Algorytmy i struktury danych

wykład 6

Plan wykładu:

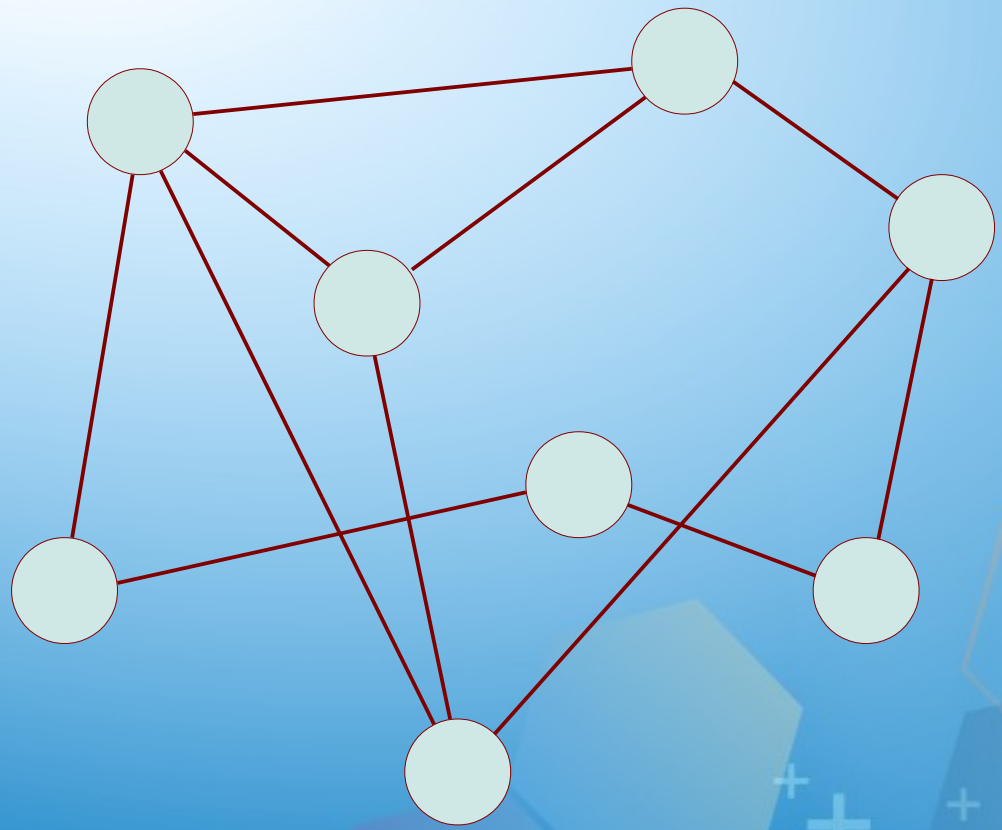
- Złożone struktury danych:
  - grafy,
  - klasy grafów,
  - reprezentacja grafów w pamięci komputera.
- Algorytmy grafowe I.

Grafy

Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

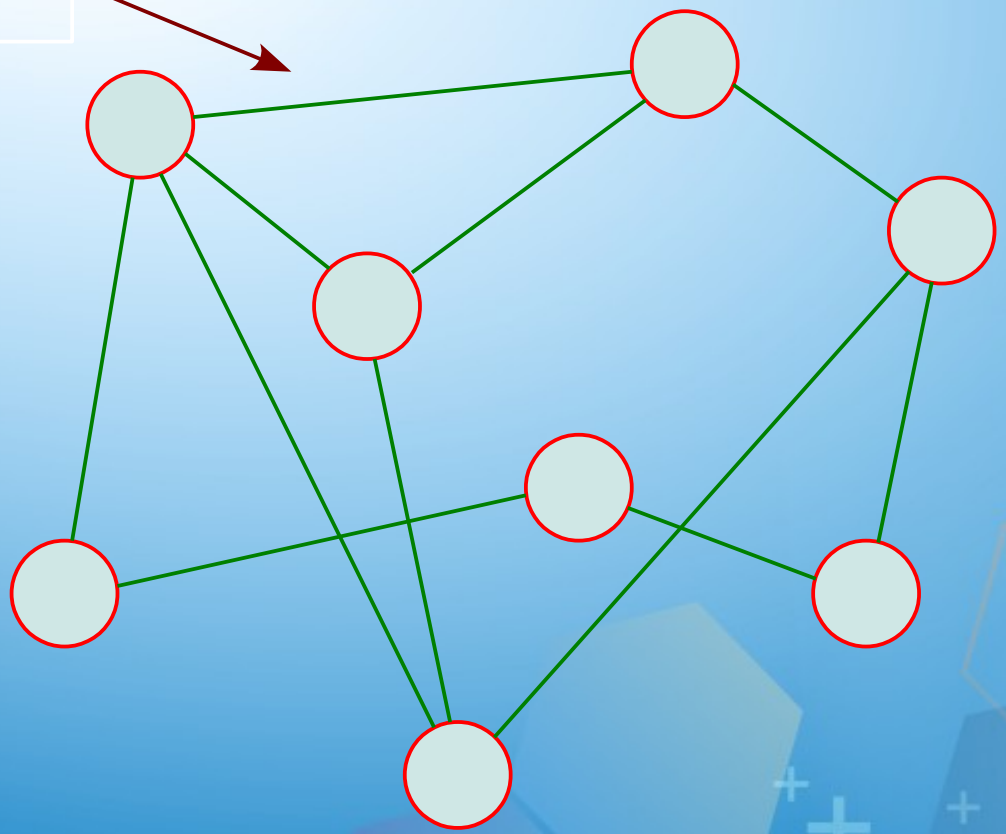
Budowa grafu:



