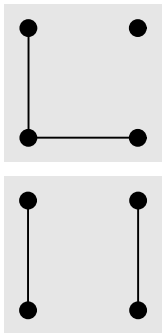
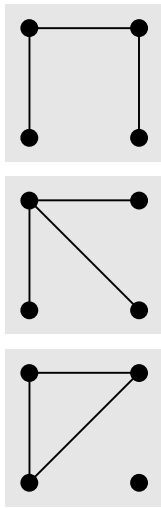
Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								2
3								4
4								

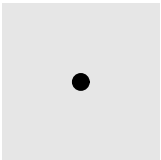
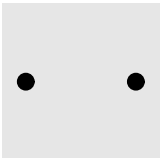
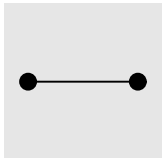
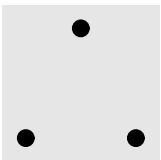
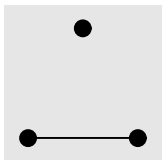
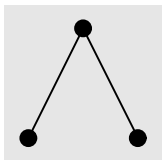
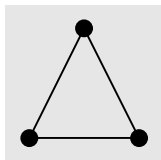
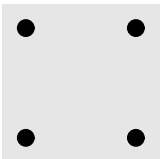
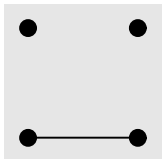
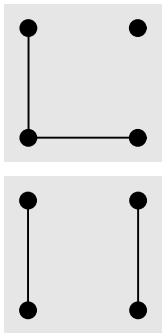
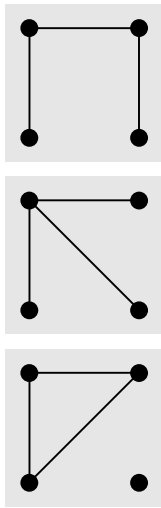
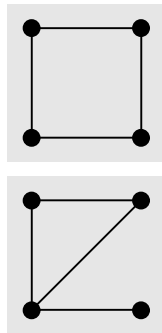
Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								2
3								4
4								

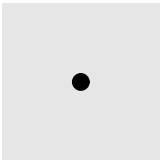
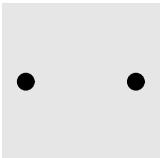
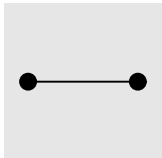
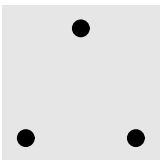
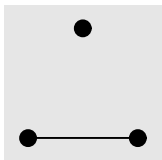
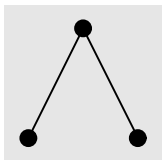
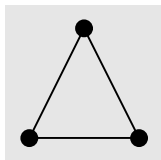
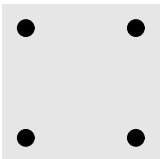
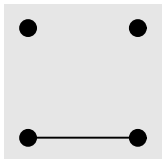
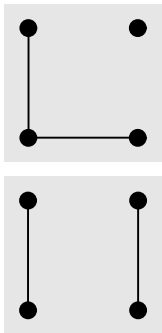
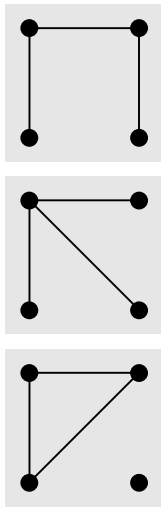
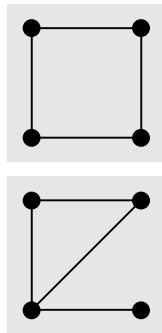
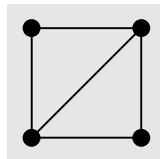
Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								2
3								4
4								

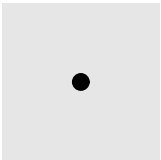
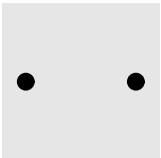
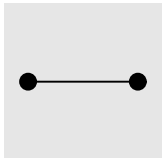
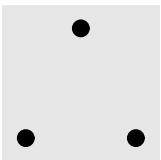
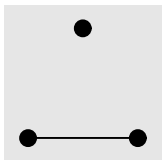
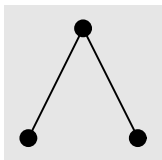
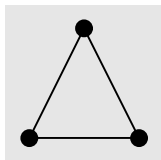
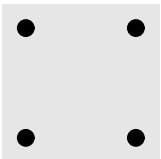
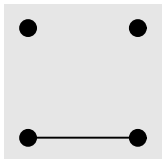
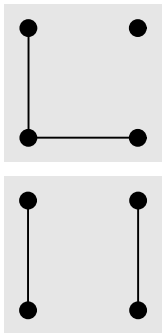
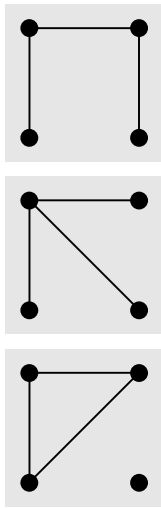
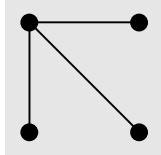
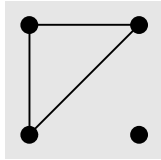
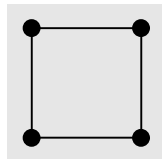
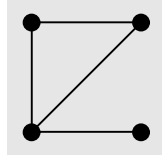
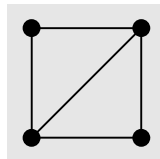
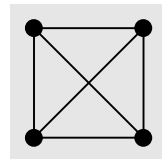
Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								2
3								4
4								

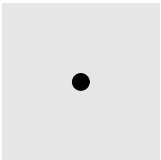
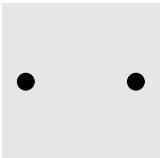
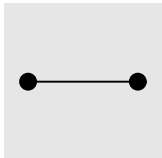
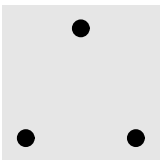
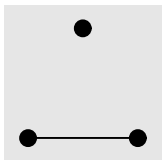
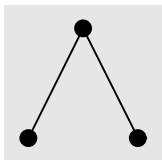
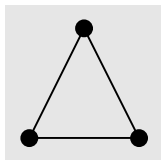
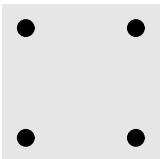
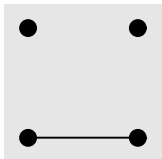
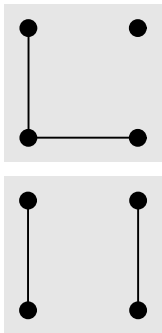
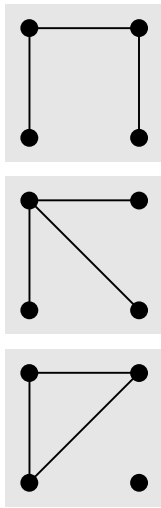
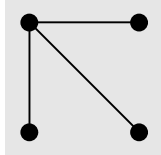
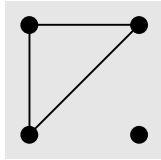
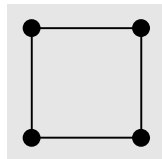
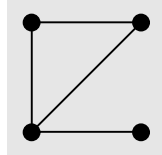
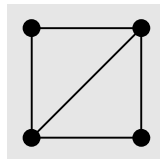
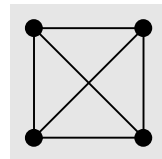
Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								2
3								4
4								

Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								2
3								4
4				  	 			

Grafy — podstawy

n	grafy o n wierzchołkach i m krawędziach							ℓ_n
	0	1	2	3	4	5	6	
1								1
2								2
3								4
4				  	 			11

Drogi w grafie

Definicja:

Droga — taki ciąg $(v_0, v_1, \dots, v_d)$ wierzchołków, że
 $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{d-1}, v_d\}$
są krawędziami.

Drogi w grafie

Definicja:

Droga — taki ciąg $(v_0, v_1, \dots, v_d)$ wierzchołków, że $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{d-1}, v_d\}$ są krawędziami.

v_0 — **początek** drogi

Drogi w grafie

Definicja:

Droga — taki ciąg $(v_0, v_1, \dots, v_d)$ wierzchołków, że
 $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{d-1}, v_d\}$
są krawędziami.

v_0 — **początek** drogi

v_d — **koniec** drogi

Drogi w grafie

Definicja:

Droga — taki ciąg $(v_0, v_1, \dots, v_d)$ wierzchołków, że $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{d-1}, v_d\}$ są krawędziami.

v_0 — **początek** drogi

v_d — **koniec** drogi

d — **długość** drogi.

Drogi w grafie

Definicja:

Droga — taki ciąg $(v_0, v_1, \dots, v_d)$ wierzchołków, że $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{d-1}, v_d\}$ są krawędziami.

v_0 — **początek** drogi

v_d — **koniec** drogi

d — **długość** drogi.

Droga jest **zamknięta** $\iff v_d = v_0$.

Drogi w grafie

Definicja:

Droga — taki ciąg $(v_0, v_1, \dots, v_d)$ wierzchołków, że $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{d-1}, v_d\}$ są krawędziami.

v_0 — **początek** drogi

v_d — **koniec** drogi

d — **długość** drogi.

Droga jest **zamknięta** $\iff v_d = v_0$.

Droga jest **prosta** $\iff$ żadna krawędź nie występuje w niej dwukrotnie

Drogi w grafie

Definicja:

Droga — taki ciąg $(v_0, v_1, \dots, v_d)$ wierzchołków, że $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{d-1}, v_d\}$ są krawędziami.

v_0 — **początek** drogi

v_d — **koniec** drogi

d — **długość** drogi.

Droga jest **zamknięta** $\iff v_d = v_0$.

Droga jest **prosta** $\iff$ żadna krawędź nie występuje w niej dwukrotnie
(nie idziemy ponownie po tej samej krawędzi)

Drogi w grafie

Definicja:

Droga — taki ciąg $(v_0, v_1, \dots, v_d)$ wierzchołków, że
 $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{d-1}, v_d\}$
 są krawędziami.

v_0 — **początek** drogi

v_d — **koniec** drogi

d — **długość** drogi.

Droga jest **zamknięta** $\iff v_d = v_0$.

Droga jest **prosta** $\iff$ żadna krawędź nie występuje w niej dwukrotnie
 (nie idziemy ponownie po tej samej krawędzi)

Droga jest **ścieżką** $\iff$ żaden wierzchołek poza (być może) $v_0 = v_d$
 nie występuje w niej dwukrotnie

Drogi w grafie

Definicja:

Droga — taki ciąg $(v_0, v_1, \dots, v_d)$ wierzchołków, że
 $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{d-1}, v_d\}$
są krawędziami.

v_0 — **początek** drogi

v_d — **koniec** drogi

d — **długość** drogi.

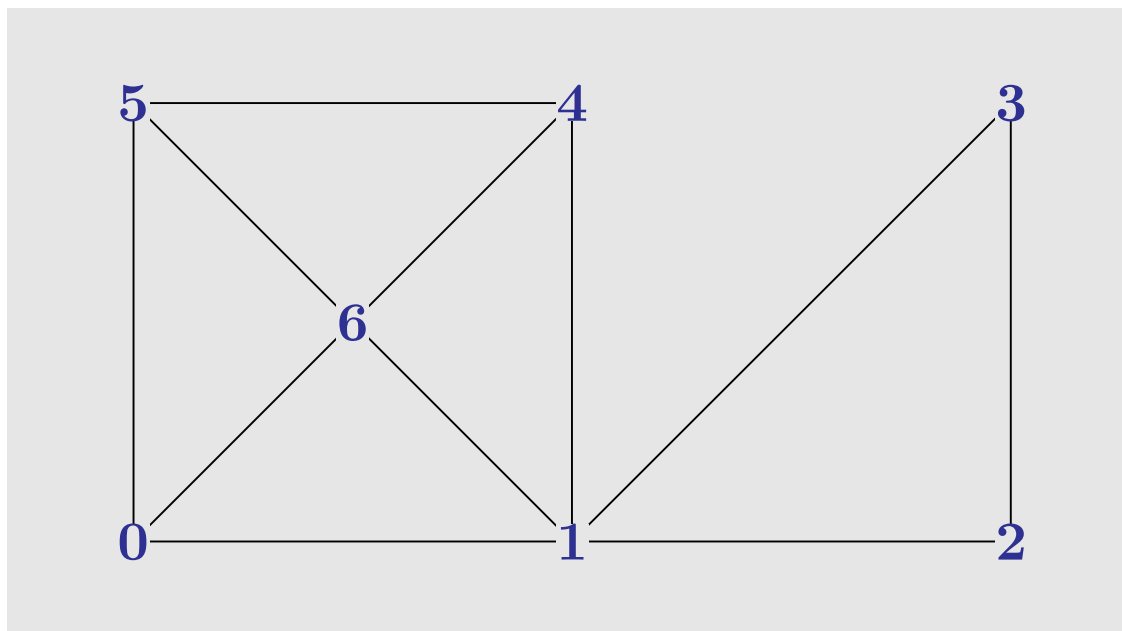
Droga jest **zamknięta** $\iff v_d = v_0$.

Droga jest **prosta** $\iff$ żadna krawędź nie występuje w niej dwukrotnie
(nie idziemy ponownie po tej samej krawędzi)

Droga jest **ścieżką** $\iff$ żaden wierzchołek poza (być może) $v_0 = v_d$
nie występuje w niej dwukrotnie
(nie przechodzimy ponownie przez ten sam wierzchołek)

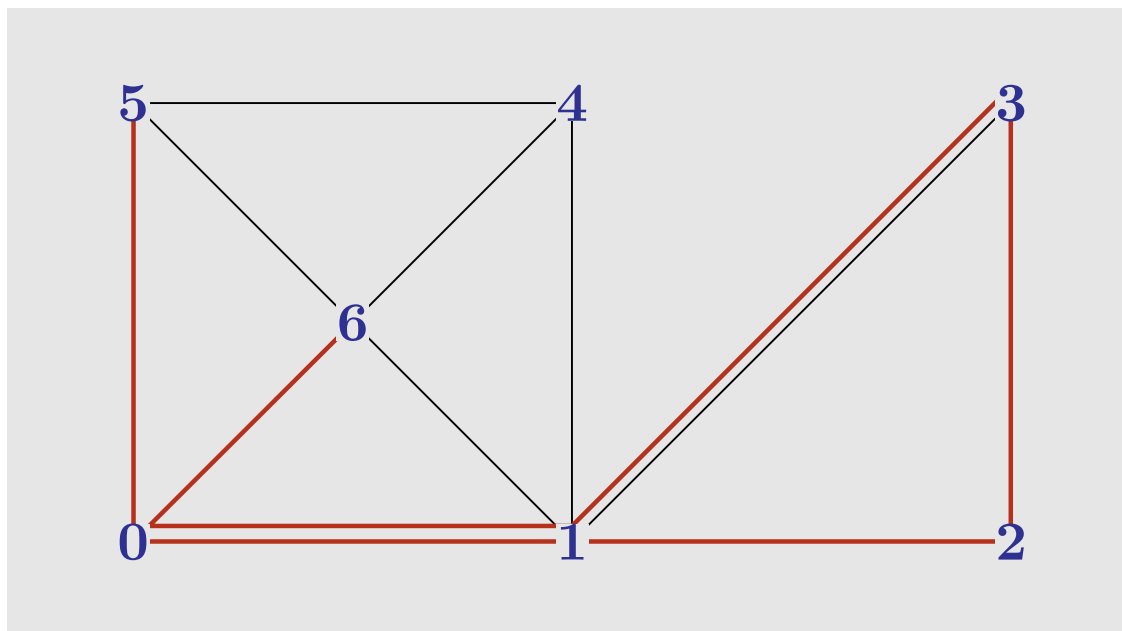
Drogi w grafie

Przykład:



Drogi w grafie

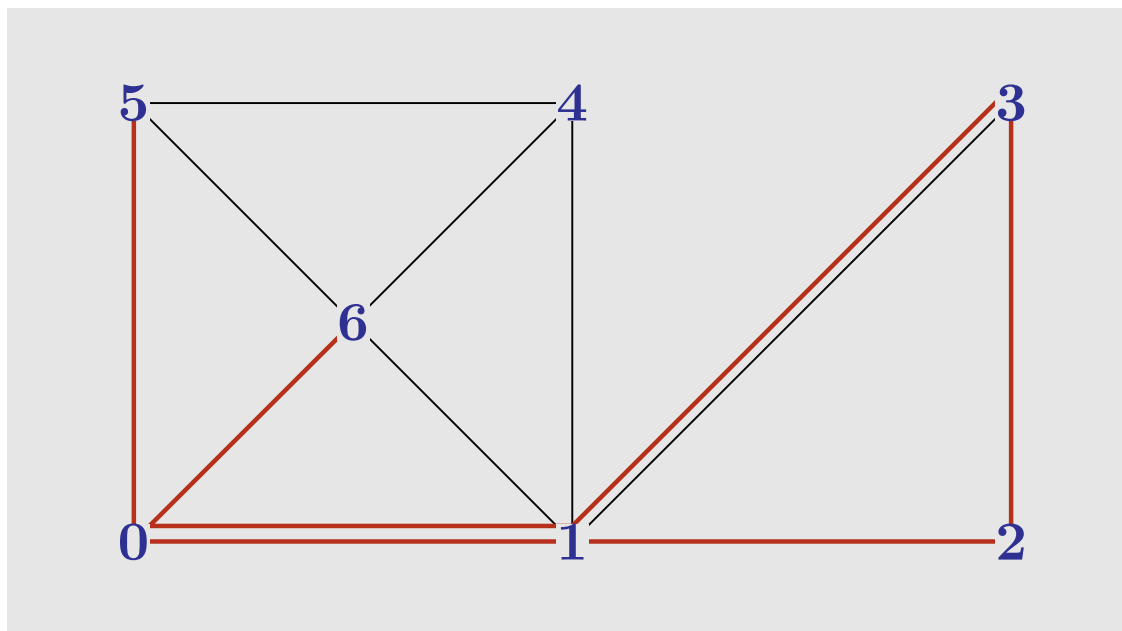
Przykład:



droga — (czyli $(5, 0, 1, 2, 3, 1, 0, 6)$) *nie jest prosta*

Drogi w grafie

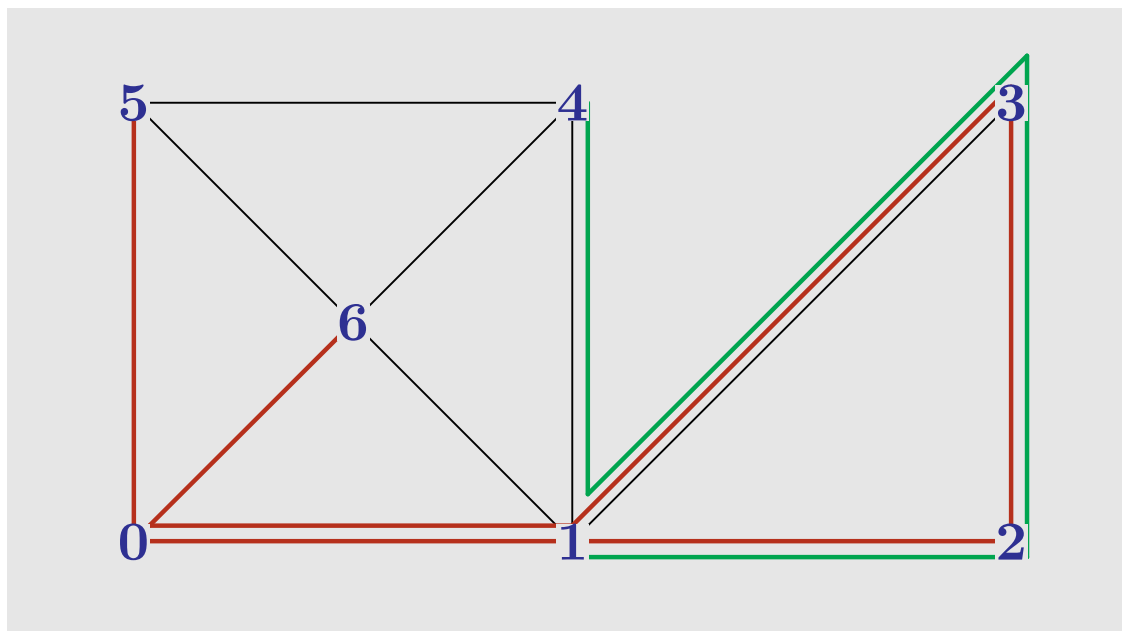
Przykład:



droga — (czyli $(5, 0, 1, 2, 3, 1, 0, 6)$) *nie jest prosta*,
bo dwukrotnie przechodzi przez krawędź $\{0, 1\}$

Drogi w grafie

Przykład:

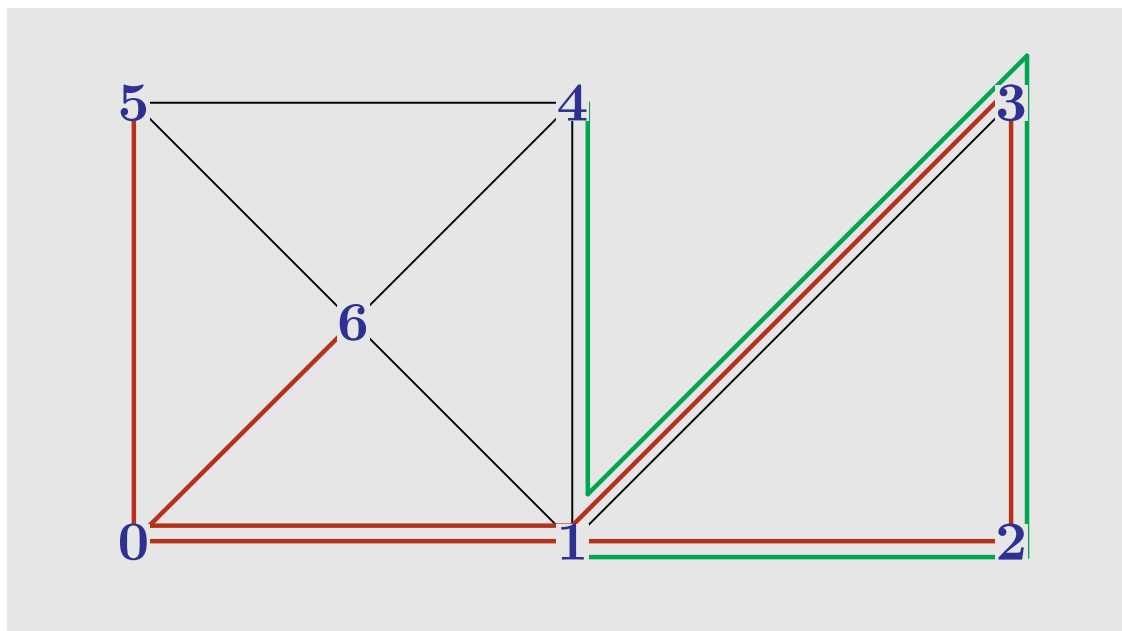


droga — (czyli $(5, 0, 1, 2, 3, 1, 0, 6)$) *nie jest prosta*,
bo dwukrotnie przechodzi przez krawędź $\{0, 1\}$

droga — (czyli $(4, 1, 3, 2, 1)$)