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Budowa grafu:

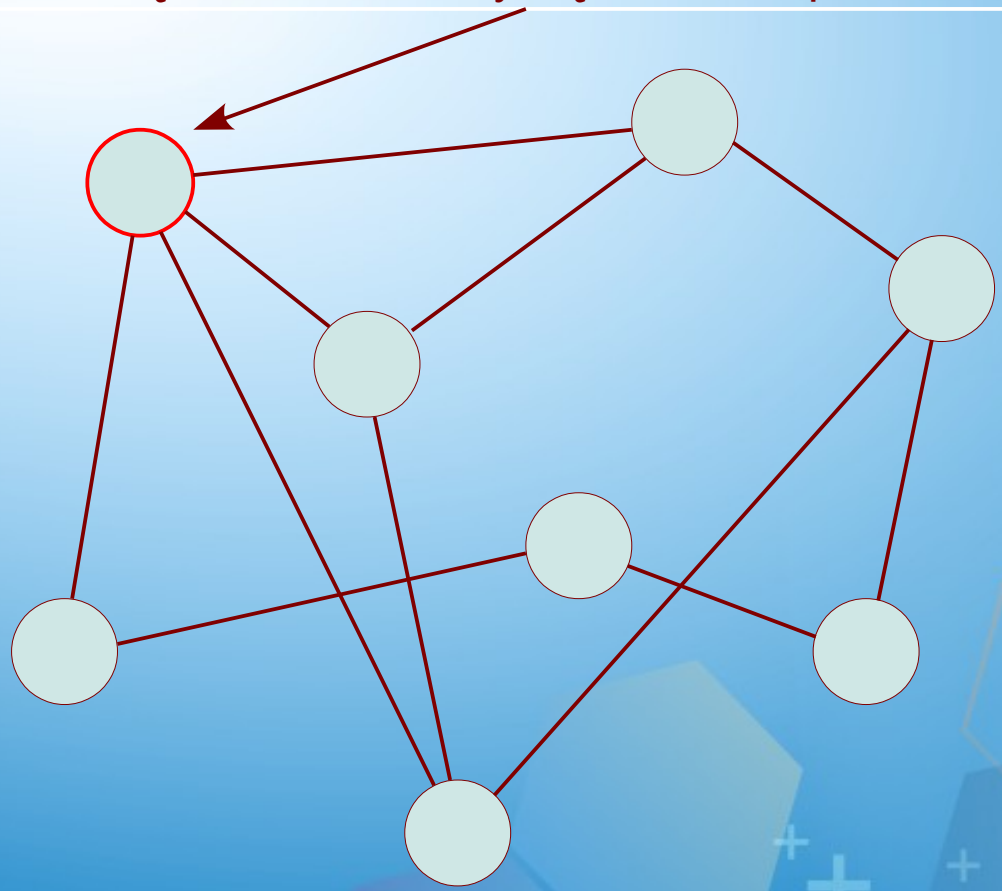
Graf G=(V, E) składa się z węzłów i krawędzi.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Budowa grafu:

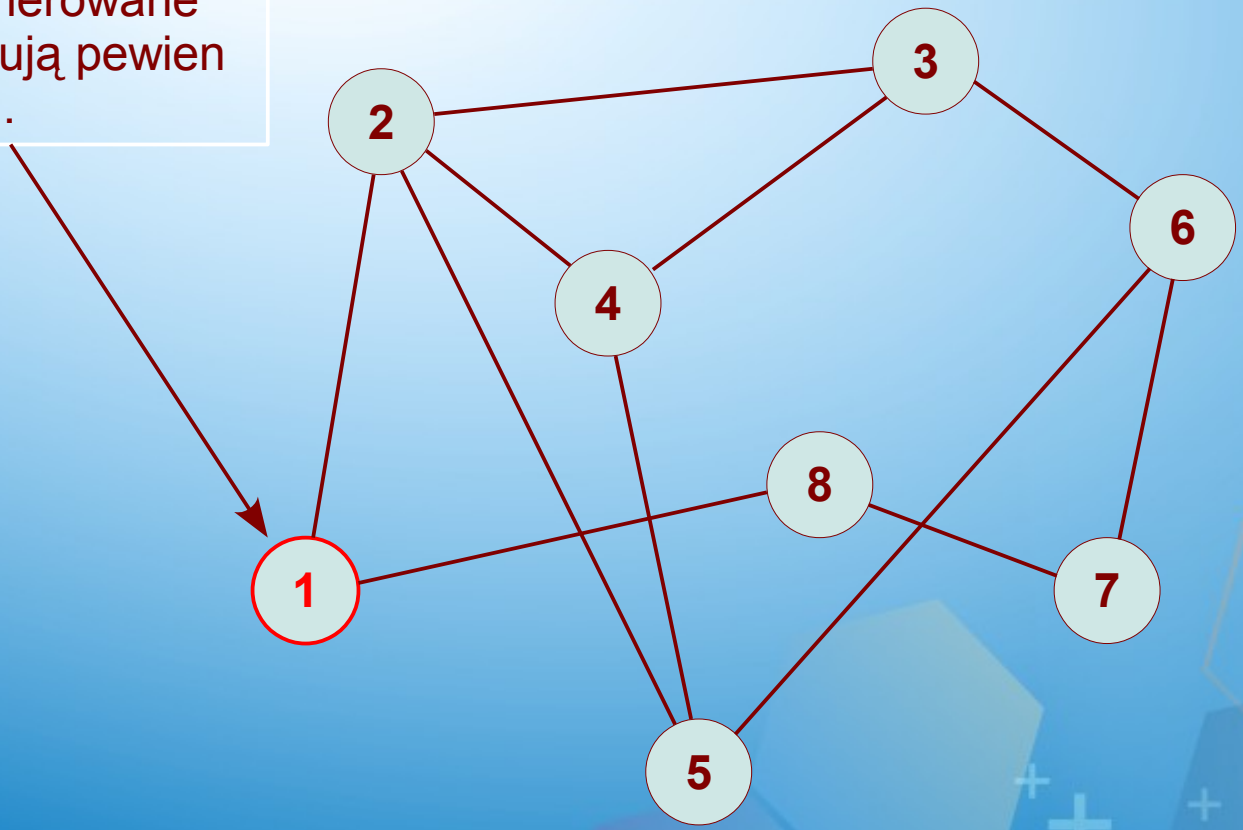
Węzeł posiada stopień, oznaczający liczbę sąsiadujących z nim krawędzi. Zaznaczony węzeł ma stopień 4.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Budowa grafu:

Węzły grafu są zazwyczaj ponumerowane i reprezentują pewien typ obiektu.