Drogi w grafie

Przykład:

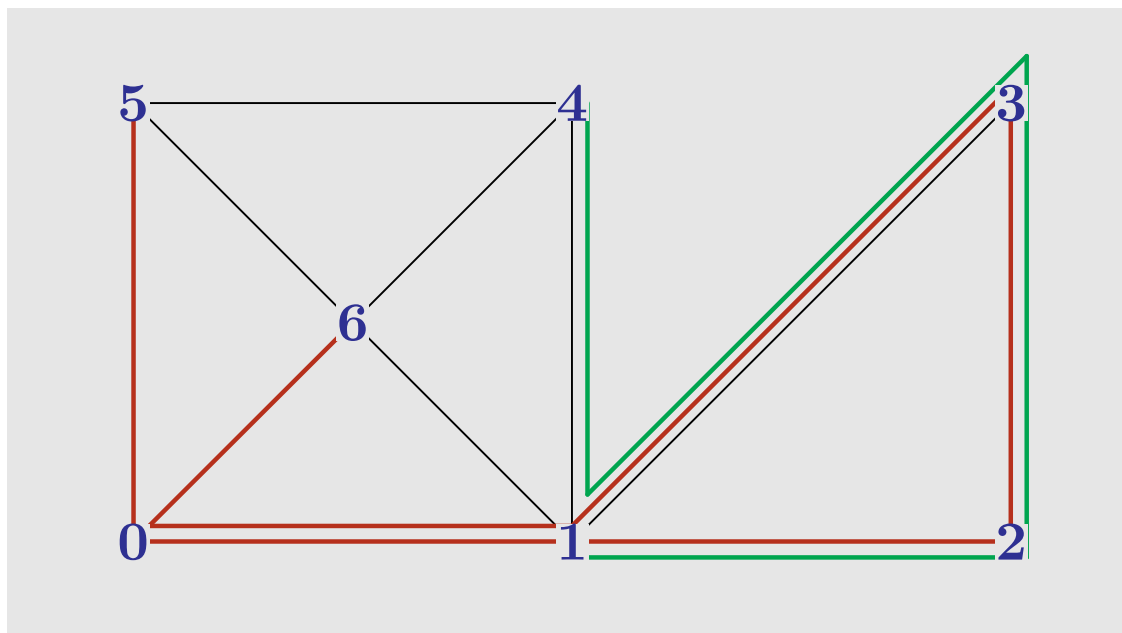


droga — (czyli $(5, 0, 1, 2, 3, 1, 0, 6)$) *nie jest prosta*,
bo dwukrotnie przechodzi przez krawędź $\{0, 1\}$

droga — (czyli $(4, 1, 3, 2, 1)$) jest prosta, ale *nie jest ścieżką*

Drogi w grafie

Przykład:



droga — (czyli $(5, 0, 1, 2, 3, 1, 0, 6)$) *nie jest prosta*,
bo dwukrotnie przechodzi przez krawędź $\{0, 1\}$

droga — (czyli $(4, 1, 3, 2, 1)$) *jest prosta*, ale *nie jest ścieżką*,
bo dwukrotnie odwiedza wierzchołek 1 , który nie jest jej początkiem i
końcem jednocześnie

Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Cykl elementarny — cykl będący ścieżką.

Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Cykl elementarny — cykl będący ścieżką.

Czyli cykl

- wychodzi z wierzchołka i do niego wraca

Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Cykl elementarny — cykl będący ścieżką.

Czyli cykl

- wychodzi z wierzchołka i do niego wraca,
- nie może dwukrotnie przechodzić przez żadną krawędź

Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Cykl elementarny — cykl będący ścieżką.

Czyli cykl

- wychodzi z wierzchołka i do niego wraca,
- nie może dwukrotnie przechodzić przez żadną krawędź,
- ale przez wierzchołek może.

Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Cykl elementarny — cykl będący ścieżką.

Czyli cykl

- wychodzi z wierzchołka i do niego wraca,
- nie może dwukrotnie przechodzić przez żadną krawędź,
- ale przez wierzchołek może.

A cykl elementarny dodatkowo

- nie odwiedza dwukrotnie żadnego wierzchołka innego niż początek i koniec.

Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Cykl elementarny — cykl będący ścieżką.

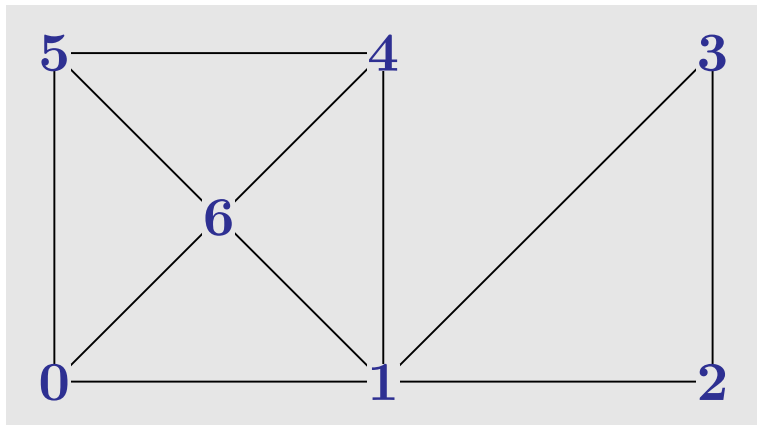
Czyli cykl

- wychodzi z wierzchołka i do niego wraca,
- nie może dwukrotnie przechodzić przez żadną krawędź,
- ale przez wierzchołek może.

A cykl elementarny dodatkowo

- nie odwiedza dwukrotnie żadnego wierzchołka innego niż początek i koniec.

Przykład:



Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Cykl elementarny — cykl będący ścieżką.

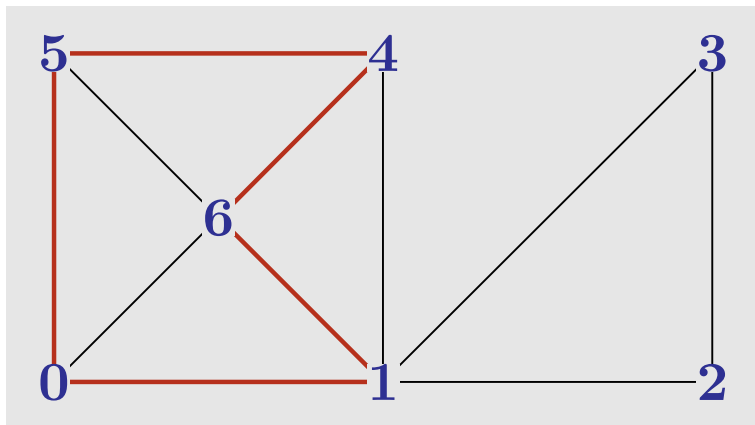
Czyli cykl

- wychodzi z wierzchołka i do niego wraca,
- nie może dwukrotnie przechodzić przez żadną krawędź,
- ale przez wierzchołek może.

A cykl elementarny dodatkowo

- nie odwiedza dwukrotnie żadnego wierzchołka innego niż początek i koniec.

Przykład:



droga — (czyli $(0, 1, 6, 4, 5, 0)$)

Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Cykl elementarny — cykl będący ścieżką.

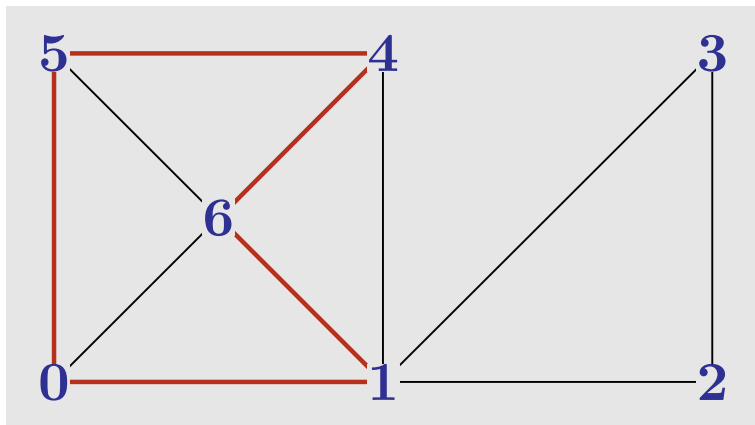
Czyli cykl

- wychodzi z wierzchołka i do niego wraca,
- nie może dwukrotnie przechodzić przez żadną krawędź,
- ale przez wierzchołek może.

A cykl elementarny dodatkowo

- nie odwiedza dwukrotnie żadnego wierzchołka innego niż początek i koniec.

Przykład:



droga — (czyli $(0, 1, 6, 4, 5, 0)$) jest cyklem elementarnym

Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Cykl elementarny — cykl będący ścieżką.

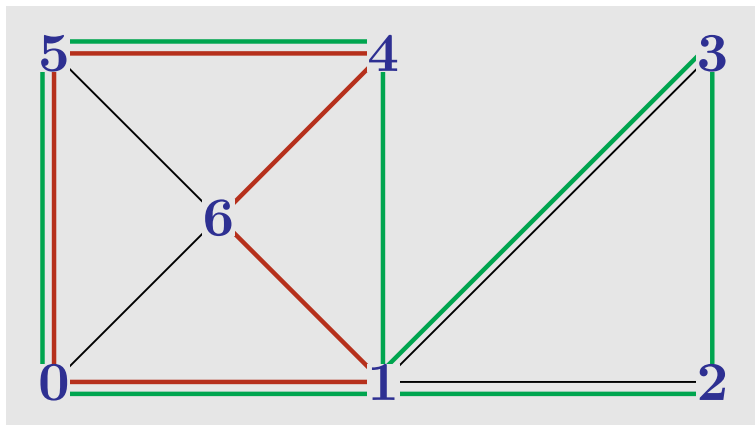
Czyli cykl

- wychodzi z wierzchołka i do niego wraca,
- nie może dwukrotnie przechodzić przez żadną krawędź,
- ale przez wierzchołek może.

A cykl elementarny dodatkowo

- nie odwiedza dwukrotnie żadnego wierzchołka innego niż początek i koniec.

Przykład:



droga — (czyli $(0, 1, 6, 4, 5, 0)$) jest cyklem elementarnym

droga — (czyli $(0, 1, 2, 3, 1, 4, 5, 0)$)

Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Cykl elementarny — cykl będący ścieżką.

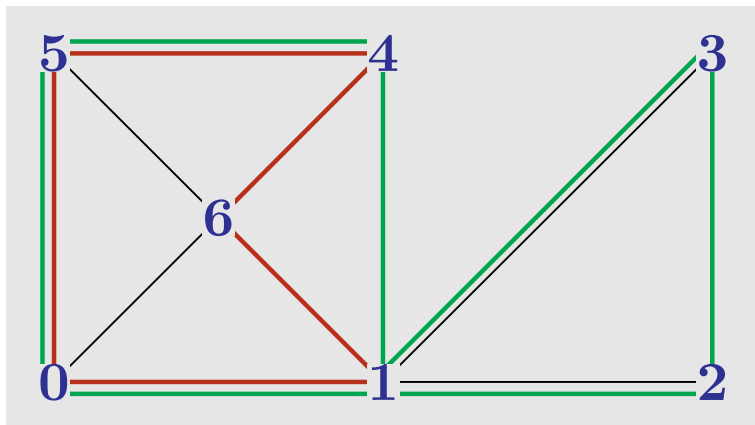
Czyli cykl

- wychodzi z wierzchołka i do niego wraca,
- nie może dwukrotnie przechodzić przez żadną krawędź,
- ale przez wierzchołek może.

A cykl elementarny dodatkowo

- nie odwiedza dwukrotnie żadnego wierzchołka innego niż początek i koniec.

Przykład:



droga — (czyli $(0, 1, 6, 4, 5, 0)$) jest cyklem elementarnym

droga — (czyli $(0, 1, 2, 3, 1, 4, 5, 0)$) jest cyklem, ale *nie elementarnym*

Drogi w grafie

Definicja:

Cykl — droga prosta zamknięta długości ≥ 2 .

Cykl elementarny — cykl będący ścieżką.

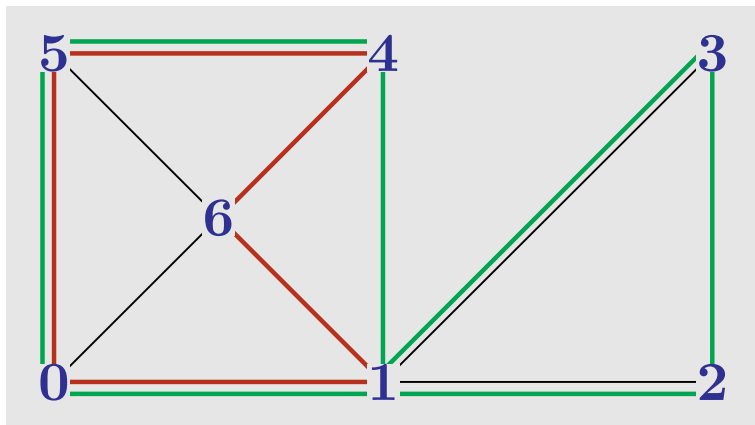
Czyli cykl

- wychodzi z wierzchołka i do niego wraca,
- nie może dwukrotnie przechodzić przez żadną krawędź,
- ale przez wierzchołek może.

A cykl elementarny dodatkowo

- nie odwiedza dwukrotnie żadnego wierzchołka innego niż początek i koniec.

Przykład:



droga — (czyli $(0, 1, 6, 4, 5, 0)$) jest cyklem elementarnym

droga — (czyli $(0, 1, 2, 3, 1, 4, 5, 0)$) jest cyklem, ale *nie elementarnym*, bo dwukrotnie odwiedza wierzchołek 1, który nie jest początkiem i końcem jednocześnie

Drogi w grafie

Definicja:

Graf jest **spójny** $\iff$ każde jego dwa wierzchołki łączy jakaś droga.

Drogi w grafie

Definicja:

Graf jest **spójny** $\iff$ każde jego dwa wierzchołki łączy jakaś droga.

Definicja:

Drzewo — graf spójny i acykliczny (czyli nie zawierający cyklu).

Drogi w grafie

Definicja:

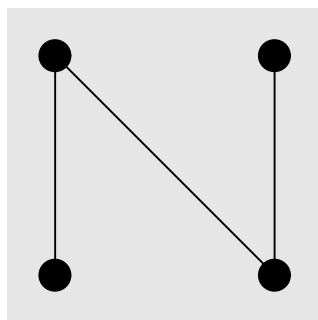
Graf jest **spójny** $\iff$ każde jego dwa wierzchołki łączy jakaś droga.

Definicja:

Drzewo — graf spójny i acykliczny (czyli nie zawierający cyklu).

Przykład:

DRZEWA



Drogi w grafie

Definicja:

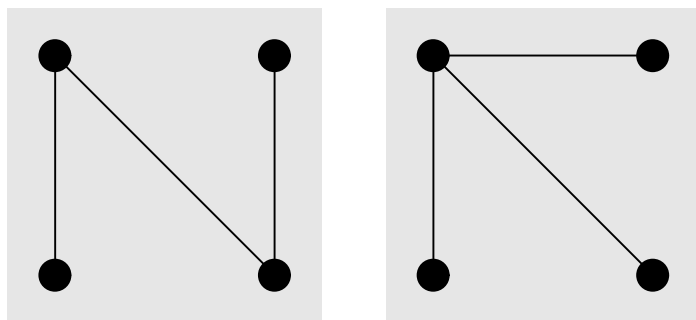
Graf jest **spójny** $\iff$ każde jego dwa wierzchołki łączy jakaś droga.

Definicja:

Drzewo — graf spójny i acykliczny (czyli nie zawierający cyklu).

Przykład:

DRZEWA



Drogi w grafie

Definicja:

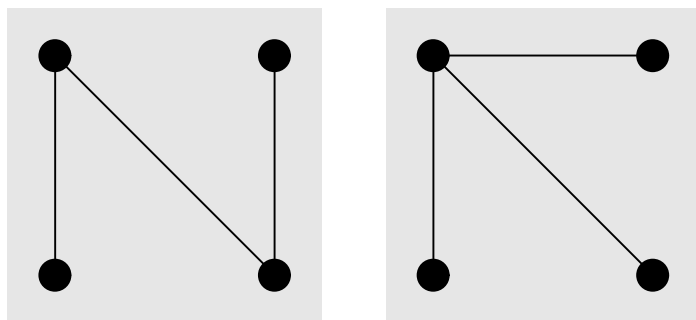
Graf jest **spójny** $\iff$ każde jego dwa wierzchołki łączy jakaś droga.

Definicja:

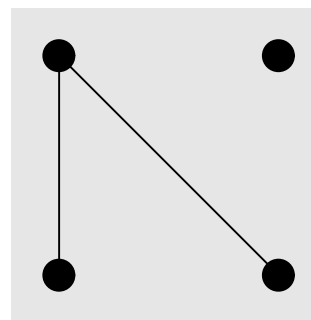
Drzewo — graf spójny i acykliczny (czyli nie zawierający cyklu).

Przykład:

DRZEWA



NIE DRZEWA



Drogi w grafie

Definicja:

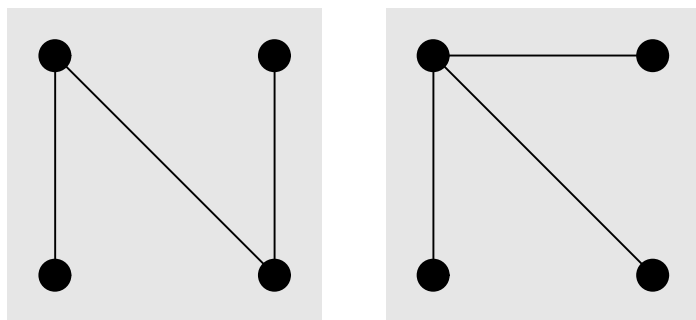
Graf jest **spójny** $\iff$ każde jego dwa wierzchołki łączy jakaś droga.

Definicja:

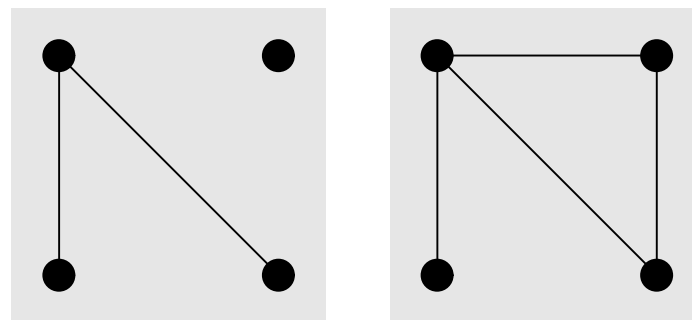
Drzewo — graf spójny i acykliczny (czyli nie zawierający cyklu).

Przykład:

DRZEWA



NIE DRZEWA



Drogi w grafie

Definicja:

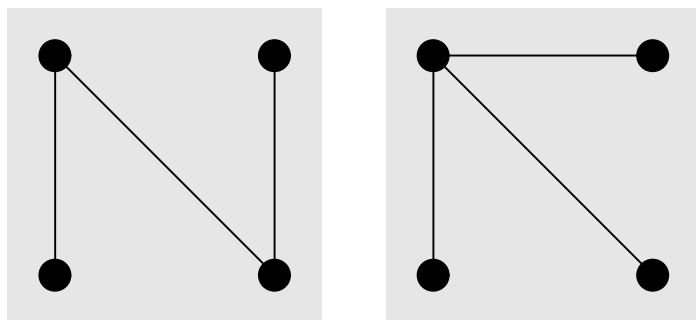
Graf jest **spójny** $\iff$ każde jego dwa wierzchołki łączy jakaś droga.

Definicja:

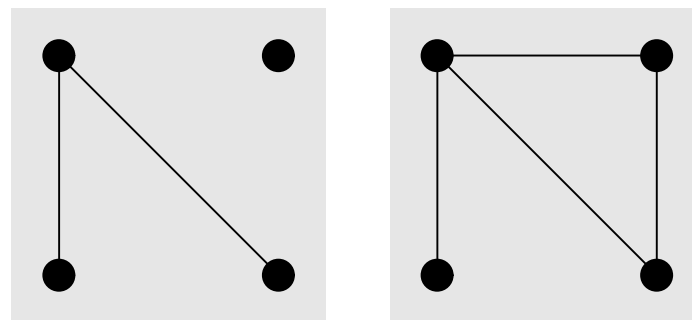
Drzewo — graf spójny i acykliczny (czyli nie zawierający cyklu).

Przykład:

DRZEWA



NIE DRZEWA

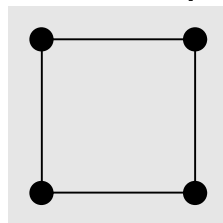


Prosty fakt:

drzewo ma krawędzi o 1 mniej niż wierzchołków

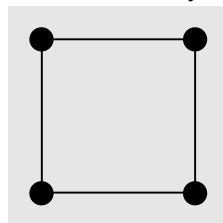
Drogi w grafie

Przykład: (spójny nie musi być drzewem)

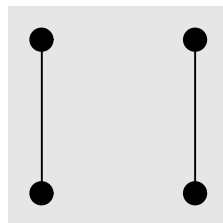


Drogi w grafie

Przykład: (spójny nie musi być drzewem)

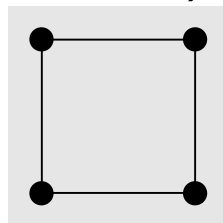


Przykład: (acykliczny nie musi być drzewem)

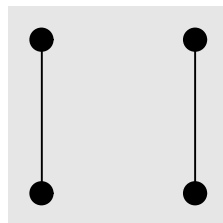


Drogi w grafie

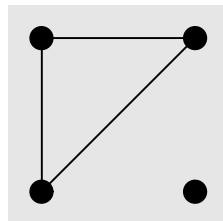
Przykład: (spójny nie musi być drzewem)



Przykład: (acykliczny nie musi być drzewem)



Przykład: (mający krawędzi o 1 mniej niż wierzchołków nie musi być drzewem)



Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;*
- (b) każde dwa wierzchołki w G łączy dokładnie jedna ścieżka*

Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;*
- (b) każde dwa wierzchołki w G łączy **dokładnie jedna** ścieżka;*
- (c) G jest spójny i ma krawędzi o **1** mniej niż wierzchołków*

Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;
- (b) każde dwa wierzchołki w G łączy *dokładnie jedna* ścieżka;
- (c) G jest spójny i ma krawędzi o 1 mniej niż wierzchołków;
- (d) G jest acykliczny i ma krawędzi o 1 mniej niż wierzchołków

Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;
- (b) każde dwa wierzchołki w G łączy **dokładnie jedna** ścieżka;
- (c) G jest spójny i ma krawędzi o **1** mniej niż wierzchołków;
- (d) G jest acykliczny i ma krawędzi o **1** mniej niż wierzchołków;
- (e) G jest acykliczny, a po dodaniu **1** krawędzi ma dokładnie jeden cykl.

Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;
- (b) każde dwa wierzchołki w G łączy **dokładnie jedna** ścieżka;
- (c) G jest spójny i ma krawędzi o **1** mniej niż wierzchołków;
- (d) G jest acykliczny i ma krawędzi o **1** mniej niż wierzchołków;
- (e) G jest acykliczny, a po dodaniu **1** krawędzi ma dokładnie jeden cykl.

Wniosek:

Jeśli G jest drzewem nietrywialnym (czyli ma co najmniej **2** wierzchołki), to ma co najmniej dwa wierzchołki stopnia **1**.

Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;
- (b) każde dwa wierzchołki w G łączy **dokładnie jedna** ścieżka;
- (c) G jest spójny i ma krawędzi o 1 mniej niż wierzchołków;
- (d) G jest acykliczny i ma krawędzi o 1 mniej niż wierzchołków;
- (e) G jest acykliczny, a po dodaniu 1 krawędzi ma dokładnie jeden cykl.

Wniosek:

Jeśli G jest drzewem nietrywialnym (czyli ma co najmniej 2 wierzchołki), to ma co najmniej dwa wierzchołki stopnia 1.

Dowód: (wniosku)

$$\text{Lemat o uściskach dłoni} \Rightarrow \sum_{v \in V} d_G(v) = 2 \cdot |E|$$

Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;
- (b) każde dwa wierzchołki w G łączy **dokładnie jedna** ścieżka;
- (c) G jest spójny i ma krawędzi o 1 mniej niż wierzchołków;
- (d) G jest acykliczny i ma krawędzi o 1 mniej niż wierzchołków;
- (e) G jest acykliczny, a po dodaniu 1 krawędzi ma dokładnie jeden cykl.

Wniosek:

Jeśli G jest drzewem nietrywialnym (czyli ma co najmniej 2 wierzchołki), to ma co najmniej dwa wierzchołki stopnia 1.

Dowód: (wniosku)

Lemat o uściskach dłoni $\Rightarrow \sum_{v \in V} d_G(v) = 2 \cdot |E| = 2 \cdot (|V| - 1)$

w drzewie:
krawędzi o 1 mniej
niż wierzchołków

Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;
- (b) każde dwa wierzchołki w G łączy **dokładnie jedna** ścieżka;
- (c) G jest spójny i ma krawędzi o 1 mniej niż wierzchołków;
- (d) G jest acykliczny i ma krawędzi o 1 mniej niż wierzchołków;
- (e) G jest acykliczny, a po dodaniu 1 krawędzi ma dokładnie jeden cykl.

Wniosek:

Jeśli G jest drzewem nietrywialnym (czyli ma co najmniej 2 wierzchołki), to ma co najmniej dwa wierzchołki stopnia 1.

Dowód: (wniosku)

Lemat o uściskach dłoni $\Rightarrow \sum_{v \in V} d_G(v) = 2 \cdot |E| = 2 \cdot (|V| - 1) = 2 \cdot |V| - 2$

w drzewie:
krawędzi o 1 mniej
niż wierzchołków

Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;
- (b) każde dwa wierzchołki w G łączy **dokładnie jedna** ścieżka;
- (c) G jest spójny i ma krawędzi o 1 mniej niż wierzchołków;
- (d) G jest acykliczny i ma krawędzi o 1 mniej niż wierzchołków;
- (e) G jest acykliczny, a po dodaniu 1 krawędzi ma dokładnie jeden cykl.

Wniosek:

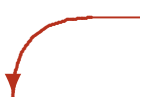
Jeśli G jest drzewem nietrywialnym (czyli ma co najmniej 2 wierzchołki), to ma co najmniej dwa wierzchołki stopnia 1.

Dowód: (wniosku)

Lemat o uściskach dłoni $\Rightarrow \sum_{v \in V} d_G(v) = 2 \cdot |E| = 2 \cdot (|V| - 1) = 2 \cdot |V| - 2$

W nietrywialnym drzewie każdy wierzchołek ma stopień ≥ 1 .

w drzewie:
krawędzi o 1 mniej
niż wierzchołków



Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;
- (b) każde dwa wierzchołki w G łączy **dokładnie jedna** ścieżka;
- (c) G jest spójny i ma krawędzi o 1 mniej niż wierzchołków;
- (d) G jest acykliczny i ma krawędzi o 1 mniej niż wierzchołków;
- (e) G jest acykliczny, a po dodaniu 1 krawędzi ma dokładnie jeden cykl.

Wniosek:

Jeśli G jest drzewem nietrywialnym (czyli ma co najmniej 2 wierzchołki), to ma co najmniej dwa wierzchołki stopnia 1.

Dowód: (wniosku)

Lemat o uściskach dłoni $\Rightarrow \sum_{v \in V} d_G(v) = 2 \cdot |E| = 2 \cdot (|V| - 1) = 2 \cdot |V| - 2$

W nietrywialnym drzewie każdy wierzchołek ma stopień ≥ 1 . Gdyby $|V| - 1$ wierzchołków miało stopień ≥ 2 , to

$$2 \cdot |V| - 2 = \sum_{v \in V} d_G(v)$$

w drzewie:
krawędzi o 1 mniej
niż wierzchołków

Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;
- (b) każde dwa wierzchołki w G łączy **dokładnie jedna** ścieżka;
- (c) G jest spójny i ma krawędzi o 1 mniej niż wierzchołków;
- (d) G jest acykliczny i ma krawędzi o 1 mniej niż wierzchołków;
- (e) G jest acykliczny, a po dodaniu 1 krawędzi ma dokładnie jeden cykl.

Wniosek:

Jeśli G jest drzewem nietrywialnym (czyli ma co najmniej 2 wierzchołki), to ma co najmniej dwa wierzchołki stopnia 1.

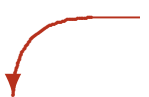
Dowód: (wniosku)

Lemat o uściskach dłoni $\Rightarrow \sum_{v \in V} d_G(v) = 2 \cdot |E| = 2 \cdot (|V| - 1) = 2 \cdot |V| - 2$

W nietrywialnym drzewie każdy wierzchołek ma stopień ≥ 1 . Gdyby $|V| - 1$ wierzchołków miało stopień ≥ 2 , to

$$2 \cdot |V| - 2 = \sum_{v \in V} d_G(v) \geq 2 \cdot (|V| - 1) + 1$$

w drzewie:
krawędzi o 1 mniej
niż wierzchołków



Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;
- (b) każde dwa wierzchołki w G łączy **dokładnie jedna** ścieżka;
- (c) G jest spójny i ma krawędzi o 1 mniej niż wierzchołków;
- (d) G jest acykliczny i ma krawędzi o 1 mniej niż wierzchołków;
- (e) G jest acykliczny, a po dodaniu 1 krawędzi ma dokładnie jeden cykl.

Wniosek:

Jeśli G jest drzewem nietrywialnym (czyli ma co najmniej 2 wierzchołki), to ma co najmniej dwa wierzchołki stopnia 1.

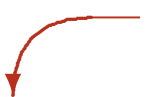
Dowód: (wniosku)

Lemat o uściskach dłoni $\Rightarrow \sum_{v \in V} d_G(v) = 2 \cdot |E| = 2 \cdot (|V| - 1) = 2 \cdot |V| - 2$

W nietrywialnym drzewie każdy wierzchołek ma stopień ≥ 1 . Gdyby $|V| - 1$ wierzchołków miało stopień ≥ 2 , to

$$2 \cdot |V| - 2 = \sum_{v \in V} d_G(v) \geq 2 \cdot (|V| - 1) + 1 = 2 \cdot |V| - 1$$

w drzewie:
krawędzi o 1 mniej
niż wierzchołków



Drogi w grafie

Twierdzenie:

Dla grafu G następujące warunki są równoważne:

- (a) G jest drzewem;
- (b) każde dwa wierzchołki w G łączy **dokładnie jedna** ścieżka;
- (c) G jest spójny i ma krawędzi o 1 mniej niż wierzchołków;
- (d) G jest acykliczny i ma krawędzi o 1 mniej niż wierzchołków;
- (e) G jest acykliczny, a po dodaniu 1 krawędzi ma dokładnie jeden cykl.

Wniosek:

Jeśli G jest drzewem nietrywialnym (czyli ma co najmniej 2 wierzchołki), to ma co najmniej dwa wierzchołki stopnia 1.

Dowód: (wniosku)

Lemat o uściskach dłoni $\Rightarrow \sum_{v \in V} d_G(v) = 2 \cdot |E| = 2 \cdot (|V| - 1) = 2 \cdot |V| - 2$

W nietrywialnym drzewie każdy wierzchołek ma stopień ≥ 1 . Gdyby $|V| - 1$ wierzchołków miało stopień ≥ 2 , to

$$2 \cdot |V| - 2 = \sum_{v \in V} d_G(v) \geq 2 \cdot (|V| - 1) + 1 = 2 \cdot |V| - 1$$

— sprzeczność.



w drzewie:
krawędzi o 1 mniej
niż wierzchołków

Grafy eulerowskie

Definicja:

Graf G jest **eulerowski** $\iff$ istnieje cykl zawierający *wszystkie* jego krawędzie.

Grafy eulerowskie

Definicja:

Graf G jest **eulerowski** $\iff$ istnieje cykl zawierający *wszystkie* jego krawędzie.

Graf G jest **semieulerowski** $\iff$ istnieje droga prosta zawierająca *wszystkie* jego krawędzie.

Grafy eulerowskie

Definicja:

Graf G jest **eulerowski** $\iff$ istnieje cykl zawierający *wszystkie* jego krawędzie.

Graf G jest **semieulerowski** $\iff$ istnieje droga prosta zawierająca *wszystkie* jego krawędzie.

Ta droga prosta w grafie semieulerowskim nie musi kończyć się tam, gdzie się zaczęła.

Grafy eulerowskie

Definicja:

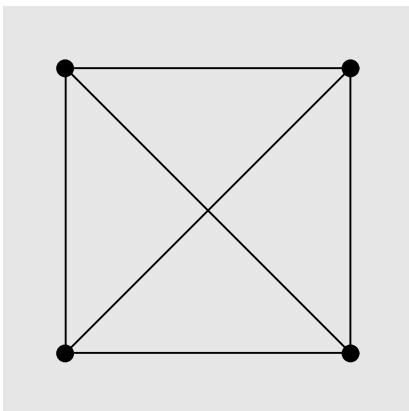
Graf G jest **eulerowski** $\iff$ istnieje cykl zawierający *wszystkie* jego krawędzie.

Graf G jest **semieulerowski** $\iff$ istnieje droga prosta zawierająca *wszystkie* jego krawędzie.

Ta droga prosta w grafie semieulerowskim nie musi kończyć się tam, gdzie się zaczęła.

Przykład:

niesemieulerowski:



Grafy eulerowskie

Definicja:

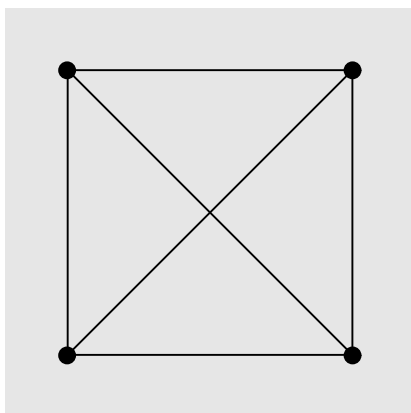
Graf G jest **eulerowski** $\iff$ istnieje cykl zawierający *wszystkie* jego krawędzie.

Graf G jest **semieulerowski** $\iff$ istnieje droga prosta zawierająca *wszystkie* jego krawędzie.

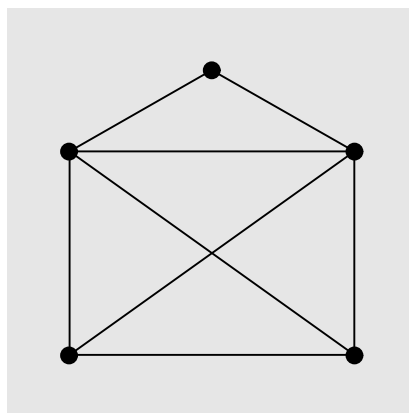
Ta droga prosta w grafie semieulerowskim nie musi kończyć się tam, gdzie się zaczęła.

Przykład:

niesemieulerowski:



semieulerowski
nieeulerowski:



Grafy eulerowskie

Definicja:

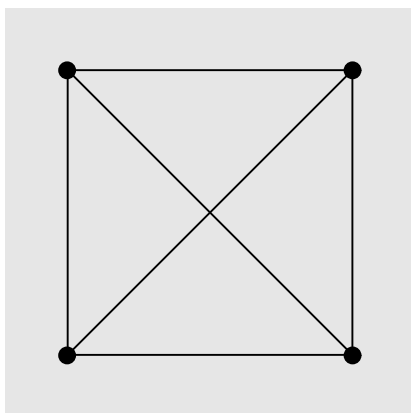
Graf G jest **eulerowski** $\iff$ istnieje cykl zawierający *wszystkie* jego krawędzie.

Graf G jest **semieulerowski** $\iff$ istnieje droga prosta zawierająca *wszystkie* jego krawędzie.

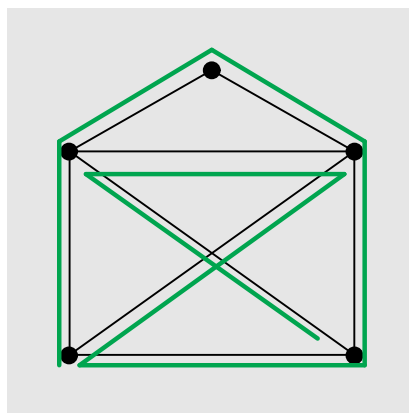
Ta droga prosta w grafie semieulerowskim nie musi kończyć się tam, gdzie się zaczęła.

Przykład:

niesemieulerowski:



semieulerowski
nieeulerowski:



Grafy eulerowskie

Definicja:

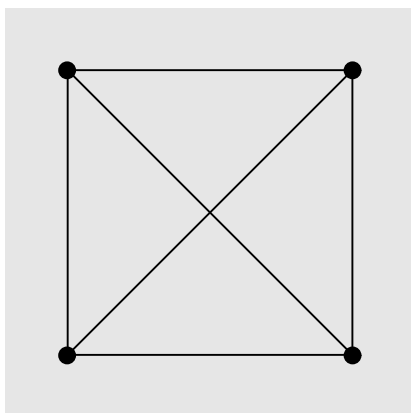
Graf G jest **eulerowski** $\iff$ istnieje cykl zawierający *wszystkie* jego krawędzie.

Graf G jest **semieulerowski** $\iff$ istnieje droga prosta zawierająca *wszystkie* jego krawędzie.

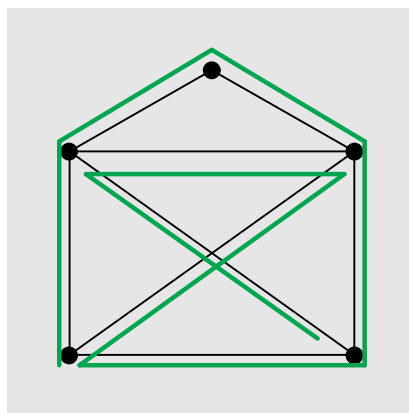
Ta droga prosta w grafie semieulerowskim nie musi kończyć się tam, gdzie się zaczęła.

Przykład:

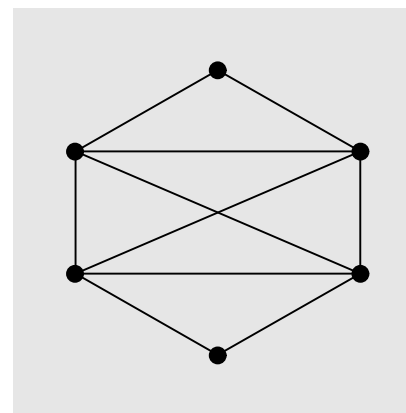
niesemieulerowski:



semieulerowski
nieeulerowski:



eulerowski:



Grafy eulerowskie

Definicja:

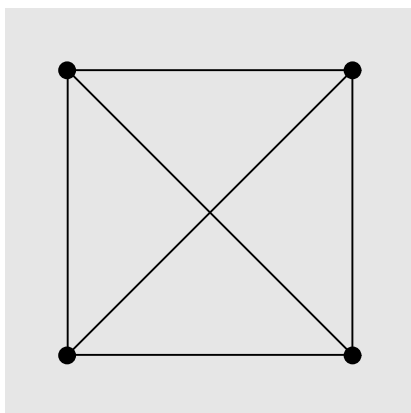
Graf G jest **eulerowski** $\iff$ istnieje cykl zawierający *wszystkie* jego krawędzie.

Graf G jest **semieulerowski** $\iff$ istnieje droga prosta zawierająca *wszystkie* jego krawędzie.

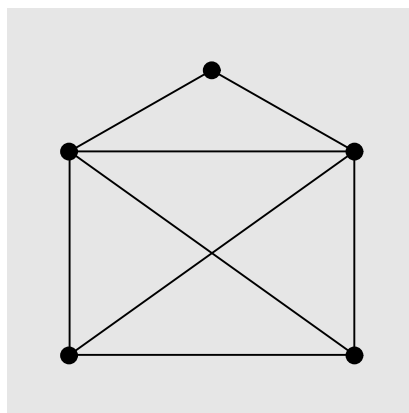
Ta droga prosta w grafie semieulerowskim nie musi kończyć się tam, gdzie się zaczęła.

Przykład:

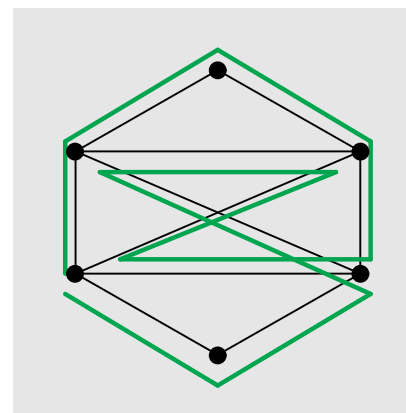
niesemieulerowski:



semieulerowski
nieeulerowski:

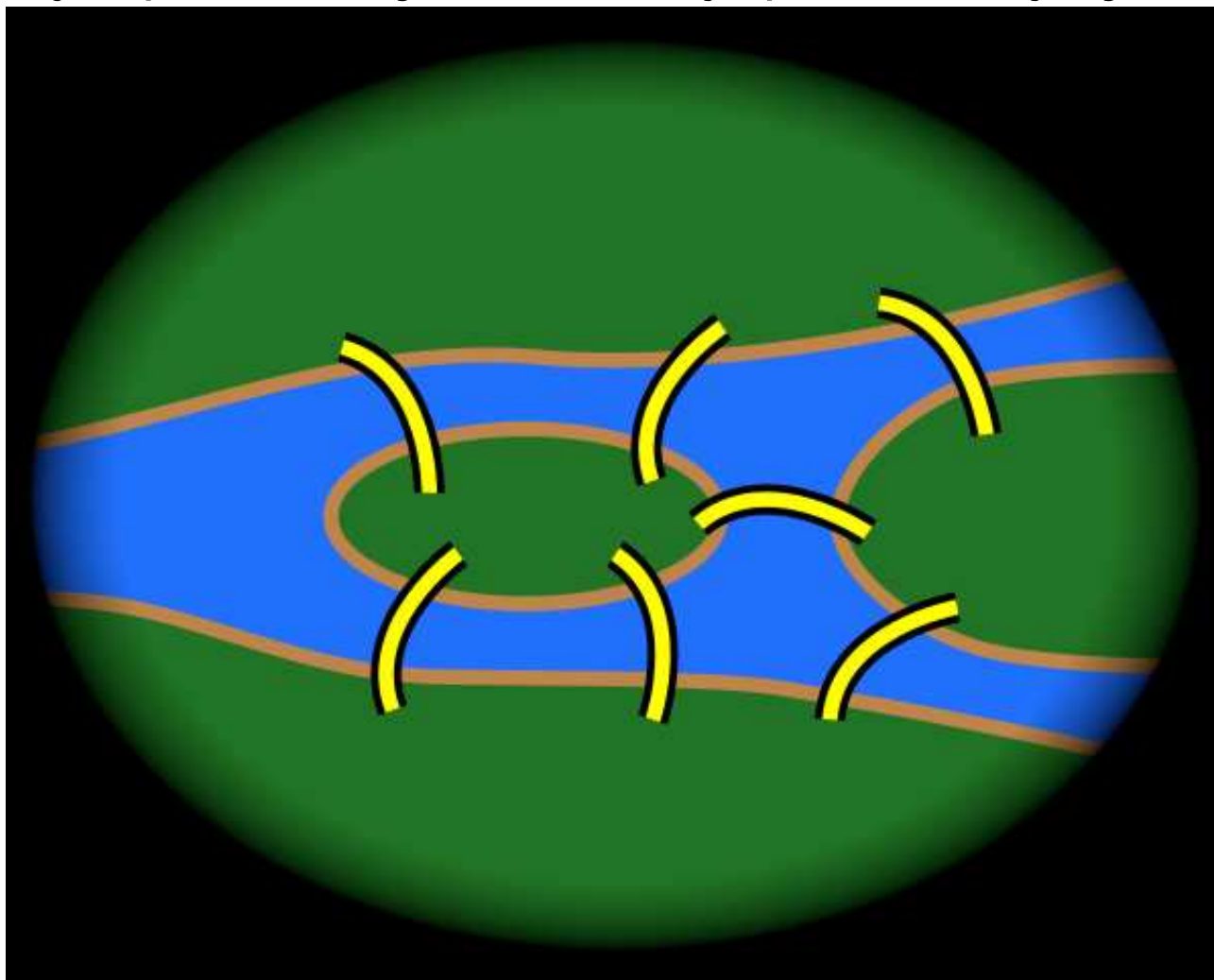


eulerowski:



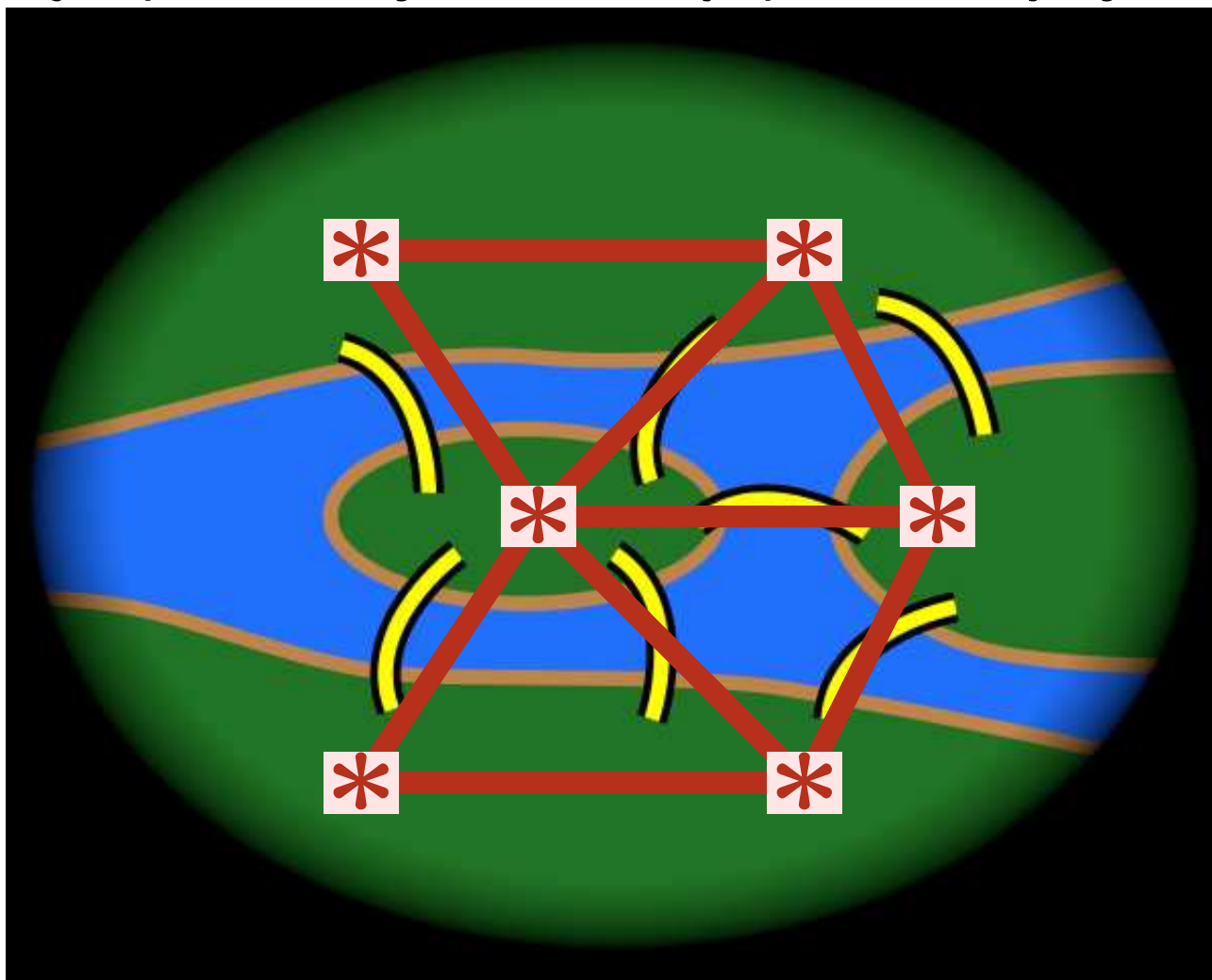
Mosty królewieckie (Euler)

Czy da się przejść przez **wszystkie** mosty, przez każdy **tylko raz**?



Mosty królewieckie (Euler)

Czy da się przejść przez **wszystkie** mosty, przez każdy **tylko raz**?



Inaczej:

Czy nałożony na nie czerwony graf jest **semieulerowski**?

Mosty królewieckie (Euler)

Twierdzenie: (Euler)

Spójny graf jest semieulerowski $\iff$ najwyżej dwa jego wierzchołki mają nieparzysty stopień.

Mosty królewieckie (Euler)

Twierdzenie: (Euler)

Spójny graf jest semieulerowski $\iff$ najwyżej dwa jego wierzchołki mają nieparzysty stopień.

Spójny graf jest eulerowski $\iff$ wszystkie jego wierzchołki mają parzysty stopień.

Mosty królewieckie (Euler)

Twierdzenie: (Euler)

Spójny graf jest semieulerowski $\iff$ najwyżej dwa jego wierzchołki mają nieparzysty stopień.

Spójny graf jest eulerowski $\iff$ wszystkie jego wierzchołki mają parzysty stopień.

Wniosek:

Zagadnienie mostów królewieckich nie ma rozwiązania.

Mosty królewieckie (Euler)

Twierdzenie: (Euler)

Spójny graf jest semieulerowski $\iff$ najwyżej dwa jego wierzchołki mają nieparzysty stopień.

Spójny graf jest eulerowski $\iff$ wszystkie jego wierzchołki mają parzysty stopień.

Wniosek:

Zagadnienie mostów królewieckich nie ma rozwiązania.

Dowód: (wniosku)

Mosty królewskie (Euler)

Twierdzenie: (Euler)

Spójny graf jest semieulerowski $\iff$ najwyżej dwa jego wierzchołki mają nieparzysty stopień.

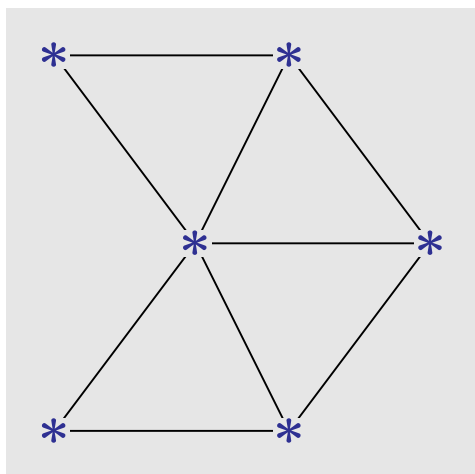
Spójny graf jest eulerowski $\iff$ wszystkie jego wierzchołki mają parzysty stopień.

Wniosek:

Zagadnienie mostów królewskich nie ma rozwiązania.

Dowód: (wniosku)

Graf



Mosty królewskie (Euler)

Twierdzenie: (Euler)

Spójny graf jest semieulerowski $\iff$ najwyżej dwa jego wierzchołki mają nieparzysty stopień.

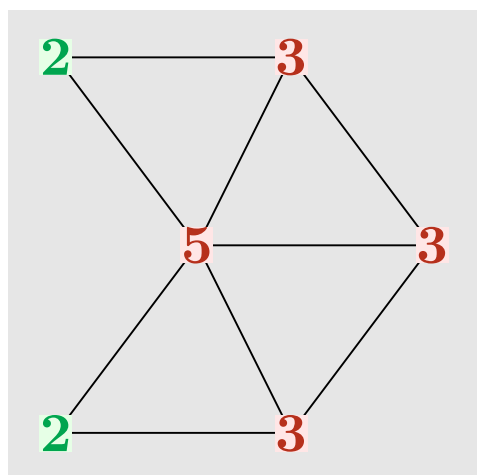
Spójny graf jest eulerowski $\iff$ wszystkie jego wierzchołki mają parzysty stopień.

Wniosek:

Zagadnienie mostów królewskich nie ma rozwiązania.

Dowód: (wniosku)

Graf



Mosty królewskie (Euler)

Twierdzenie: (Euler)

Spójny graf jest semieulerowski $\iff$ najwyżej dwa jego wierzchołki mają nieparzysty stopień.

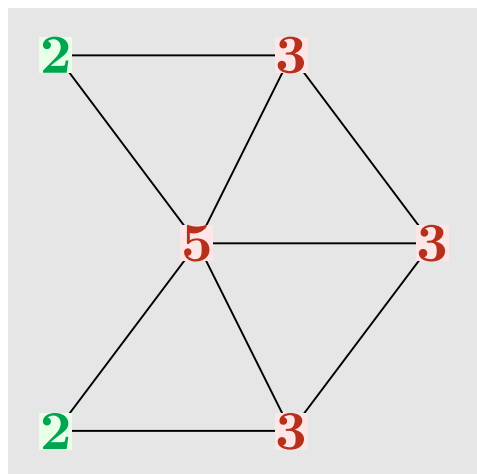
Spójny graf jest eulerowski $\iff$ wszystkie jego wierzchołki mają parzysty stopień.

Wniosek:

Zagadnienie mostów królewskich nie ma rozwiązania.

Dowód: (wniosku)

Graf



ma aż cztery wierzchołki o nieparzystym stopniu.



Grafy ważone

Grafy ważone

Definicja:

Graf ważony — trójka $G = \langle V, E, w \rangle$

- $\langle V, E \rangle$ — graf

Grafy ważone

Definicja:

Graf ważony — trójka $G = \langle V, E, w \rangle$

- $\langle V, E \rangle$ — graf
- $w : E \rightarrow \mathbb{R}_+$ — **wagi** krawędzi

Grafy ważone

Definicja:

Graf ważony — trójka $G = \langle V, E, w \rangle$

- $\langle V, E \rangle$ — graf
- $w : E \rightarrow \mathbb{R}_+$ — **wagi** krawędzi

Wagi mogą w praktyce oznaczać odległości, albo czas, itp.

Grafy ważone

Definicja:

Graf ważony — trójka $G = \langle V, E, w \rangle$

- $\langle V, E \rangle$ — graf
- $w : E \rightarrow \mathbb{R}_+$ — **wagi** krawędzi

Wagi mogą w praktyce oznaczać odległości, albo czas, itp.

Definicja:

Waga drogi — suma wag wszystkich krawędzi tej drogi

Grafy ważone

Definicja:

Graf ważony — trójka $G = \langle V, E, w \rangle$

- $\langle V, E \rangle$ — graf
- $w : E \rightarrow \mathbb{R}_+$ — **wagi** krawędzi

Wagi mogą w praktyce oznaczać odległości, albo czas, itp.

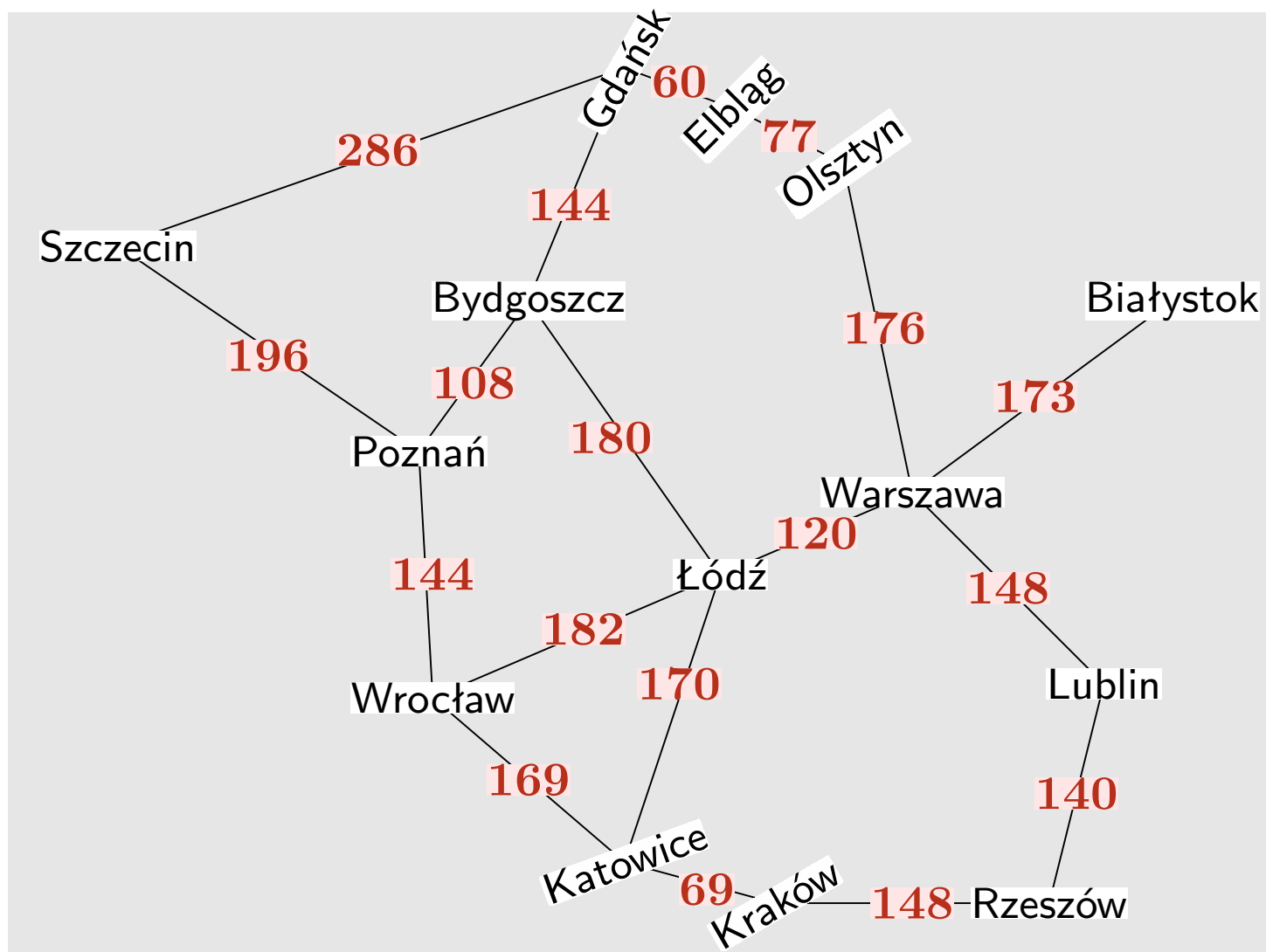
Definicja:

Waga drogi — suma wag wszystkich krawędzi tej drogi

Wiele zagadnień sprowadza się do znalezienia drogi o minimalnej wadze między dwoma wierzchołkami grafu.

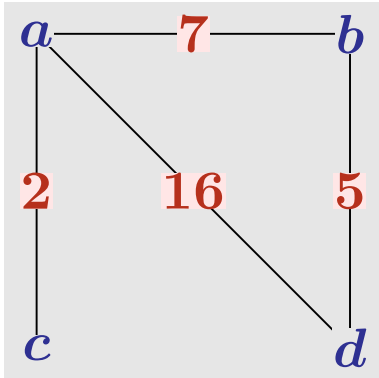
Grafy ważone

Przykład:



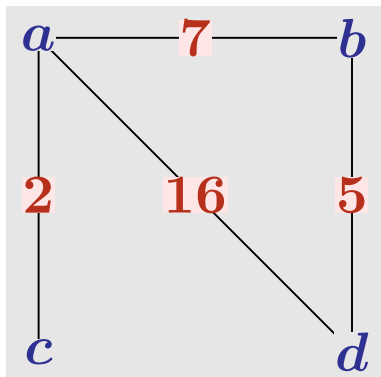
Grafy ważone

Przykład:



Grafy ważone

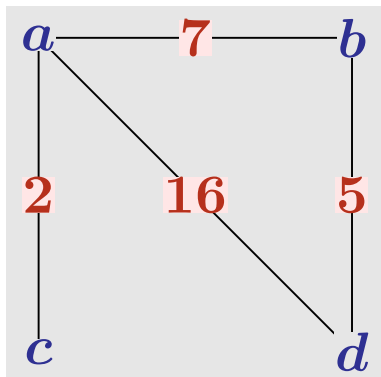
Przykład:



waga drogi (c, a, d) : 18

Grafy ważone

Przykład:

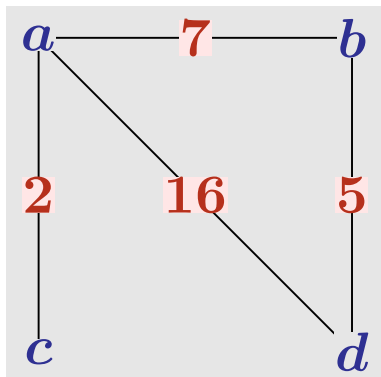


waga drogi (c, a, d) : 18

waga drogi (c, a, b, d) : 14

Grafy ważone

Przykład:



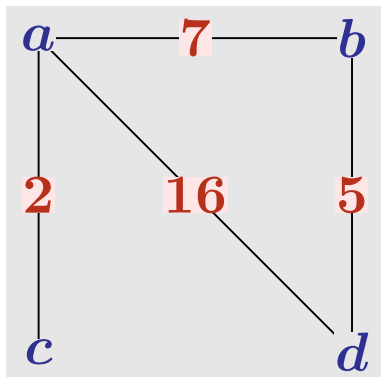
waga drogi (c, a, d) : 18

waga drogi (c, a, b, d) : 14

warto jechać przez b , a nie bezpośrednio...

Grafy ważone

Przykład:

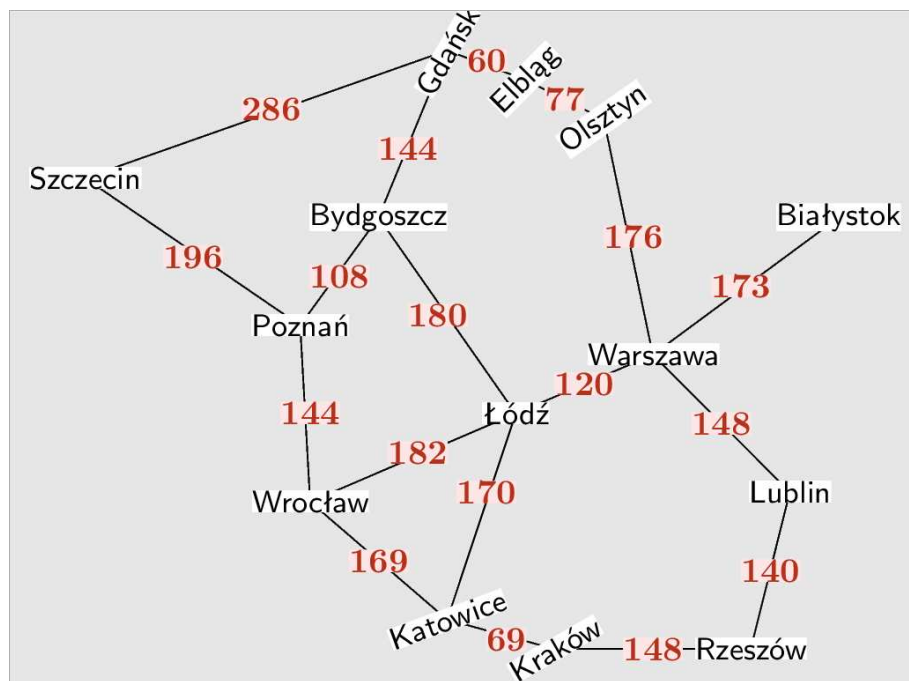


waga drogi (c, a, d) : 18

waga drogi (c, a, b, d) : 14

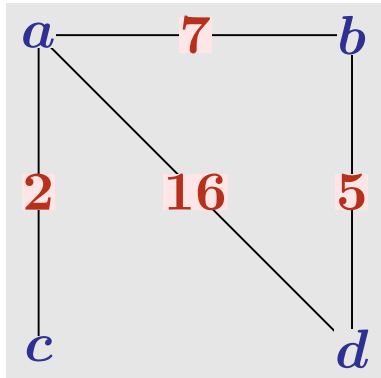
warto jechać przez b , a nie bezpośrednio...

Przykład:



Grafy ważone

Przykład:

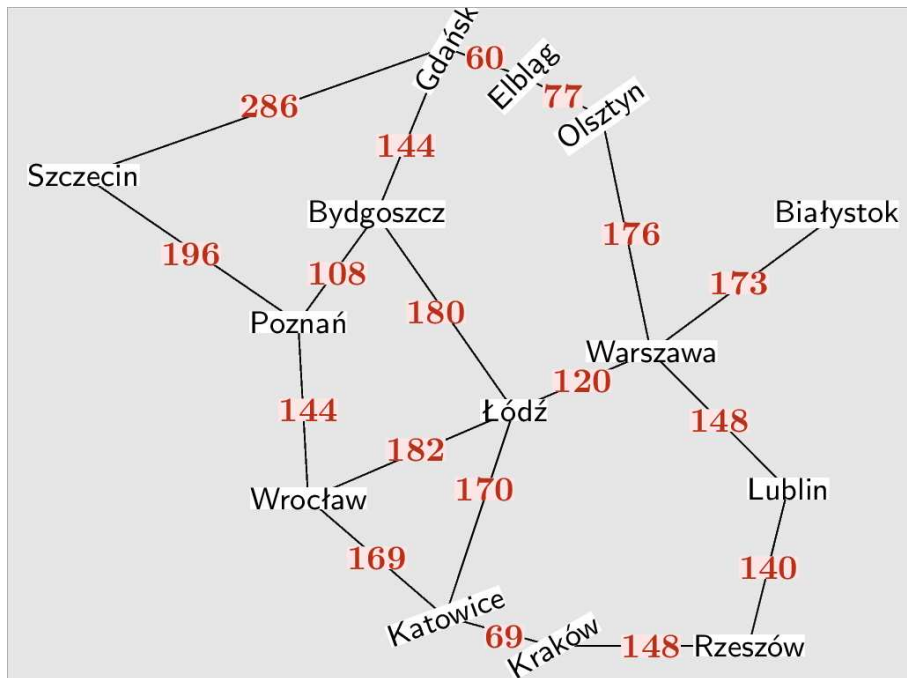


waga drogi (c, a, d) : 18

waga drogi (c, a, b, d) : 14

warto jechać przez b , a nie bezpośrednio...