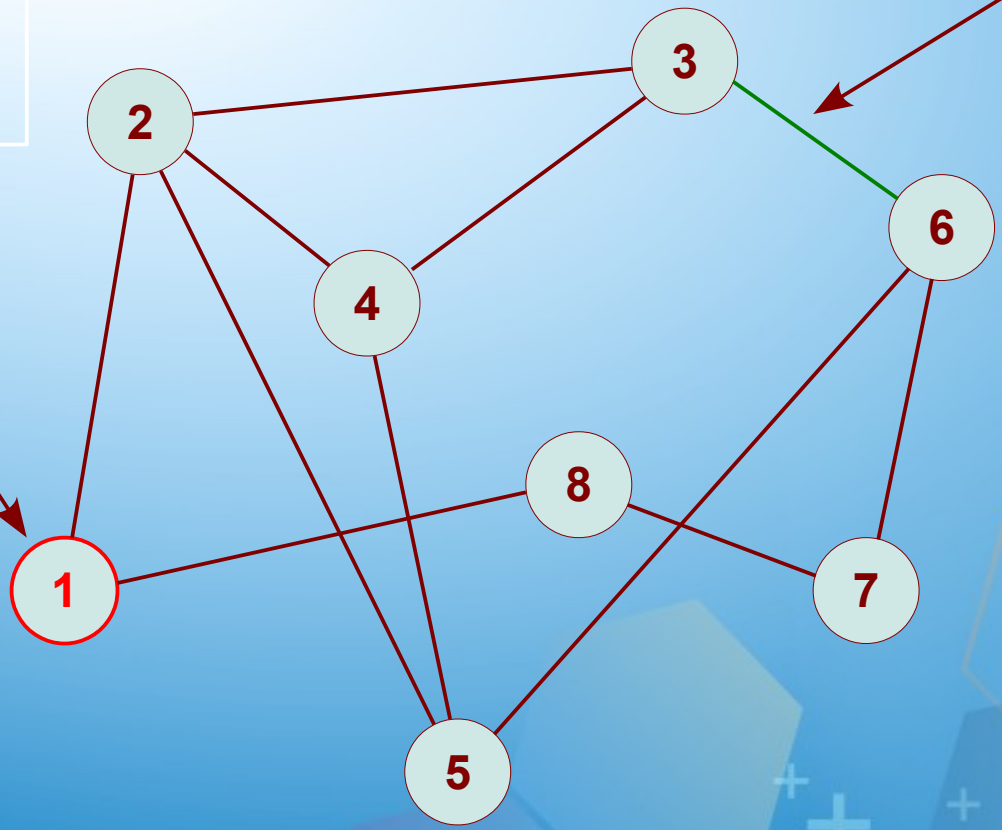


Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Budowa grafu:

Węzły grafu są zazwyczaj ponumerowane i reprezentują pewien typ obiektu...

... a krawędzie mogą reprezentować relacje między tymi obiektami.

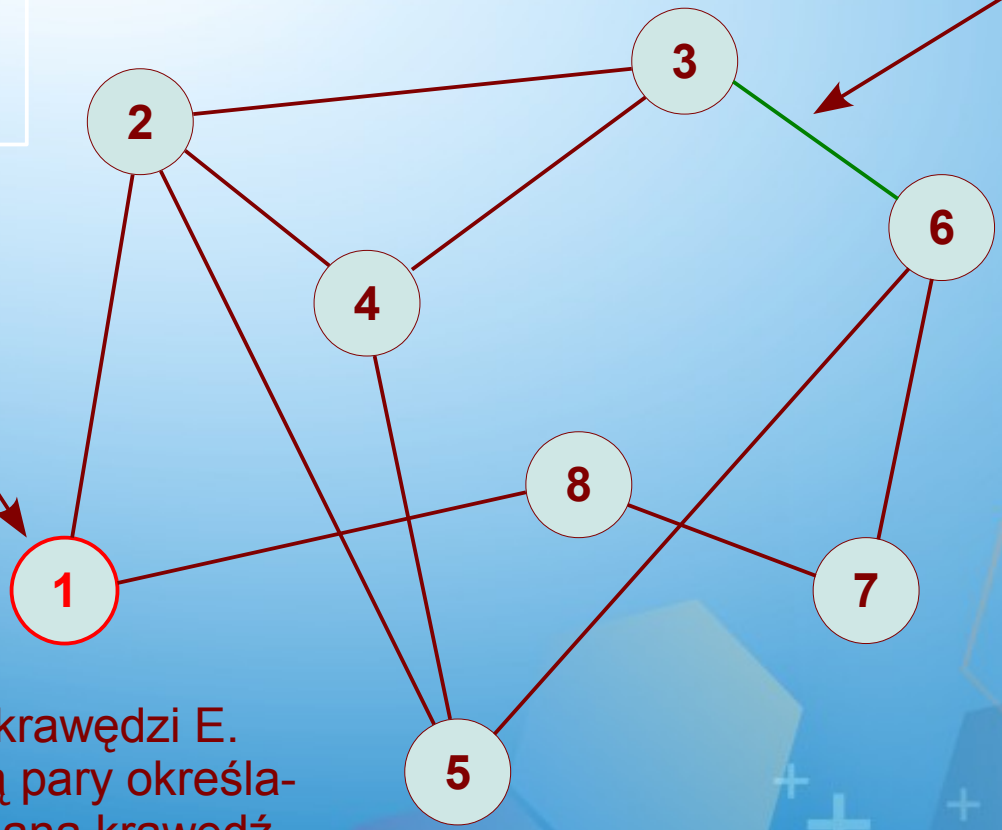


Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Budowa grafu:

Węzły grafu są zazwyczaj ponumerowane i reprezentują pewien typ obiektu...

... a krawędzie mogą reprezentować relacje między tymi obiektami.

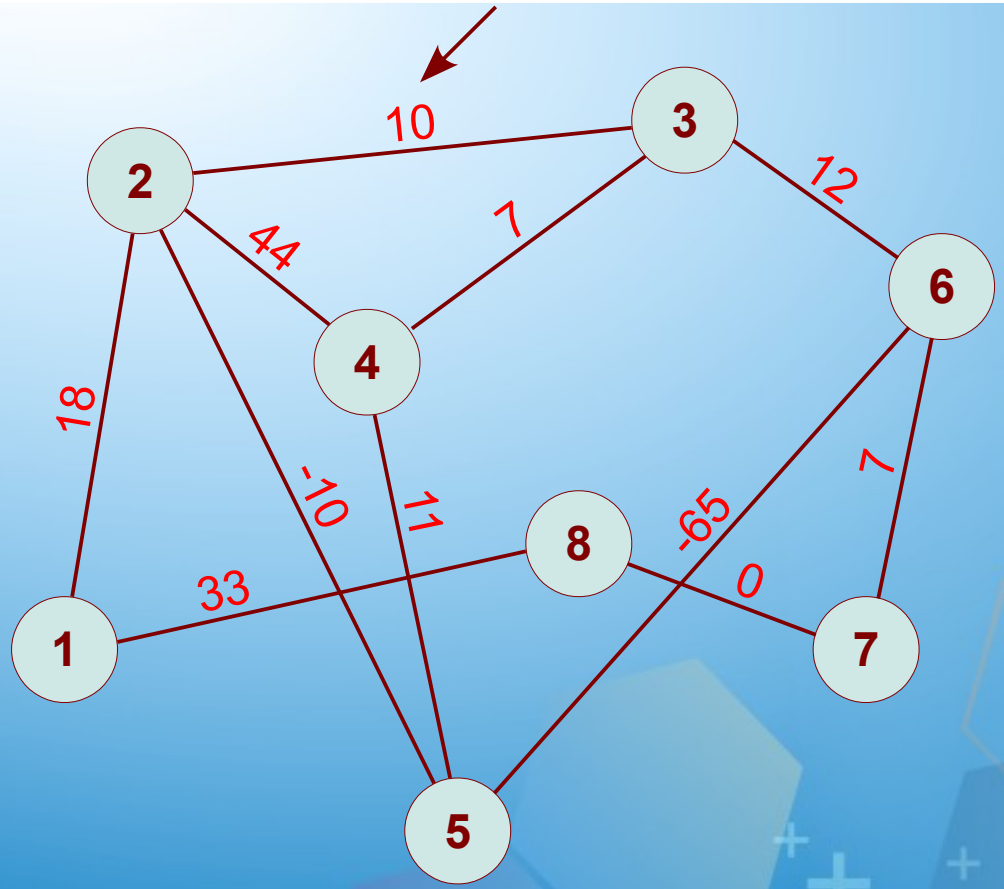


Graf opisuje się przez dwa zbiory: węzłów V i krawędzi E. Elementami zbioru E są pary określające które węzły łączy dana krawędź.

Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Budowa grafu:

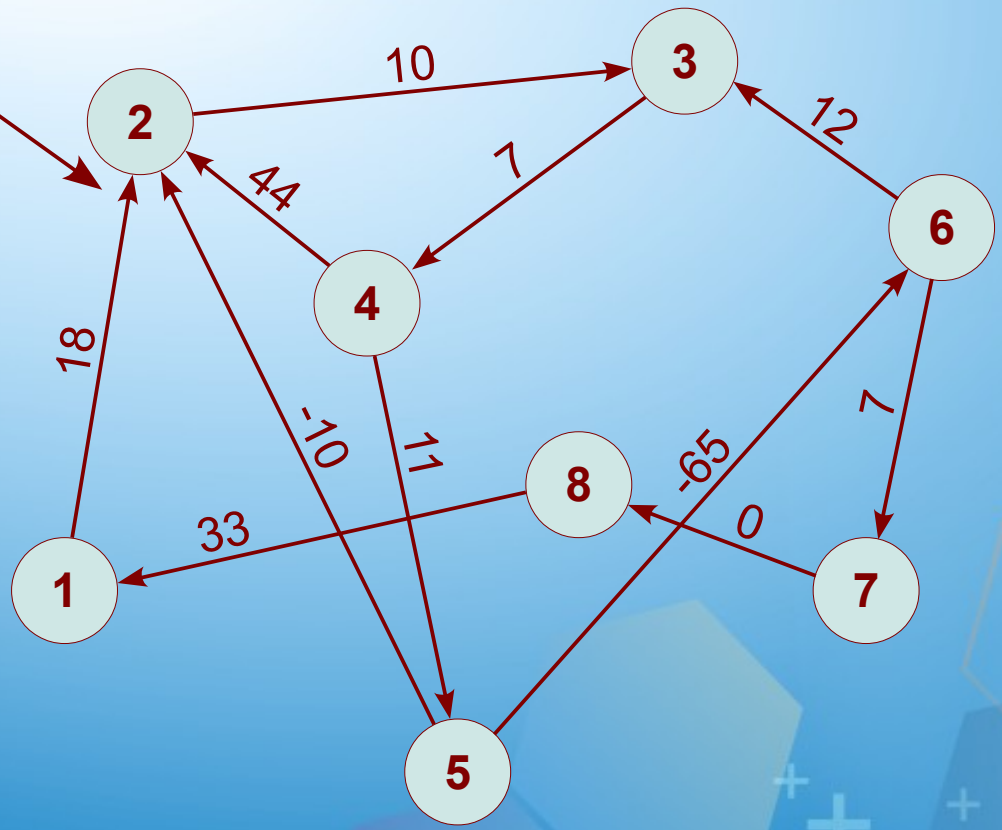
W ogólnym przypadku do krawędzi przypisuje się wagę, określając tym samym właściwości relacji.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Budowa grafu:

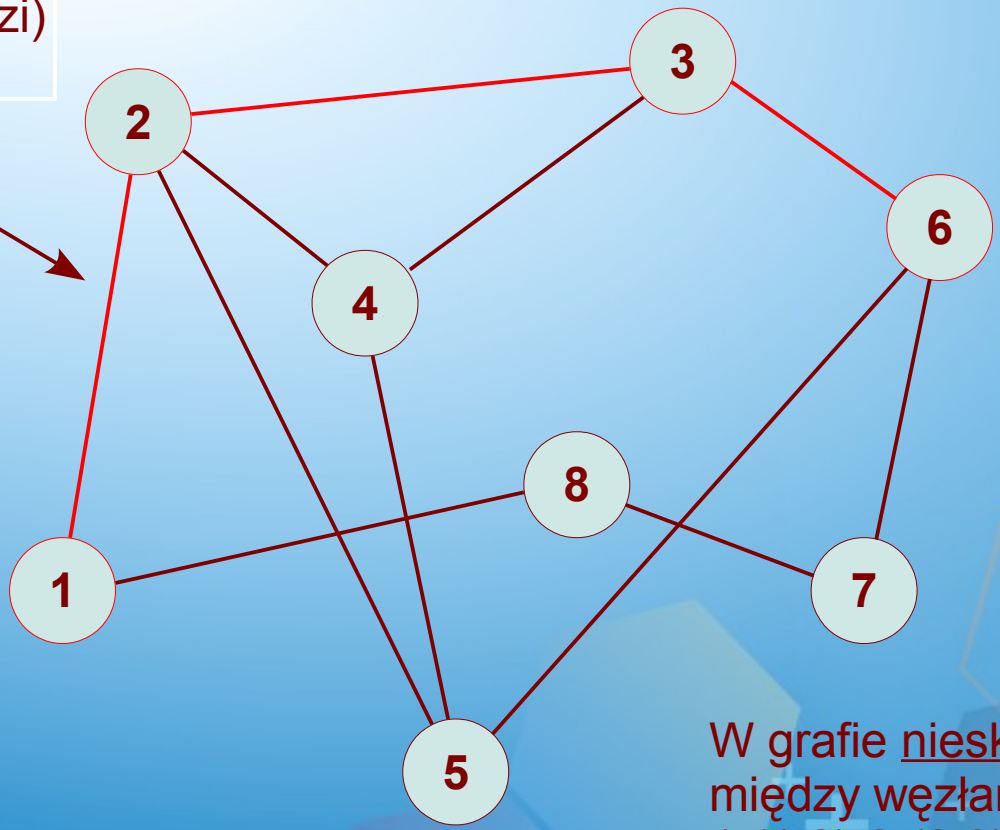
Krawędzie mogą posiadać kierunek. Taki graf jest grafem skierowanym.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Pojęcia związane z budową grafu:

Ścieżka to zbiór węzłów (czasem także krawędzi) łączących dwa węzły.