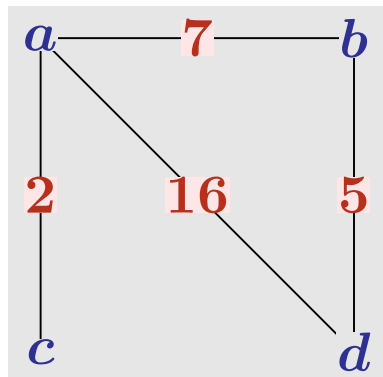
Przykład:



(Gdańsk, Elbląg, Olsztyn, Warszawa, Lublin, Rzeszów, Kraków) — 749 km

Grafy ważone

Przykład:

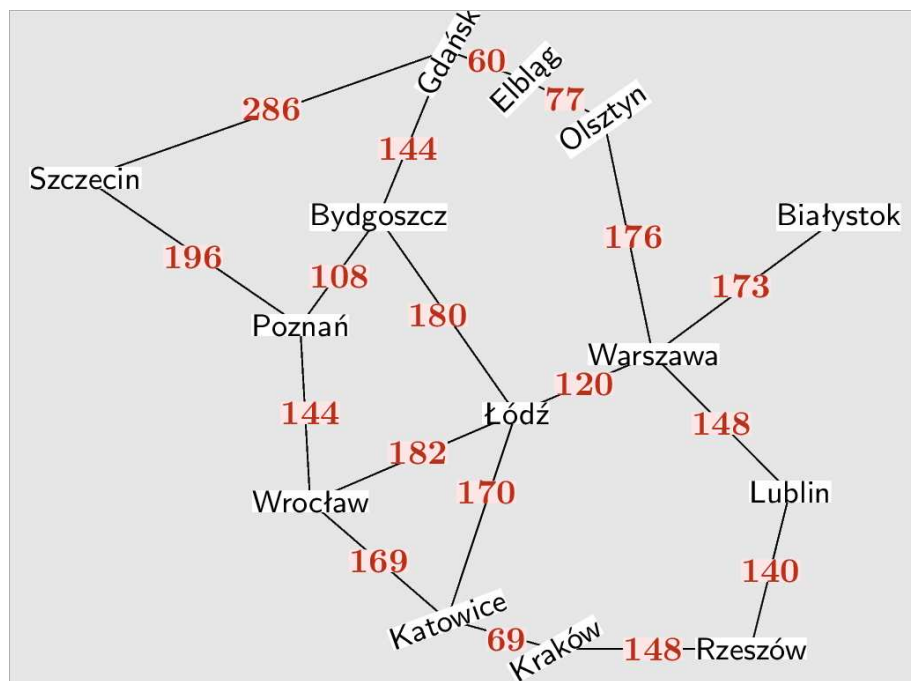


waga drogi (c, a, d) : 18

waga drogi (c, a, b, d) : 14

warto jechać przez b , a nie bezpośrednio...

Przykład:

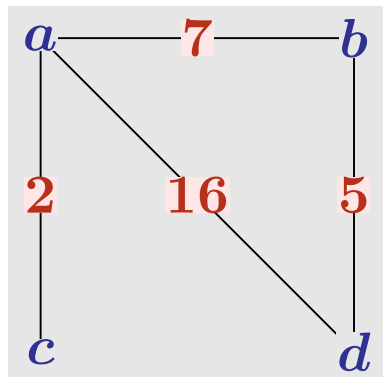


(Gdańsk, Elbląg, Olsztyn, Warszawa, Lublin, Rzeszów, Kraków) — 749 km

(Gdańsk, Elbląg, Olsztyn, Warszawa, Łódź, Katowice, Kraków) — 672 km

Grafy ważone

Przykład:

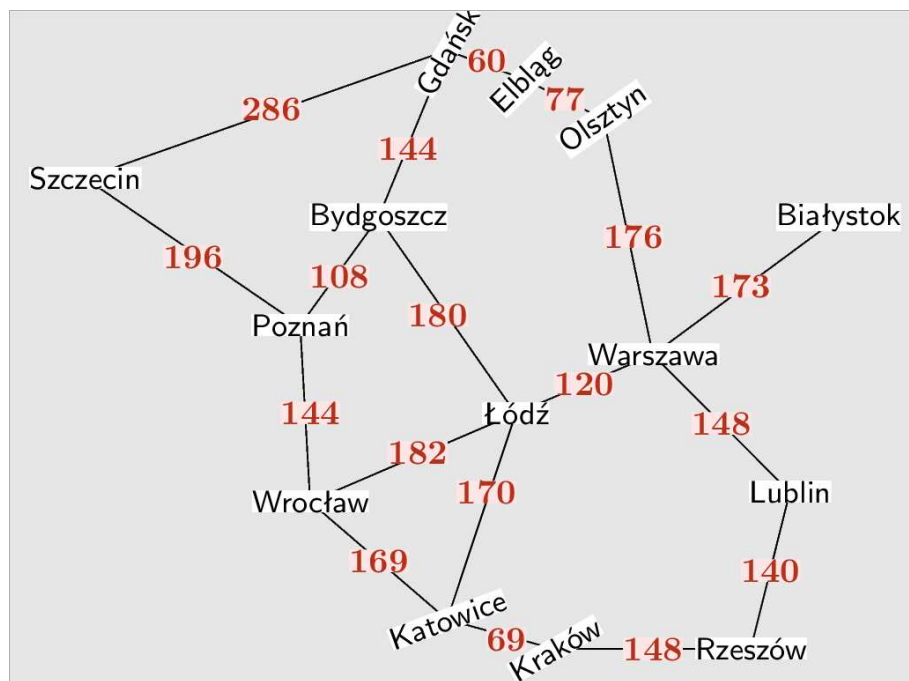


waga drogi (c, a, d) : 18

waga drogi (c, a, b, d) : 14

warto jechać przez b , a nie bezpośrednio...

Przykład:



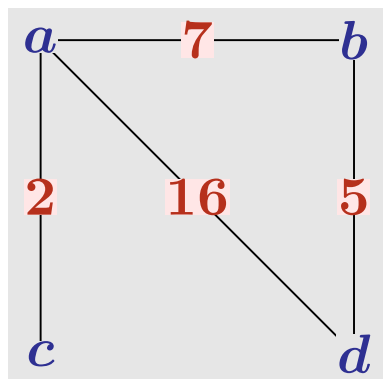
(Gdańsk, Elbląg, Olsztyn, Warszawa, Lublin, Rzeszów, Kraków) — 749 km

(Gdańsk, Elbląg, Olsztyn, Warszawa, Łódź, Katowice, Kraków) — 672 km

(Gdańsk, Bydgoszcz, Łódź, Katowice, Kraków) — 563 km

Grafy ważone

Przykład:

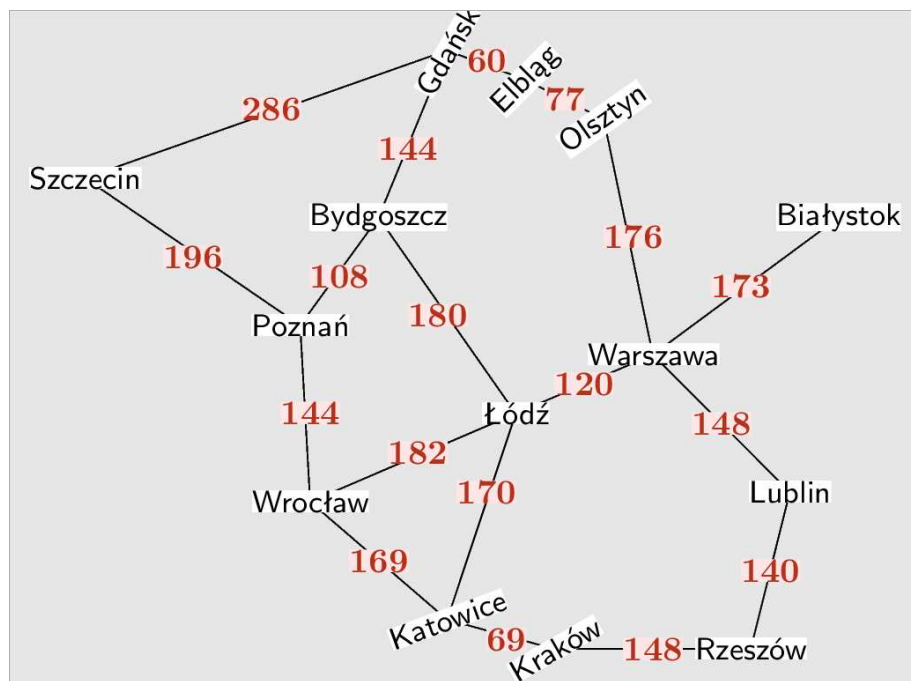


waga drogi (c, a, d) : 18

waga drogi (c, a, b, d) : 14

warto jechać przez b , a nie bezpośrednio...

Przykład:



(Gdańsk, Elbląg, Olsztyn, Warszawa, Lublin, Rzeszów, Kraków) — 749 km

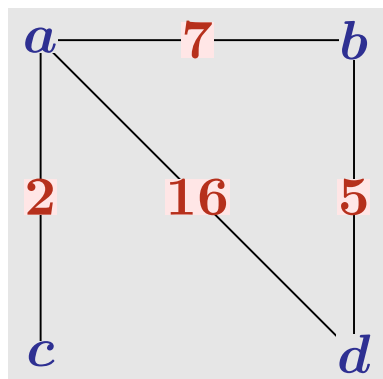
(Gdańsk, Elbląg, Olsztyn, Warszawa, Łódź, Katowice, Kraków) — 672 km

(Gdańsk, Bydgoszcz, Łódź, Katowice, Kraków) — 563 km

(Gdańsk, Bydgoszcz, Poznań, Wrocław, Katowice, Kraków) — 634 km

Grafy ważone

Przykład:

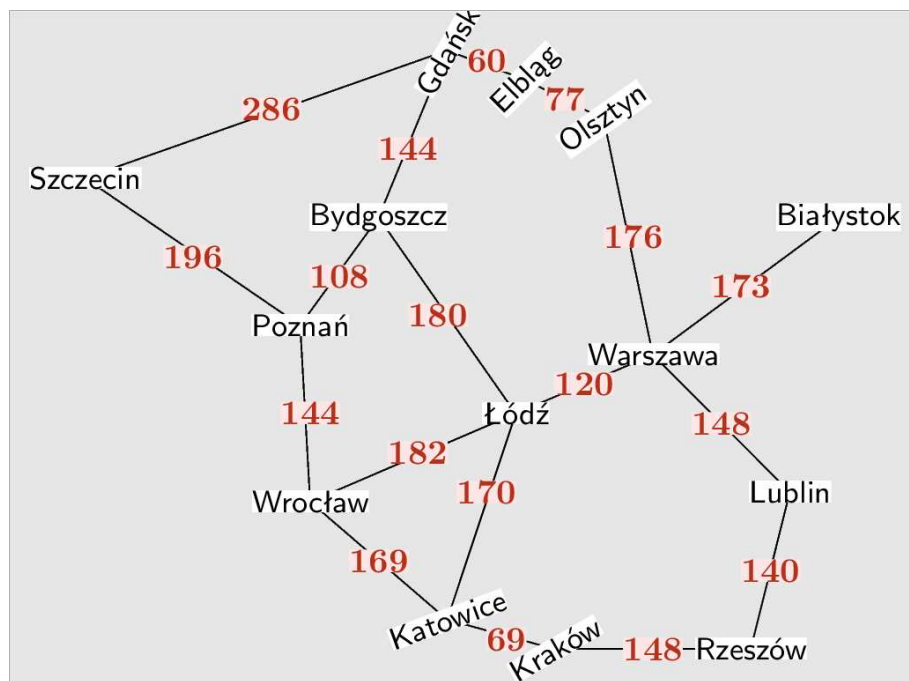


waga drogi (c, a, d) : 18

waga drogi (c, a, b, d) : 14

warto jechać przez b , a nie bezpośrednio...

Przykład:



(Gdańsk, Elbląg, Olsztyn, Warszawa, Lublin, Rzeszów, Kraków) — 749 km

(Gdańsk, Elbląg, Olsztyn, Warszawa, Łódź, Katowice, Kraków) — 672 km

(Gdańsk, Bydgoszcz, Łódź, Katowice, Kraków) — 563 km

(Gdańsk, Bydgoszcz, Poznań, Wrocław, Katowice, Kraków) — 634 km

Grafy ważone — algorytm Dijkstry

Algorytm Dijkstry

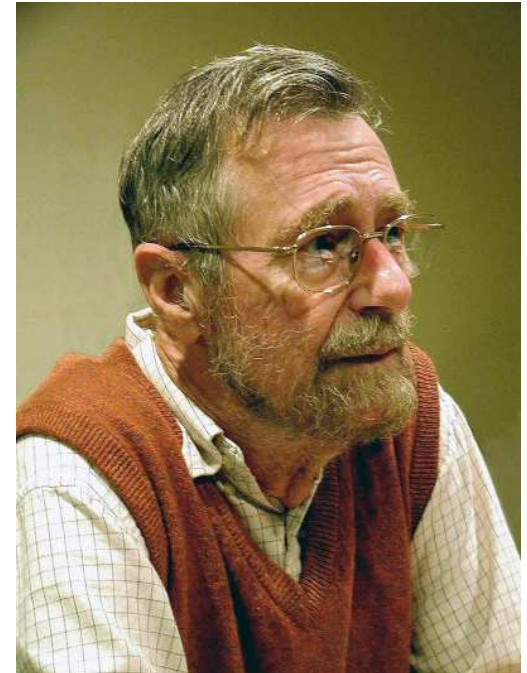


Edsger Wybe Dijkstra
1930–2002

Grafy ważone — algorytm Dijkstry

Algorytm Dijkstry

- znajduje drogi o najmniejszej wadze z wybranego wierzchołka grafu do wszystkich pozostałych wierzchołków

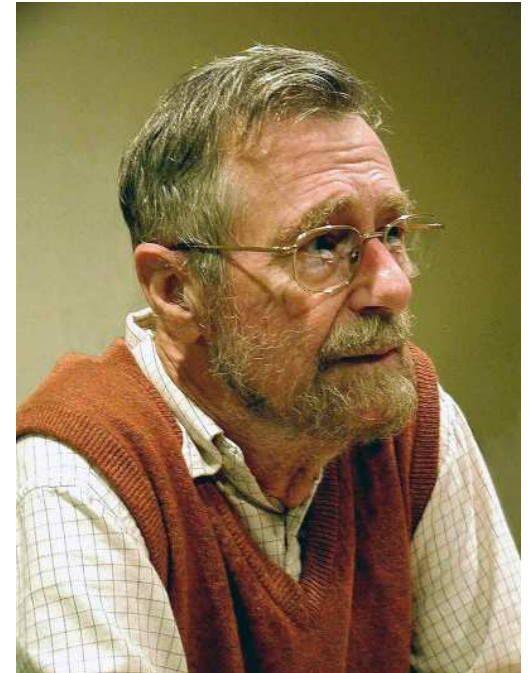


Edsger Wybe Dijkstra
1930–2002

Grafy ważone — algorytm Dijkstry

Algorytm Dijkstry

- znajduje drogi o najmniejszej wadze z wybranego wierzchołka grafu do wszystkich pozostałych wierzchołków;
- opracowany w latach 1950-tych

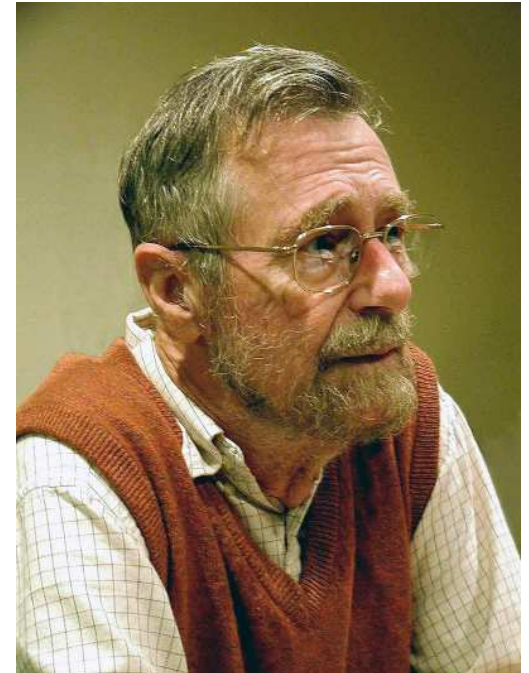


Edsger Wybe Dijkstra
1930–2002

Grafy ważone — algorytm Dijkstry

Algorytm Dijkstry

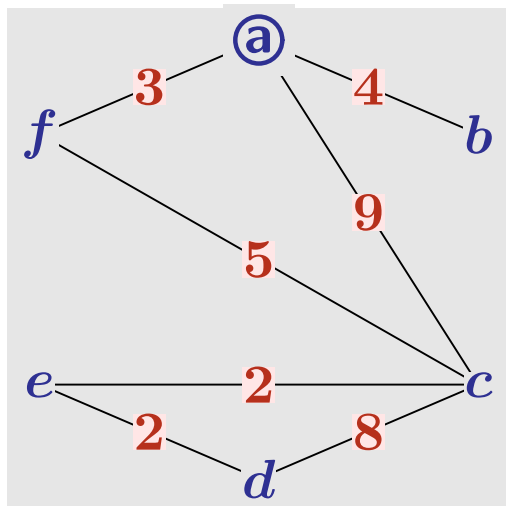
- znajduje drogi o najmniejszej wadze z wybranego wierzchołka grafu do wszystkich pozostałych wierzchołków;
- opracowany w latach 1950-tych;
- ma wiele zastosowań; m.in. w nawigacji GPS.



Edsger Wybe Dijkstra
1930–2002

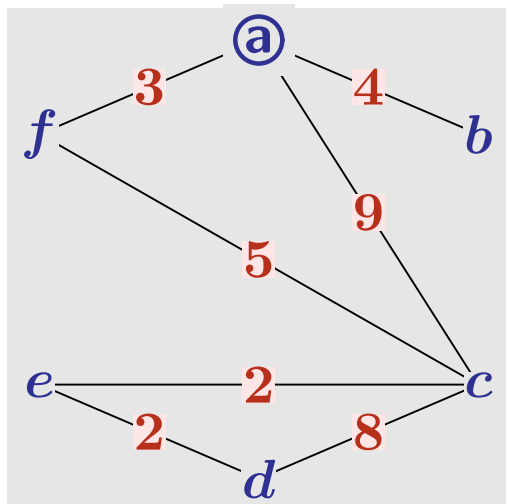
Grafy wazone — algorytm Dijkstry

graf:



Grafy ważone — algorytm Dijkstry

graf:

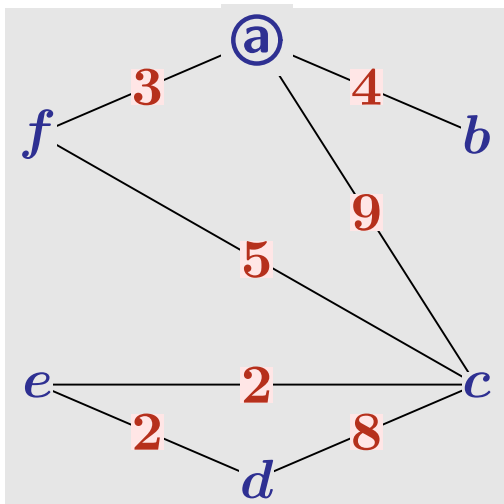


odl. bezpośrednie:

	Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Ⓐ						
<i>b</i>						
<i>c</i>						
<i>d</i>						
<i>e</i>						
<i>f</i>						

Grafy ważone — algorytm Dijkstry

graf:

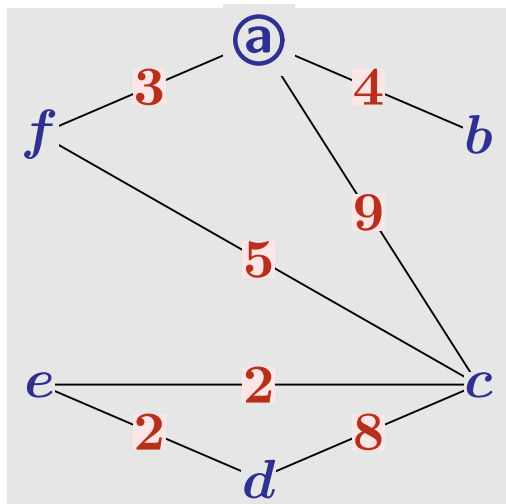


odl. bezpośrednie:

	Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Ⓐ		4	9			3
<i>b</i>						
<i>c</i>						
<i>d</i>						
<i>e</i>						
<i>f</i>						

Grafy ważone — algorytm Dijkstry

graf:

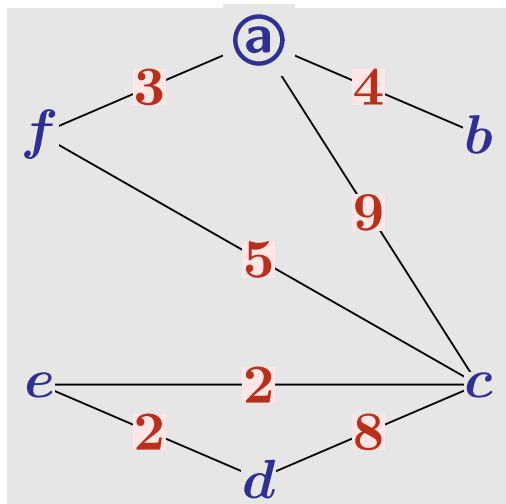


odl. bezpośrednie:

	Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>						
<i>d</i>						
<i>e</i>						
<i>f</i>						

Grafy wazone — algorytm Dijkstry

graf:

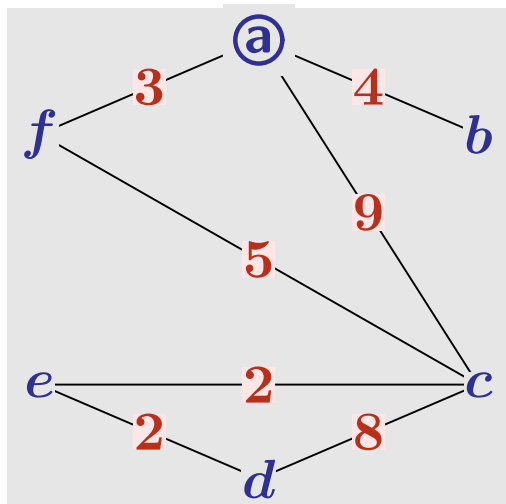


odl. bezpośrednie:

	Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>						
<i>e</i>						
<i>f</i>						

Grafy ważone — algorytm Dijkstry

graf:

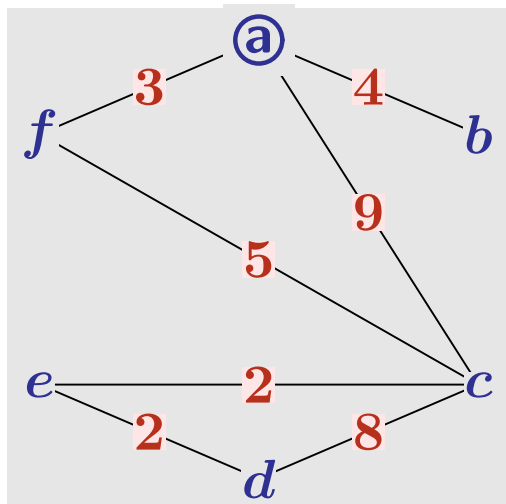


odl. bezpośrednie:

	Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>						
<i>f</i>						

Grafy ważone — algorytm Dijkstry

graf:

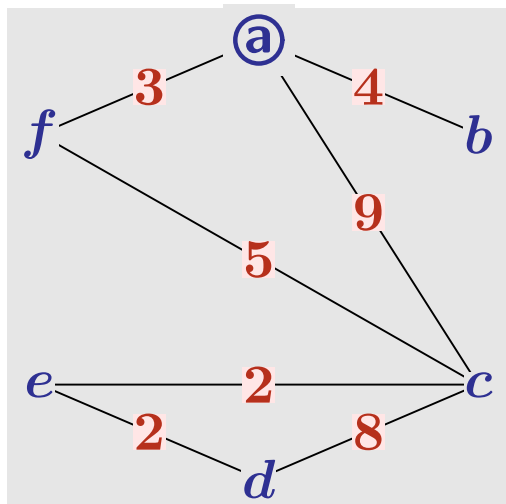


odl. bezpośrednie:

	Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>						

Grafy ważone — algorytm Dijkstry

graf:

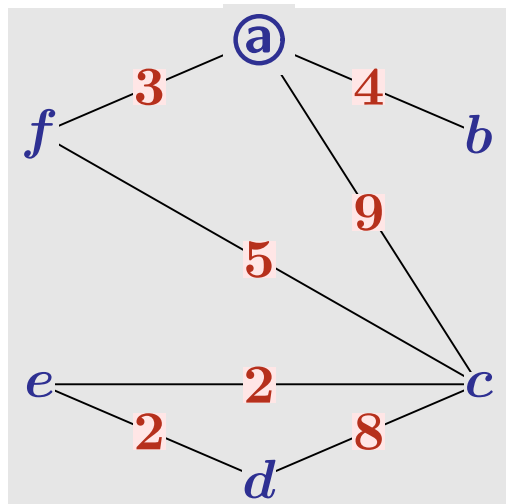


odl. bezpośrednie:

	Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od Ⓐ:

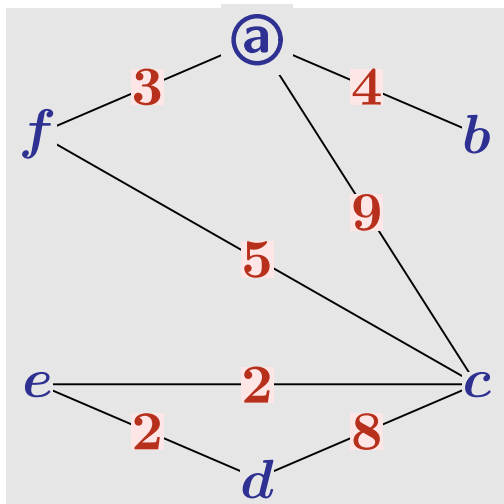
Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

wyliczone odległości od Ⓐ:

Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0					

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

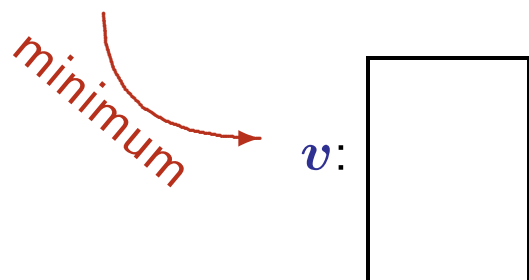
	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

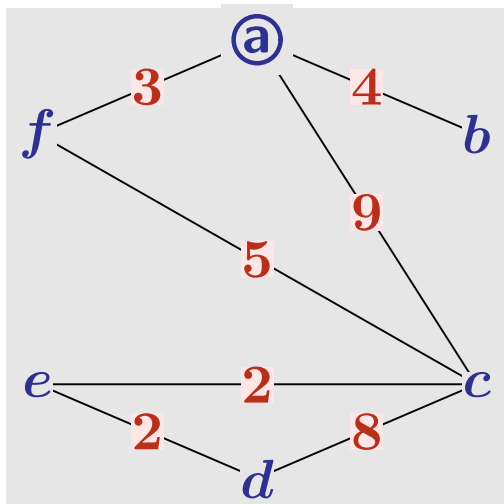
wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0					



Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0					

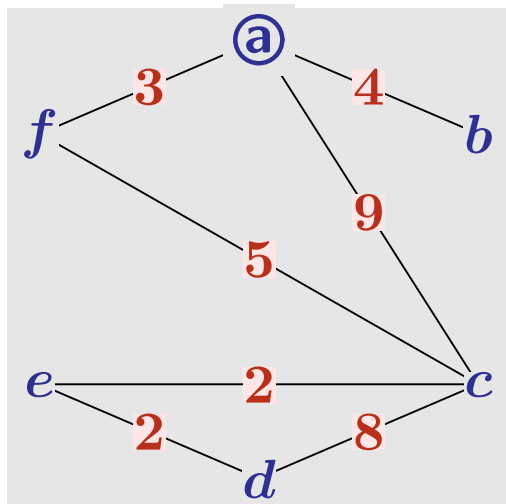
minimum

v:

a
0

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0					

minimum

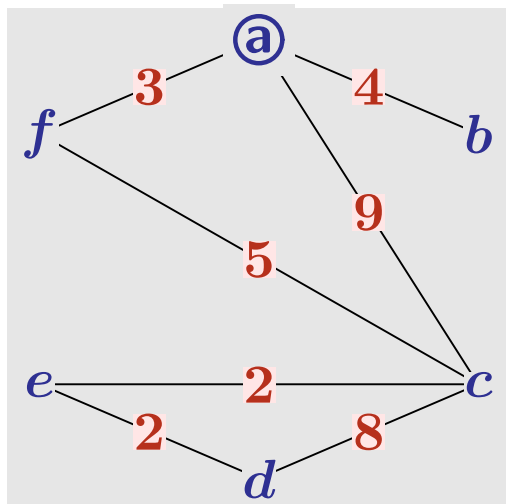
v:

a
0

a $\xrightarrow{4}$ *b* : $0 + 4 < +\infty$ więc 4 do *b*

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0					

minimum

v:

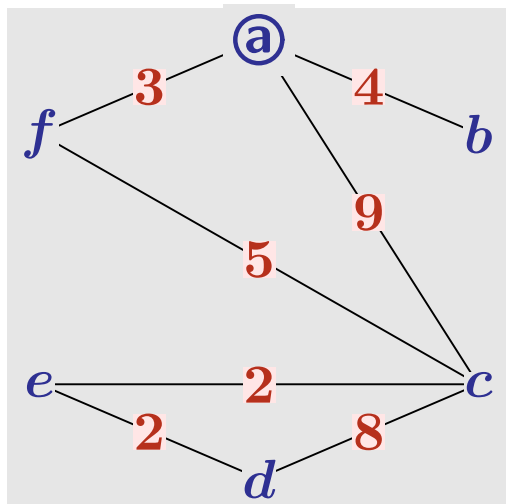
a
0

a $\xrightarrow{4}$ *b* : $0 + 4 < +\infty$ więc 4 do *b*

a $\xrightarrow{9}$ *c* : $0 + 9 < +\infty$ więc 9 do *c*

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0					

minimum

v:

a
0

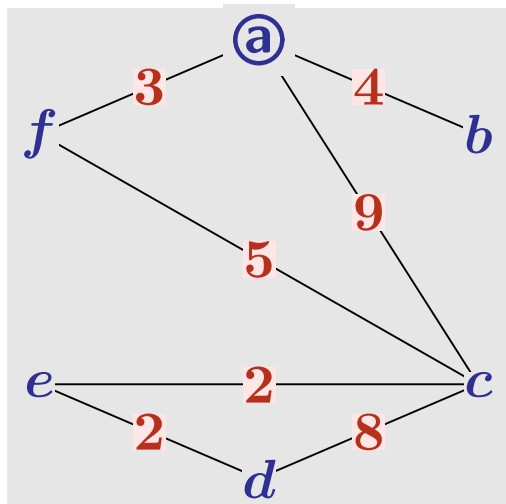
a $\xrightarrow{4}$ *b* : $0 + 4 < +\infty$ więc 4 do *b*

a $\xrightarrow{9}$ *c* : $0 + 9 < +\infty$ więc 9 do *c*

a $\xrightarrow{3}$ *f* : $0 + 3 < +\infty$ więc 3 do *f*

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	9	$+\infty$	$+\infty$	3

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	9			3

minimum

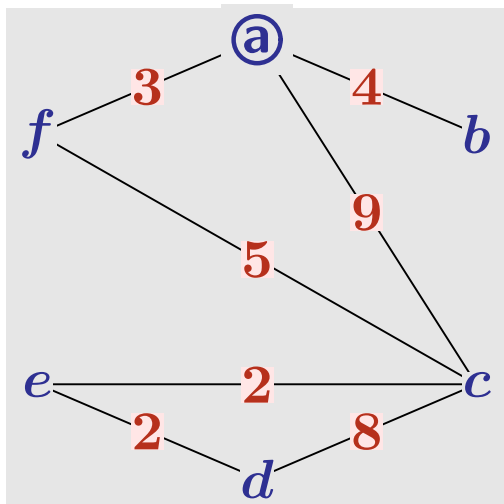
v:

a
0

- a** $\xrightarrow{4}$ *b* : $0 + 4 < +\infty$ więc 4 do *b*
a $\xrightarrow{9}$ *c* : $0 + 9 < +\infty$ więc 9 do *c*
a $\xrightarrow{3}$ *f* : $0 + 3 < +\infty$ więc 3 do *f*

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

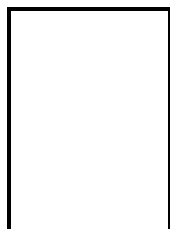
<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
4	9	$+\infty$	$+\infty$	3

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	9			3

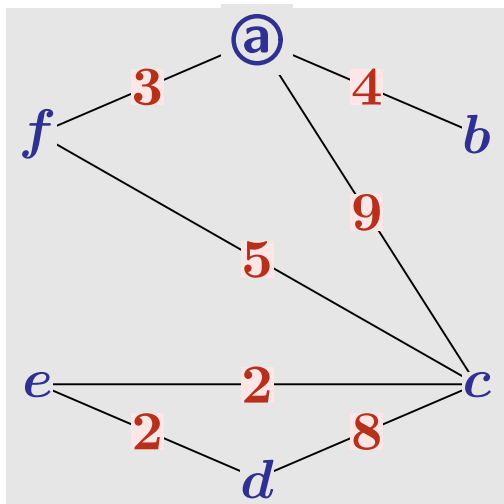
minimum

v:



Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
4	9	$+\infty$	$+\infty$	3

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	9			3

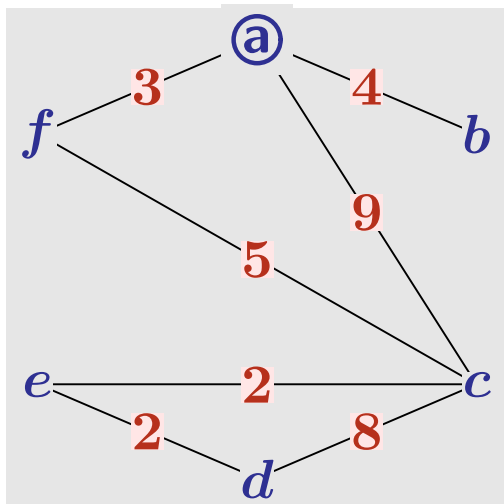
minimum

v:

$$\frac{f}{3}$$

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
4	9	$+\infty$	$+\infty$	3

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	9			3

minimum

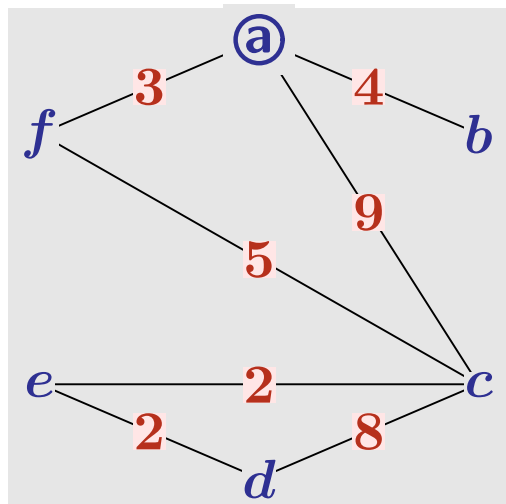
v:

$$\frac{f}{3}$$

$f \xrightarrow{3} \mathbf{a}$: nic, bo **a** już obsłużone

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od ⓐ:

<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
4	9	$+\infty$	$+\infty$	3

wyliczone odległości od ⓐ:

ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	9			3

minimum

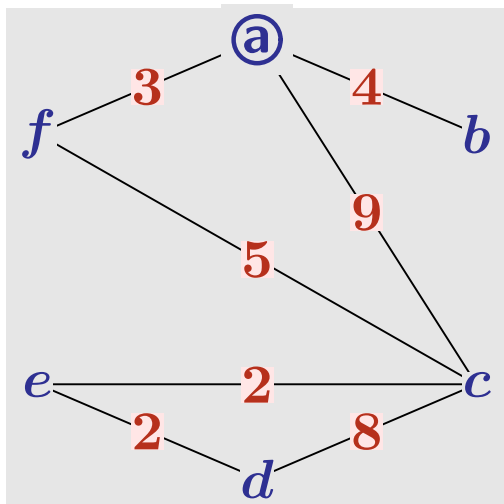
v:

$$\frac{f}{3}$$

$f \xrightarrow{3} \textcircled{a}$: nic, bo ⓐ już obsłużone
 $f \xrightarrow{5} c$: $3 + 5 < 9$ więc 8 do *c*

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od ⓐ:

<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
4	8	$+\infty$	$+\infty$	3

wyliczone odległości od ⓐ:

ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8			3

minimum

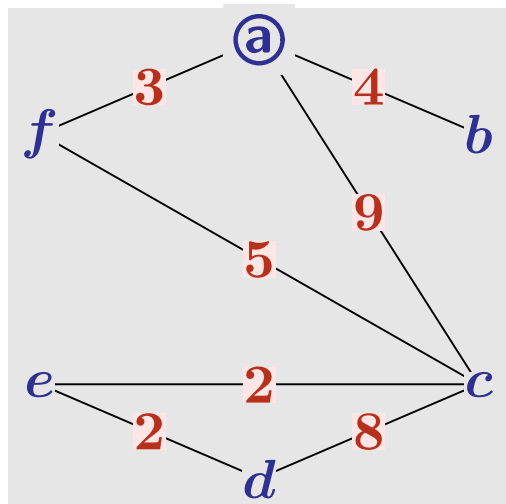
v:

$$\frac{f}{3}$$

$f \xrightarrow{3} \textcircled{a}$: nic, bo ⓐ już obsłużone
 $f \xrightarrow{5} c$: $3 + 5 < 9$ więc 8 do *c*

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

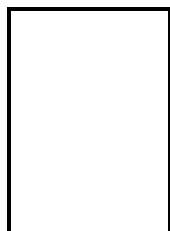
<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
4	8	$+\infty$	$+\infty$

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8			3

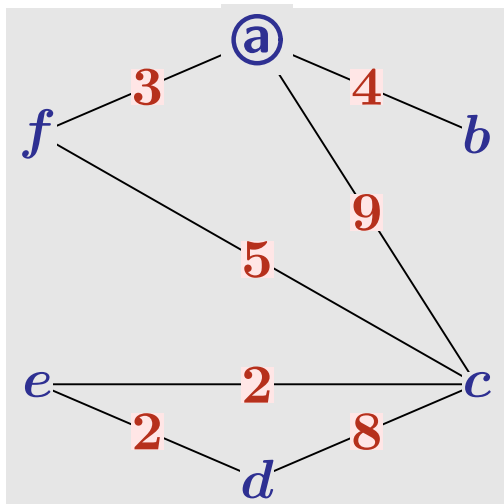
minimum

v:



Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
4	8	$+\infty$	$+\infty$

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8			3

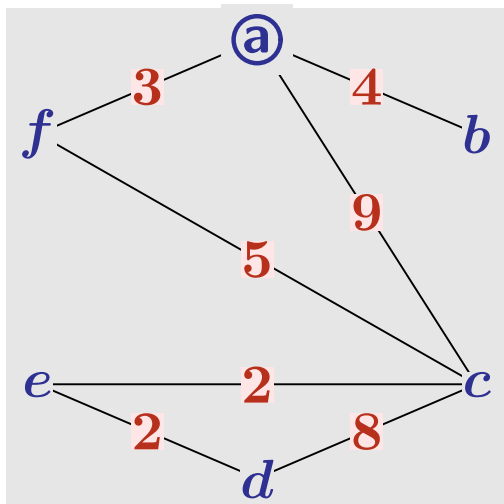
minimum

v:

$\frac{b}{4}$

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od Ⓐ:

<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
4	8	$+\infty$	$+\infty$

wyliczone odległości od Ⓐ:

Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8			3

minimum

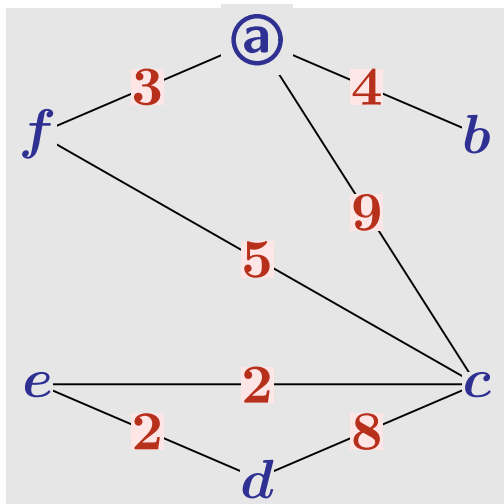
v:

$\frac{b}{4}$

$b \xrightarrow{4} \text{Ⓐ}$: nic, bo Ⓐ już obsłużone

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

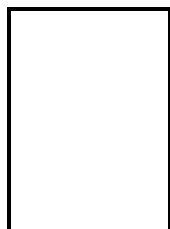
<i>c</i>	<i>d</i>	<i>e</i>
8	$+\infty$	$+\infty$

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8			3

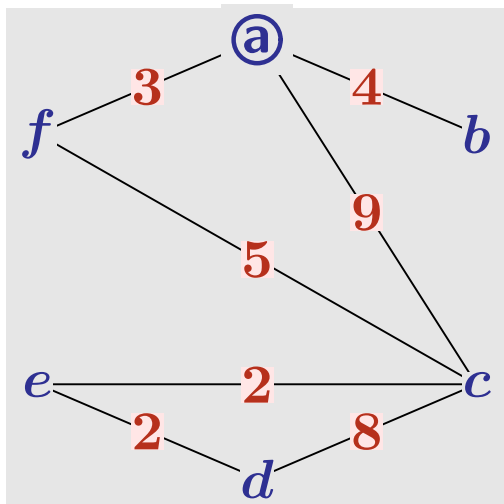
minimum

v:



Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

<i>c</i>	<i>d</i>	<i>e</i>
8	$+\infty$	$+\infty$

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8			3

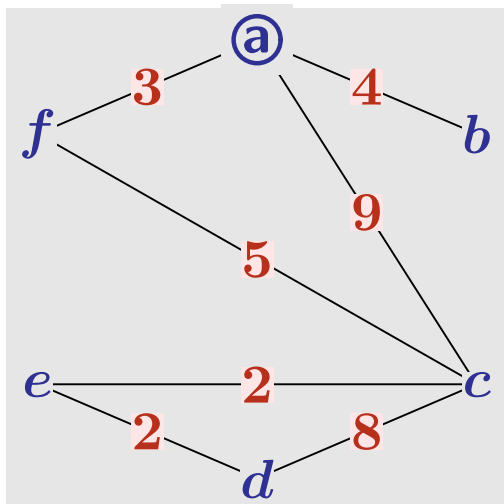
minimum

v:

$\frac{c}{8}$

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

<i>c</i>	<i>d</i>	<i>e</i>
8	$+\infty$	$+\infty$

minimum

v:

$\frac{c}{8}$

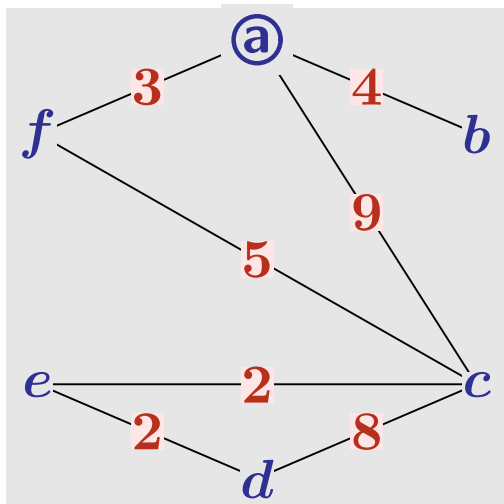
wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8			3

$c \xrightarrow{9} \mathbf{a}$: nic, bo **a** już obsłużone

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

<i>c</i>	<i>d</i>	<i>e</i>
8	$+\infty$	$+\infty$

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8			3

minimum

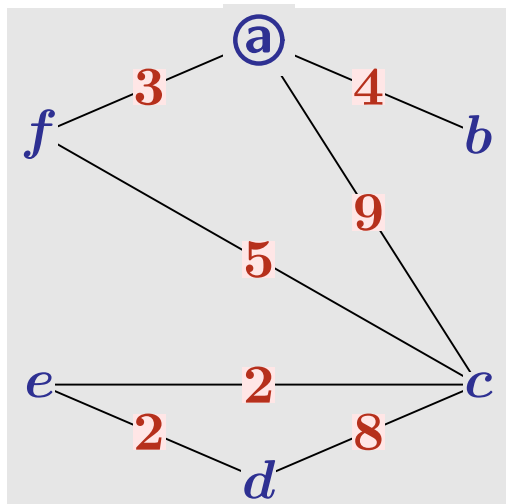
v:

$\frac{c}{8}$

$c \xrightarrow{9} \mathbf{a}$: nic, bo **a** już obsłużone
 $c \xrightarrow{8} d$: $8 + 8 < +\infty$ więc 16 do *d*