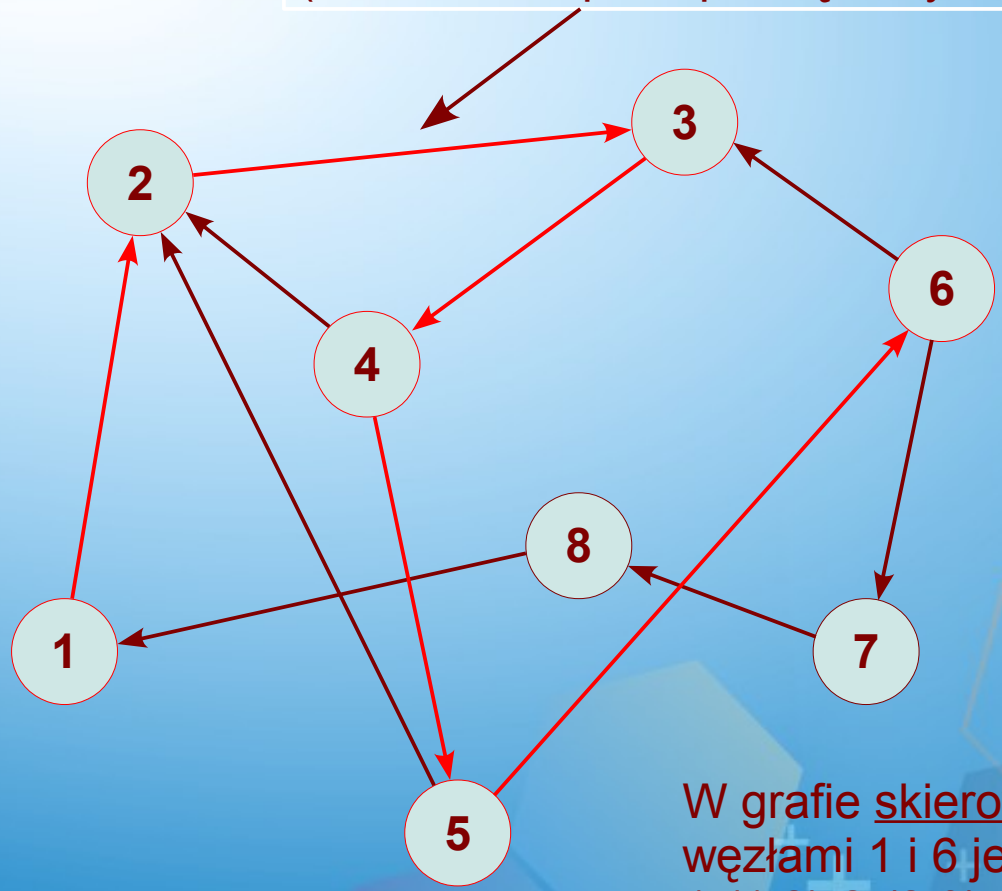


W grafie nieskierowanym ścieżka między węzłami 1 i 6 jest następująca:
1-(1,2)-2-(2,3)-3-(3,6)-6

Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Pojęcia związane z budową grafu:

Droga to ścieżka w której węzły nie powielają się (ewentualnie poza początkowym i końcowym).