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

<i>c</i>	<i>d</i>	<i>e</i>
8	$+\infty$	$+\infty$

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8			3

minimum

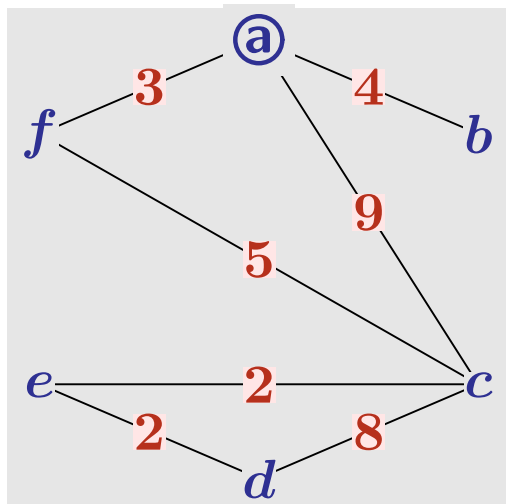
v:

$\frac{c}{8}$

$c \xrightarrow{9} \mathbf{a}$: nic, bo **a** już obsłużone
 $c \xrightarrow{8} d$: $8 + 8 < +\infty$ więc 16 do *d*
 $c \xrightarrow{2} e$: $8 + 2 < +\infty$ więc 10 do *e*

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

<i>c</i>	<i>d</i>	<i>e</i>
8	$+\infty$	$+\infty$

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8			3

minimum

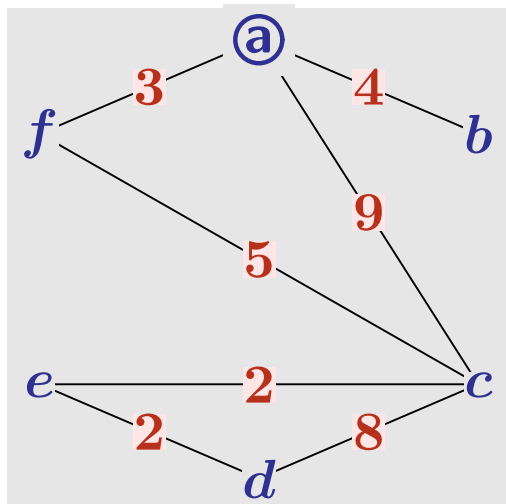
v:

$\frac{c}{8}$

$c \xrightarrow{9} \mathbf{a}$: nic, bo **a** już obsłużone
 $c \xrightarrow{8} d$: $8 + 8 < +\infty$ więc 16 do *d*
 $c \xrightarrow{2} e$: $8 + 2 < +\infty$ więc 10 do *e*
 $c \xrightarrow{5} f$: $8 + 5 \geq 3$ więc nic

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

<i>c</i>	<i>d</i>	<i>e</i>
8	16	10

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8	16	10	3

minimum

v:

$\frac{c}{8}$

$c \xrightarrow{9} \mathbf{a}$: nic, bo **a** już obsługiwane

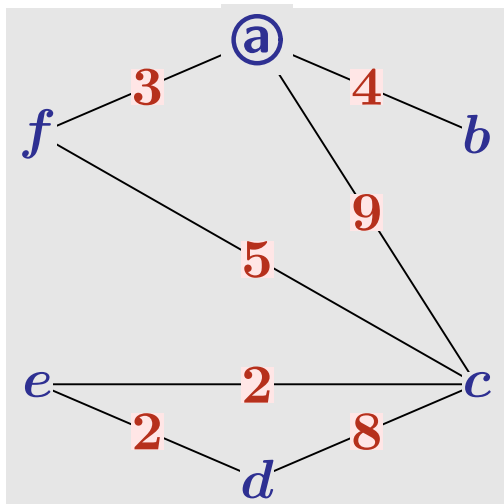
$c \xrightarrow{8} d$: $8 + 8 < +\infty$ więc 16 do *d*

$c \xrightarrow{2} e$: $8 + 2 < +\infty$ więc 10 do *e*

$c \xrightarrow{5} f$: $8 + 5 \geq 3$ więc nic

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od Ⓐ:

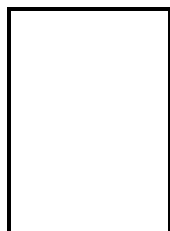
<i>d</i>	<i>e</i>
16	10

wyliczone odległości od Ⓐ:

Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8	16	10	3

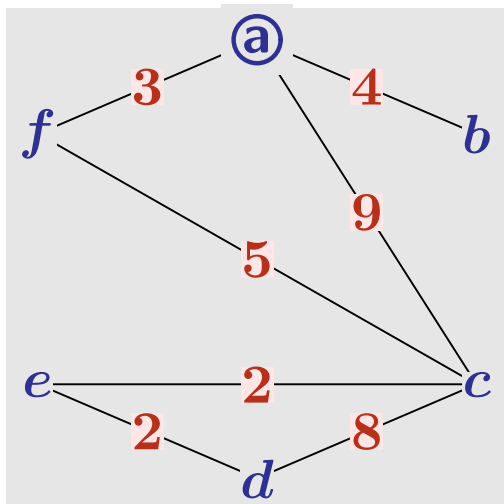
minimum

v:



Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od ⓐ:

<i>d</i>	<i>e</i>
16	10

wyliczone odległości od ⓐ:

ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8	16	10	3

minimum

v:

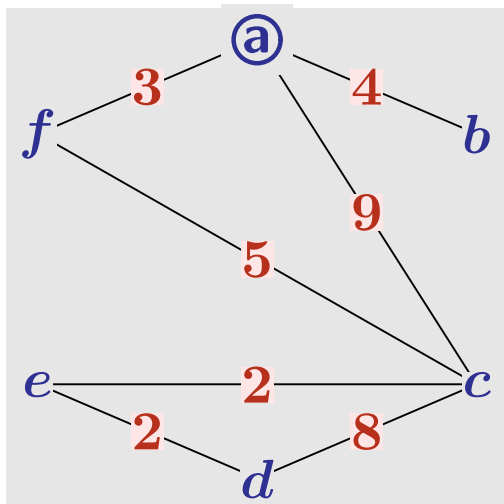
<i>e</i>
10

$e \xrightarrow{2} c : 10 + 2 \geq 8$ więc nic

$e \xrightarrow{2} d : 10 + 2 < 16$ więc 12 do *d*

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

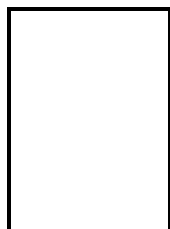
<i>d</i>
12

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8	12	10	3

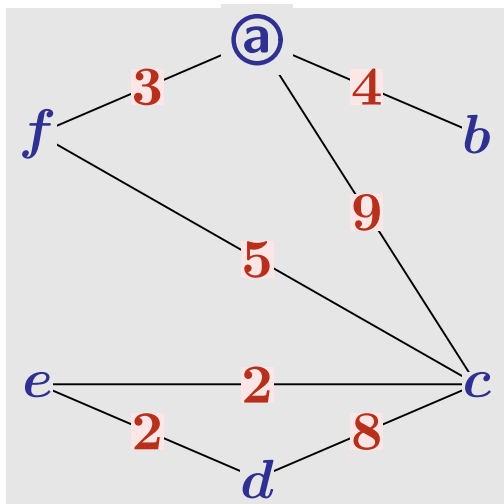
minimum

v:



Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od Ⓐ:

$\frac{d}{12}$

wyliczone odległości od Ⓐ:

Ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8	12	10	3

minimum

v:

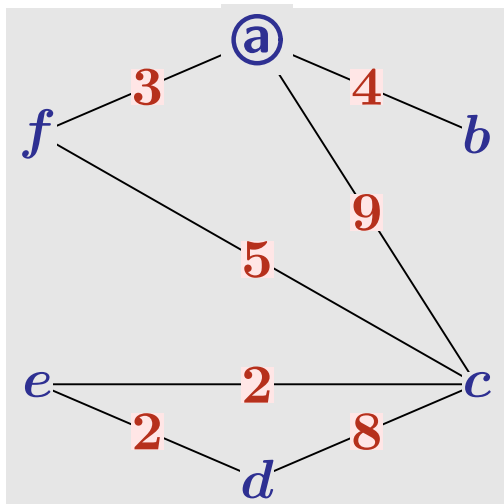
$\frac{d}{12}$

$d \xrightarrow{8} c : 12 + 8 \geq 8$ więc nie

$d \xrightarrow{2} e : 12 + 2 \geq 10$ więc nie

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
ⓐ		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od ⓐ:

wyliczone odległości od ⓐ:

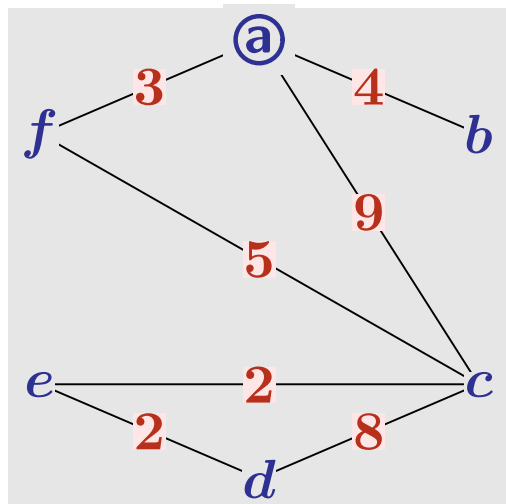
ⓐ	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8	12	10	3

minimum

v:

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8	12	10	3

minimum

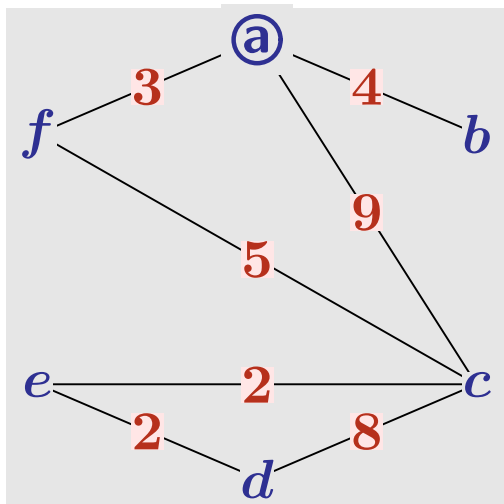
v:

pusto, więc

KONIEC

Grafy ważone — algorytm Dijkstry

graf:



odl. bezpośrednie:

	a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
a		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

tymczasowe odległości od **a**:

wyliczone odległości od **a**:

a	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	4	8	12	10	3

minimum

v:

pusto, więc
KONIEC
tu są wyniki

Grafy ważone — algorytm Dijkstry

Sam graf może być reprezentowany przez *macierz sąsiedztwa* (jak wyżej), ale lepiej przez *listy sąsiedztw*.

Grafy ważone — algorytm Dijkstry

Sam graf może być reprezentowany przez *macierz sąsiedztwa* (jak wyżej), ale lepiej przez *listy sąsiedztw*.

Informacje opisane wyżej jako „**tymczasowe odległości**” zwykle implementuje się przy pomocy kolejki z priorytetami.

Grafy ważone — algorytm Dijkstry

Sam graf może być reprezentowany przez *macierz sąsiedztwa* (jak wyżej), ale lepiej przez *listy sąsiedztw*.

Informacje opisane wyżej jako „**tymczasowe odległości**” zwykle implementuje się przy pomocy kolejki z priorytetami. Wierzchołek o minimalnej wadze znajduje się na jej początku.

Grafy ważone — algorytm Dijkstry

Sam graf może być reprezentowany przez *macierz sąsiedztwa* (jak wyżej), ale lepiej przez *listy sąsiedztw*.

Informacje opisane wyżej jako „**tymczasowe odległości**” zwykle implementuje się przy pomocy kolejki z priorytetami. Wierzchołek o minimalnej wadze znajduje się na jej początku.

Łatwo wzbogacić ten algorytm tak, żeby **wskazywał** drogę o najniższej wadze, a **nie tylko podawał jej wagę**.

Reprezentacje grafu (informatyka)

Macierz sąsiedztwa

Reprezentacje grafu (informatyka)

Macierz sąsiedztwa

— dwuwymiarowa tablica $T \subseteq V \times V$ (czyli indeksowana parami wierzchołków)

Reprezentacje grafu (informatyka)

Macierz sąsiedztwa

— dwuwymiarowa tablica $T \subseteq V \times V$ (czyli indeksowana parami wierzchołków)

$$T[v_1][v_2] = \left\{ \right.$$

Reprezentacje grafu (informatyka)

Macierz sąsiedztwa

— dwuwymiarowa tablica $T \subseteq V \times V$ (czyli indeksowana parami wierzchołków)

$$T[v_1][v_2] = \begin{cases} \text{dla grafu bez wag — jeśli } \{v_1, v_2\} \in E, \text{ to } \mathbf{1} \\ \end{cases}$$

Reprezentacje grafu (informatyka)

Macierz sąsiedztwa

— dwuwymiarowa tablica $T \subseteq V \times V$ (czyli indeksowana parami wierzchołków)

$$T[v_1][v_2] = \begin{cases} \text{dla grafu bez wag — jeśli } \{v_1, v_2\} \in E, \text{ to } \mathbf{1}; \\ \text{a jeśli nie, to oznaczenie braku krawędzi} \end{cases}$$

Reprezentacje grafu (informatyka)

Macierz sąsiedztwa

— dwuwymiarowa tablica $T \subseteq V \times V$ (czyli indeksowana parami wierzchołków)

$$T[v_1][v_2] = \begin{cases} \text{dla grafu bez wag — jeśli } \{v_1, v_2\} \in E, \text{ to } \mathbf{1}; \\ \qquad \qquad \qquad \text{a jeśli nie, to oznaczenie braku krawędzi} \\ \text{dla grafu z wagami — jeśli } \{v_1, v_2\} \in E, \text{ to waga krawędzi} \end{cases}$$

Reprezentacje grafu (informatyka)

Macierz sąsiedztwa

— dwuwymiarowa tablica $T \subseteq V \times V$ (czyli indeksowana parami wierzchołków)

$$T[v_1][v_2] = \begin{cases} \text{dla grafu bez wag — jeśli } \{v_1, v_2\} \in E, \text{ to } \mathbf{1}; \\ \qquad \qquad \qquad \text{a jeśli nie, to oznaczenie braku krawędzi} \\ \text{dla grafu z wagami — jeśli } \{v_1, v_2\} \in E, \text{ to waga krawędzi;} \\ \qquad \qquad \qquad \text{a jeśli nie, to oznaczenie braku krawędzi} \end{cases}$$

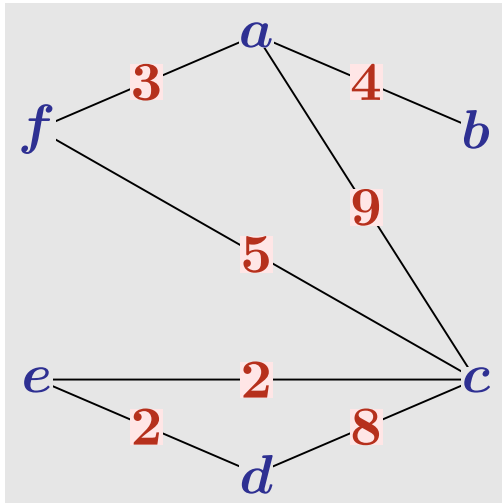
Reprezentacje grafu (informatyka)

Macierz sąsiedztwa

— dwuwymiarowa tablica $T \subseteq V \times V$ (czyli indeksowana parami wierzchołków)

$$T[v_1][v_2] = \begin{cases} \text{dla grafu bez wag — jeśli } \{v_1, v_2\} \in E, \text{ to } \mathbf{1}; \\ \quad \text{a jeśli nie, to oznaczenie braku krawędzi} \\ \text{dla grafu z wagami — jeśli } \{v_1, v_2\} \in E, \text{ to waga krawędzi;} \\ \quad \text{a jeśli nie, to oznaczenie braku krawędzi} \end{cases}$$

graf:



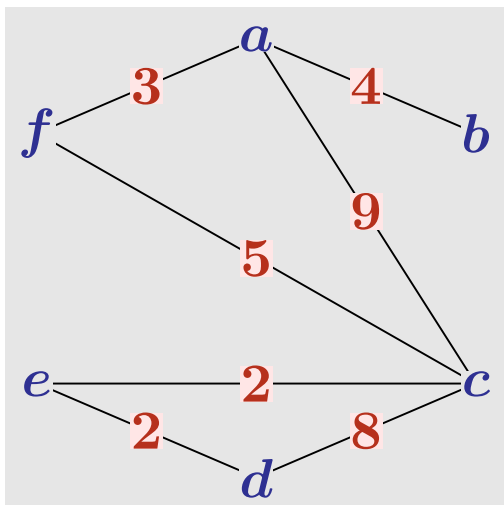
Reprezentacje grafu (informatyka)

Macierz sąsiedztwa

— dwuwymiarowa tablica $T \subseteq V \times V$ (czyli indeksowana parami wierzchołków)

$$T[v_1][v_2] = \begin{cases} \text{dla grafu bez wag — jeśli } \{v_1, v_2\} \in E, \text{ to } \mathbf{1}; \\ \text{a jeśli nie, to oznaczenie braku krawędzi} \\ \text{dla grafu z wagami — jeśli } \{v_1, v_2\} \in E, \text{ to waga krawędzi;} \\ \text{a jeśli nie, to oznaczenie braku krawędzi} \end{cases}$$

graf:



macierz:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
<i>a</i>		4	9			3
<i>b</i>	4					
<i>c</i>	9			8	2	5
<i>d</i>			8		2	
<i>e</i>			2	2		
<i>f</i>	3		5			

Reprezentacje grafu (informatyka)

Listy sąsiedztw

Reprezentacje grafu (informatyka)

Listy sąsiedztw

— tablica indeksowana wierzchołkami

Reprezentacje grafu (informatyka)

Listy sąsiedztw

— tablica indeksowana wierzchołkami

$S[v]$ = lista sąsiadów wierzchołka v

Reprezentacje grafu (informatyka)

Listy sąsiedztw

— tablica indeksowana wierzchołkami

$S[v]$ = lista sąsiadów wierzchołka v
a dla grafu z wagami — lista par (sąsiad, waga krawędzi)

Reprezentacje grafu (informatyka)

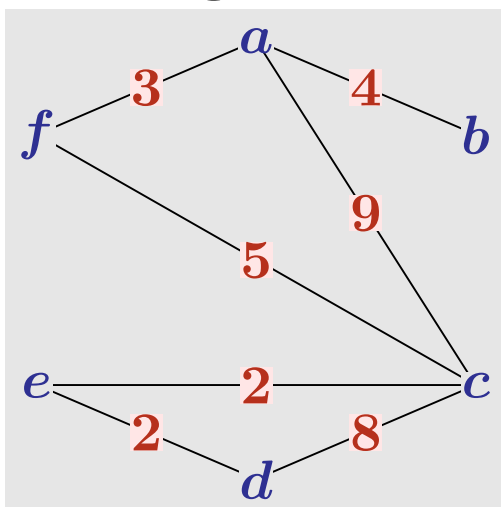
Listy sąsiedztw

— tablica indeksowana wierzchołkami

$S[v]$ = lista sąsiadów wierzchołka v

a dla grafu z wagami — lista par (sąsiad, waga krawędzi)

graf:



Reprezentacje grafu (informatyka)

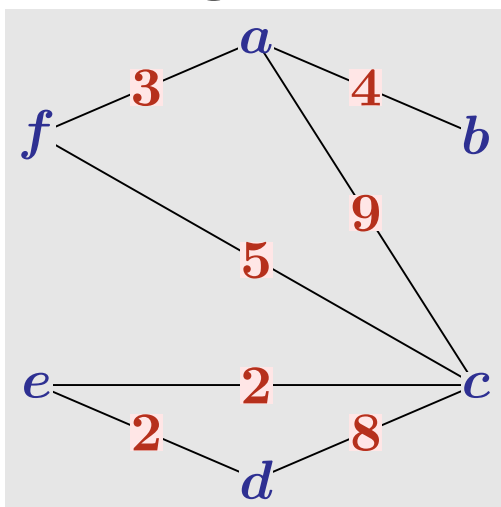
Listy sąsiedztw

— tablica indeksowana wierzchołkami

$S[v]$ = lista sąsiadów wierzchołka v

a dla grafu z wagami — lista par (sąsiad, waga krawędzi)

graf:



tablica list:

a	
b	
c	
d	
e	
f	

Reprezentacje grafu (informatyka)

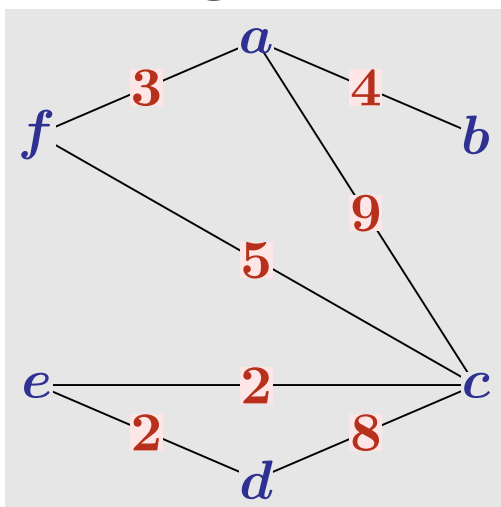
Listy sąsiedztw

— tablica indeksowana wierzchołkami

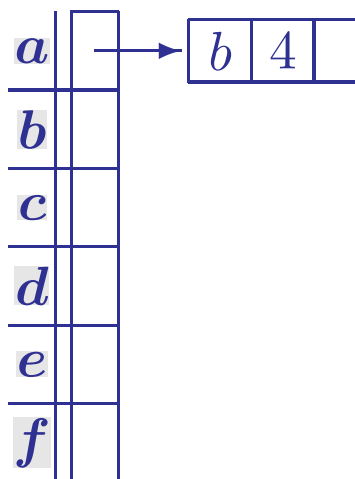
$S[v]$ = lista sąsiadów wierzchołka v

a dla grafu z wagami — lista par (sąsiad, waga krawędzi)

graf:



tablica list:



Reprezentacje grafu (informatyka)

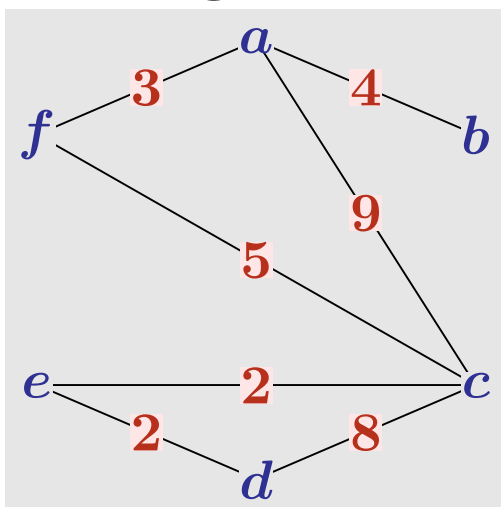
Listy sąsiedztw

— tablica indeksowana wierzchołkami

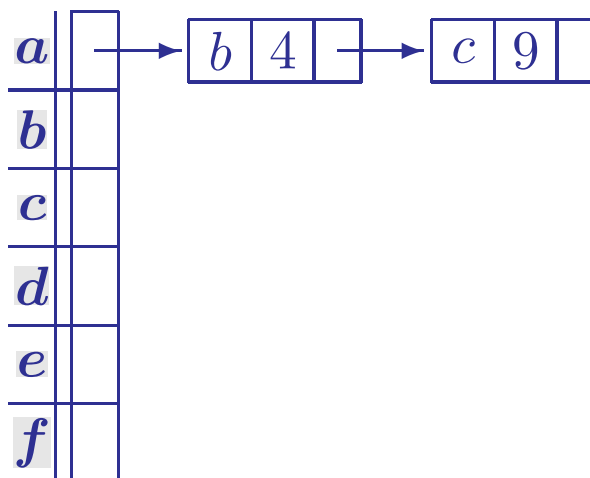
$S[v]$ = lista sąsiadów wierzchołka v

a dla grafu z wagami — lista par (sąsiad, waga krawędzi)

graf:



tablica list:



Reprezentacje grafu (informatyka)

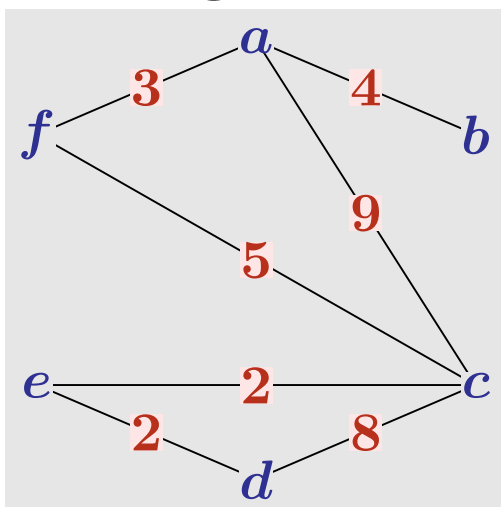
Listy sąsiedztw

— tablica indeksowana wierzchołkami

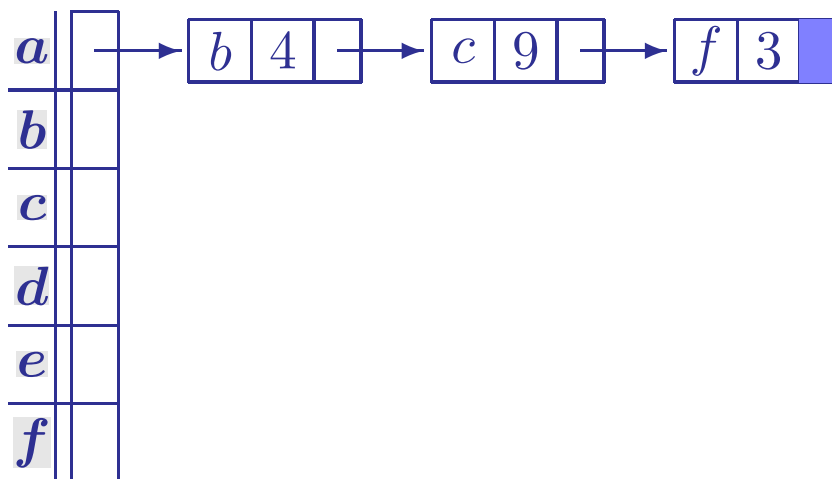
$S[v]$ = lista sąsiadów wierzchołka v

a dla grafu z wagami — lista par (sąsiad, waga krawędzi)

graf:



tablica list:



Reprezentacje grafu (informatyka)

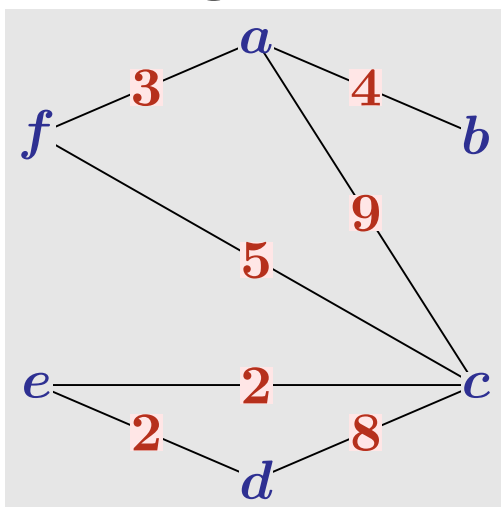
Listy sąsiedztw

— tablica indeksowana wierzchołkami

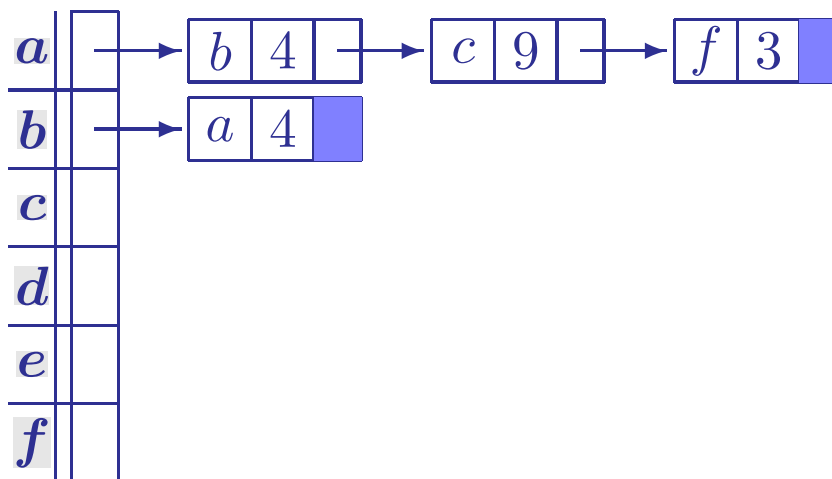
$S[v]$ = lista sąsiadów wierzchołka v

a dla grafu z wagami — lista par (sąsiad, waga krawędzi)

graf:



tablica list:



Reprezentacje grafu (informatyka)

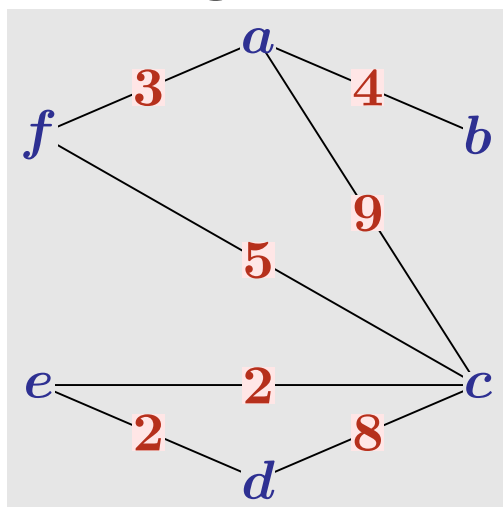
Listy sąsiedztw

— tablica indeksowana wierzchołkami

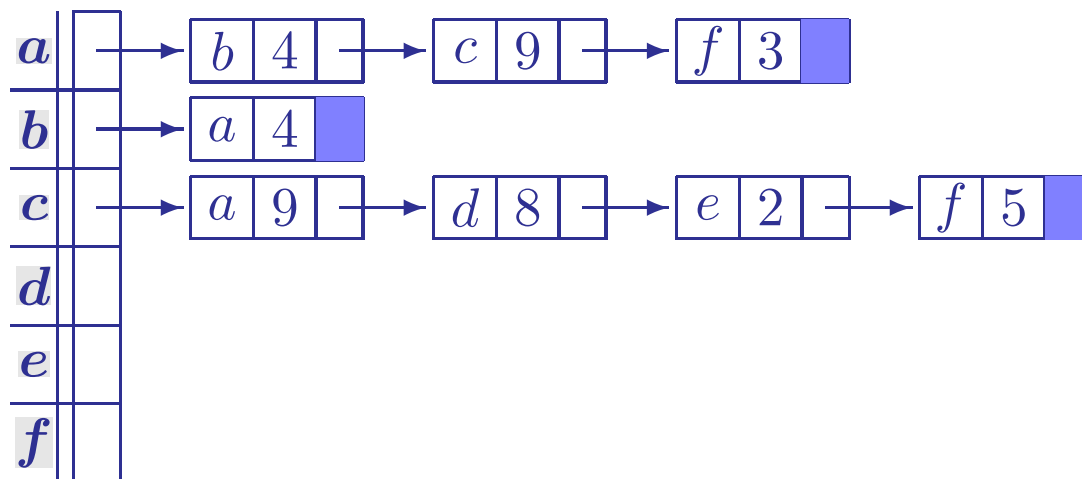
$S[v]$ = lista sąsiadów wierzchołka v

a dla grafu z wagami — lista par (sąsiad, waga krawędzi)

graf:



tablica list:



Reprezentacje grafu (informatyka)

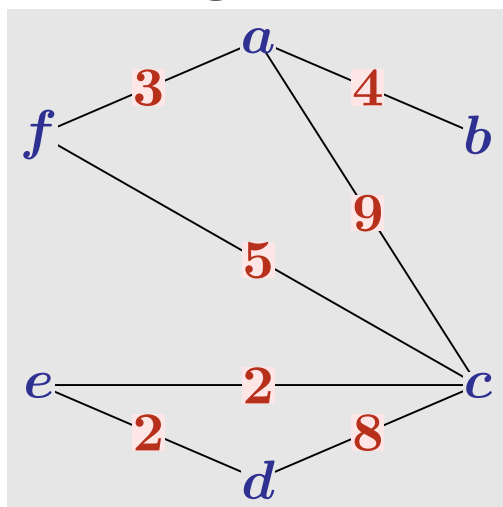
Listy sąsiedztw

— tablica indeksowana wierzchołkami

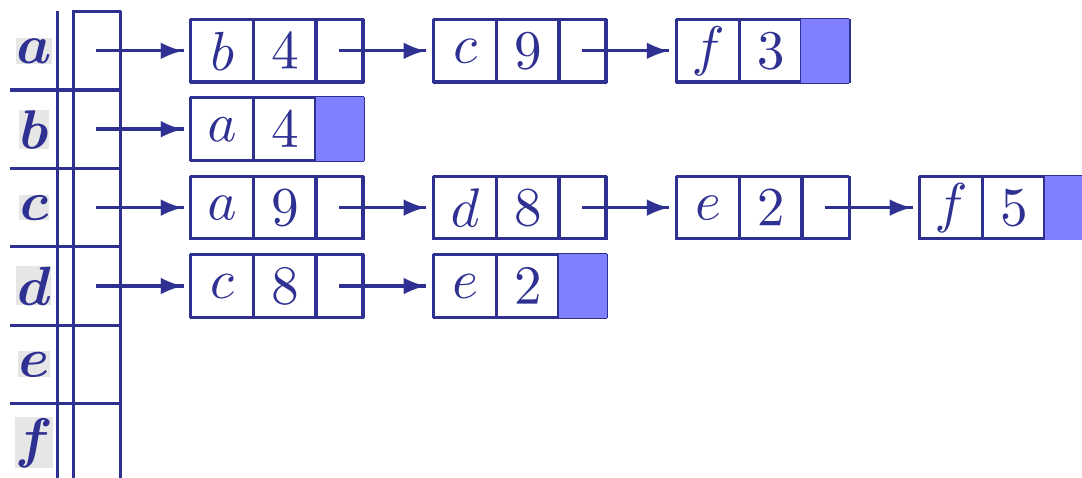
$S[v]$ = lista sąsiadów wierzchołka v

a dla grafu z wagami — lista par (sąsiad, waga krawędzi)

graf:



tablica list:



Reprezentacje grafu (informatyka)

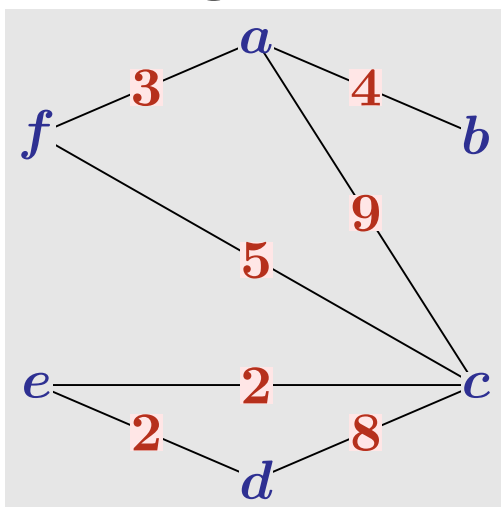
Listy sąsiedztw

— tablica indeksowana wierzchołkami

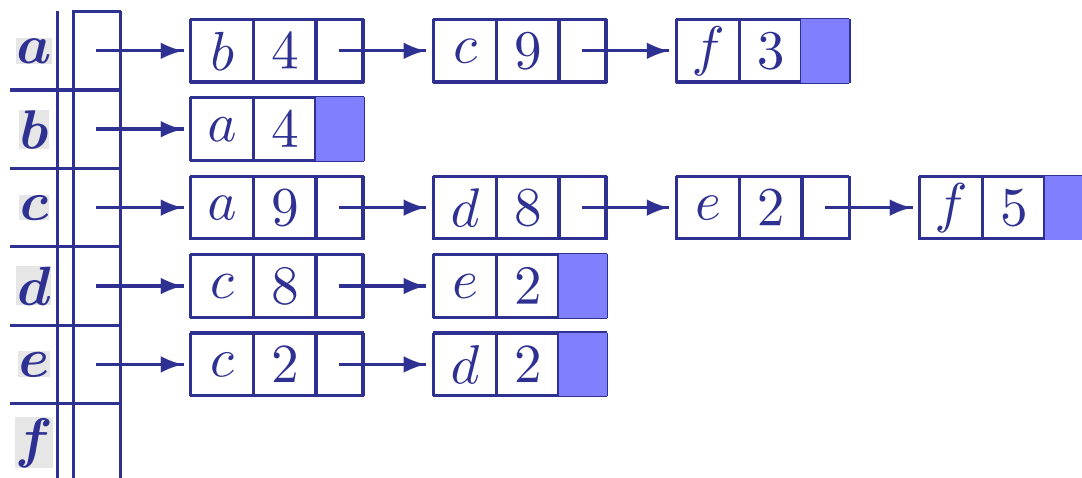
$S[v]$ = lista sąsiadów wierzchołka v

a dla grafu z wagami — lista par (sąsiad, waga krawędzi)

graf:



tablica list:



Reprezentacje grafu (informatyka)

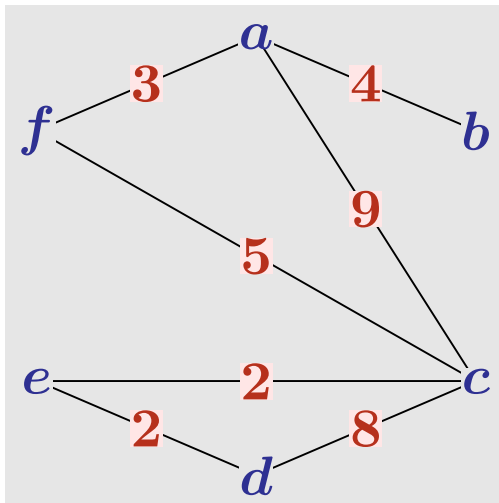
Listy sąsiedztw

— tablica indeksowana wierzchołkami

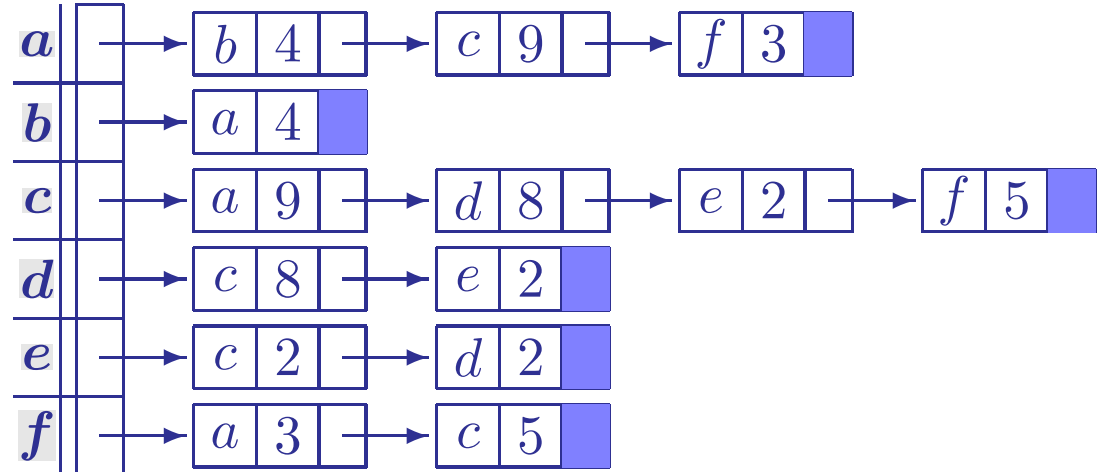
$S[v]$ = lista sąsiadów wierzchołka v

a dla grafu z wagami — lista par (sąsiad, waga krawędzi)

graf:



tablica list:



Reprezentacje grafu (informatyka)

Porównanie

Reprezentacje grafu (informatyka)

Porównanie:

	macierz sąsiedztwa	listy sąsiedztw
redundancja:	symetria: $T[v_1][v_2] = T[v_2][v_1]$	

Reprezentacje grafu (informatyka)

Porównanie:

	macierz sąsiedztwa	listy sąsiedztw
redundancja:	symetria: $T[v_1][v_2] = T[v_2][v_1]$	w liście $S[v_1]$ jest para $\langle v_2, k \rangle$ $\iff$ w liście $S[v_2]$ jest para $\langle v_1, k \rangle$

Reprezentacje grafu (informatyka)

Porównanie:

	macierz sąsiedztwa	listy sąsiedztw
redundancja:	symetria: $T[v_1][v_2] = T[v_2][v_1]$	w liście $S[v_1]$ jest para $\langle v_2, k \rangle$ $\iff$ w liście $S[v_2]$ jest para $\langle v_1, k \rangle$
dostęp do krawędzi $\{v_1, v_2\}$:	bezpośrednio dany	

Reprezentacje grafu (informatyka)

Porównanie:

	macierz sąsiedztwa	listy sąsiedztw
redundancja:	symetria: $T[v_1][v_2] = T[v_2][v_1]$	w liście $S[v_1]$ jest para $\langle v_2, k \rangle$ $\iff$ w liście $S[v_2]$ jest para $\langle v_1, k \rangle$
dostęp do krawędzi $\{v_1, v_2\}$:	bezpośrednio dany	wymaga przeszukania listy

Reprezentacje grafu (informatyka)

Porównanie:

	macierz sąsiedztwa	listy sąsiedztw
redundancja:	symetria: $T[v_1][v_2] = T[v_2][v_1]$	w liście $S[v_1]$ jest para $\langle v_2, k \rangle$ $\iff$ w liście $S[v_2]$ jest para $\langle v_1, k \rangle$
dostęp do krawędzi $\{v_1, v_2\}$:	bezpośrednio dany	wymaga przeszukania listy
znalezienie sąsiadów v :	wymaga przeszukania wiersza	

Reprezentacje grafu (informatyka)

Porównanie:

	macierz sąsiedztwa	listy sąsiedztw
redundancja:	symetria: $T[v_1][v_2] = T[v_2][v_1]$	w liście $S[v_1]$ jest para $\langle v_2, k \rangle$ $\iff$ w liście $S[v_2]$ jest para $\langle v_1, k \rangle$
dostęp do krawędzi $\{v_1, v_2\}$:	bezpośrednio dany	wymaga przeszukania listy
znalezienie sąsiadów v :	wymaga przeszukania wiersza	bezpośrednio dane

Reprezentacje grafu (informatyka)

Porównanie:

	macierz sąsiedztwa	listy sąsiedztw
redundancja:	symetria: $T[v_1][v_2] = T[v_2][v_1]$	w liście $S[v_1]$ jest para $\langle v_2, k \rangle$ $\iff$ w liście $S[v_2]$ jest para $\langle v_1, k \rangle$
dostęp do krawędzi $\{v_1, v_2\}$:	bezpośrednio dany	wymaga przeszukania listy
znalezienie sąsiadów v :	wymaga przeszukania wiersza	bezpośrednio dane
zajętość pamięci — kiedy warto:	warto w grafach „gęstych”	

Reprezentacje grafu (informatyka)

Porównanie:

	macierz sąsiedztwa	listy sąsiedztw
redundancja:	symetria: $T[v_1][v_2] = T[v_2][v_1]$	w liście $S[v_1]$ jest para $\langle v_2, k \rangle$ $\iff$ w liście $S[v_2]$ jest para $\langle v_1, k \rangle$
dostęp do krawędzi $\{v_1, v_2\}$:	bezpośrednio dany	wymaga przeszukania listy
znalezienie sąsiadów v :	wymaga przeszukania wiersza	bezpośrednio dane
zajętość pamięci — kiedy warto:	warto w grafach „gęstych” w „rzadkich” wielka macierz wypełniona niczym	

Reprezentacje grafu (informatyka)

Porównanie:

	macierz sąsiedztwa	listy sąsiedztw
redundancja:	symetria: $T[v_1][v_2] = T[v_2][v_1]$	w liście $S[v_1]$ jest para $\langle v_2, k \rangle$ $\iff$ w liście $S[v_2]$ jest para $\langle v_1, k \rangle$
dostęp do krawędzi $\{v_1, v_2\}$:	bezpośrednio dany	wymaga przeszukania listy
znalezienie sąsiadów v :	wymaga przeszukania wiersza	bezpośrednio dane
zajętość pamięci — kiedy warto:	warto w grafach „gęstych” w „rzadkich” wielka macierz wypełniona niczym	warto w grafach „rzadkich”

Reprezentacje grafu (informatyka)

Porównanie:

	macierz sąsiedztwa	listy sąsiedztw
redundancja:	symetria: $T[v_1][v_2] = T[v_2][v_1]$	w liście $S[v_1]$ jest para $\langle v_2, k \rangle$ $\iff$ w liście $S[v_2]$ jest para $\langle v_1, k \rangle$
dostęp do krawędzi $\{v_1, v_2\}$:	bezpośrednio dany	wymaga przeszukania listy
znalezienie sąsiadów v :	wymaga przeszukania wiersza	bezpośrednio dane
zajętość pamięci — kiedy warto:	warto w grafach „gęstych” w „rzadkich” wielka macierz wypełniona niczym	warto w grafach „rzadkich” w „gęstych” listy zajmują więcej miejsca niż wiersze macierzy