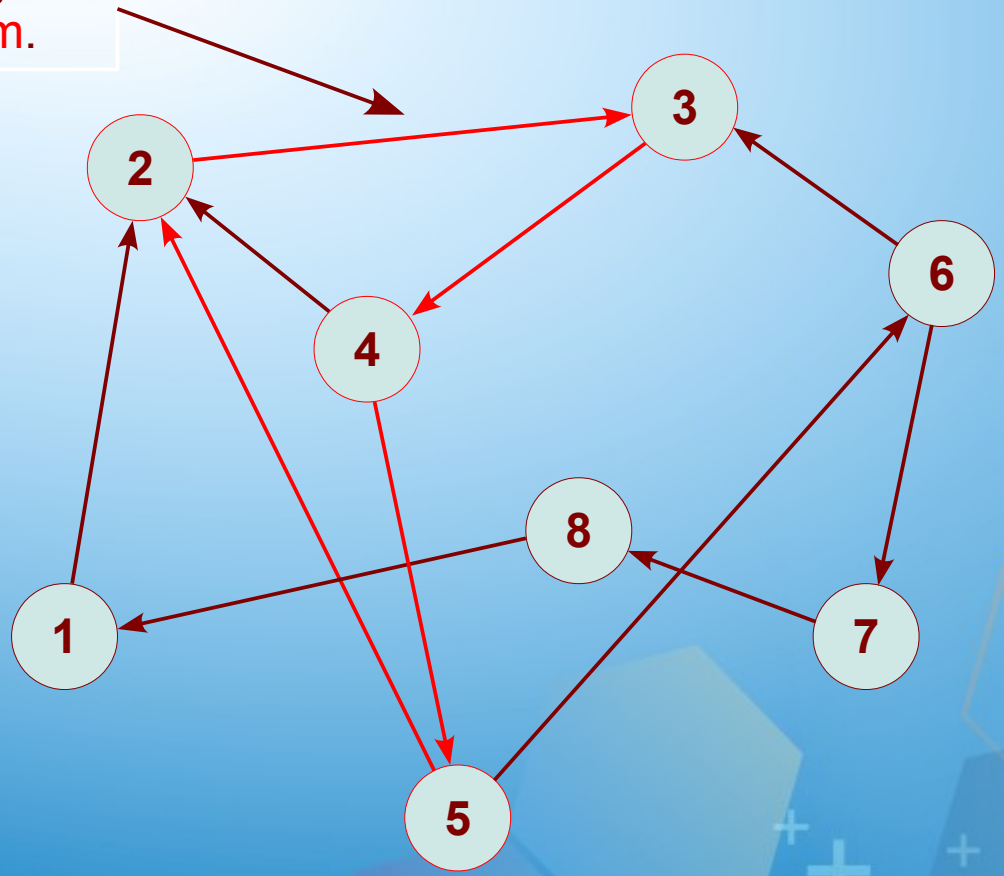


W grafie skierowanym droga między węzłami 1 i 6 jest następująca:
1-(1,2)-2-(2,3)-3-(3,4)-4-(4,5)-5-(5-6)-6

Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Pojęcia związane z budową grafu:

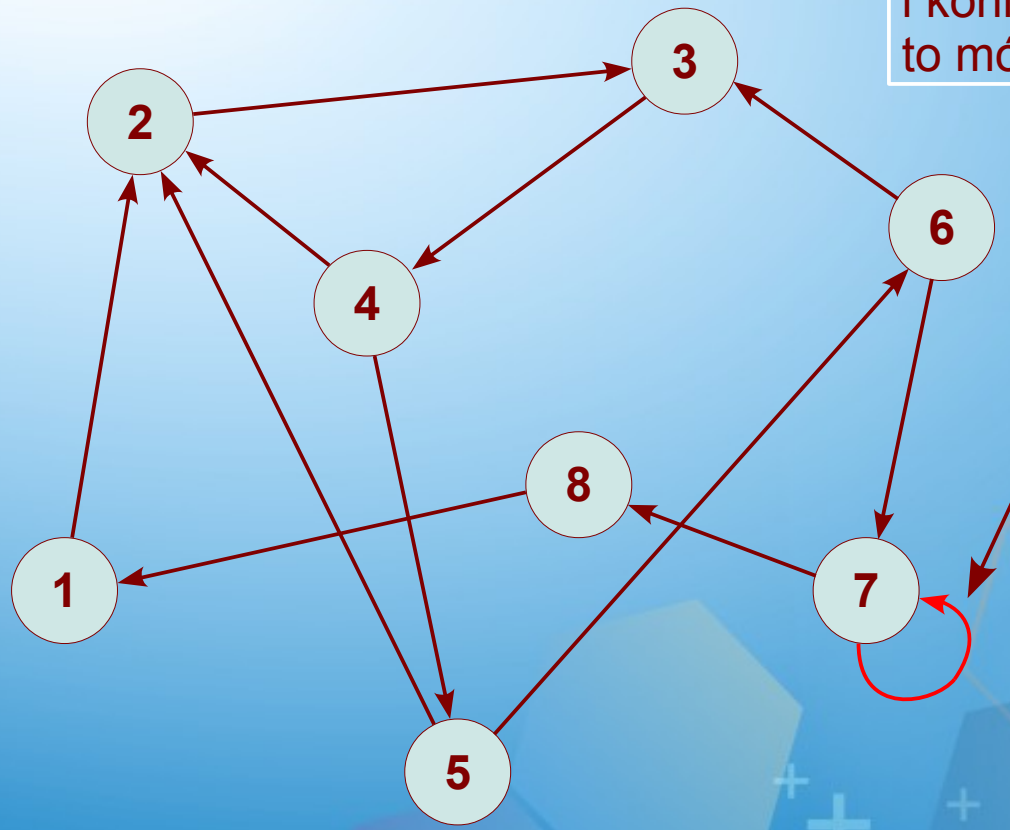
Zamknięta droga w grafie nazywana jest cyklem.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Pojęcia związane z budową grafu:

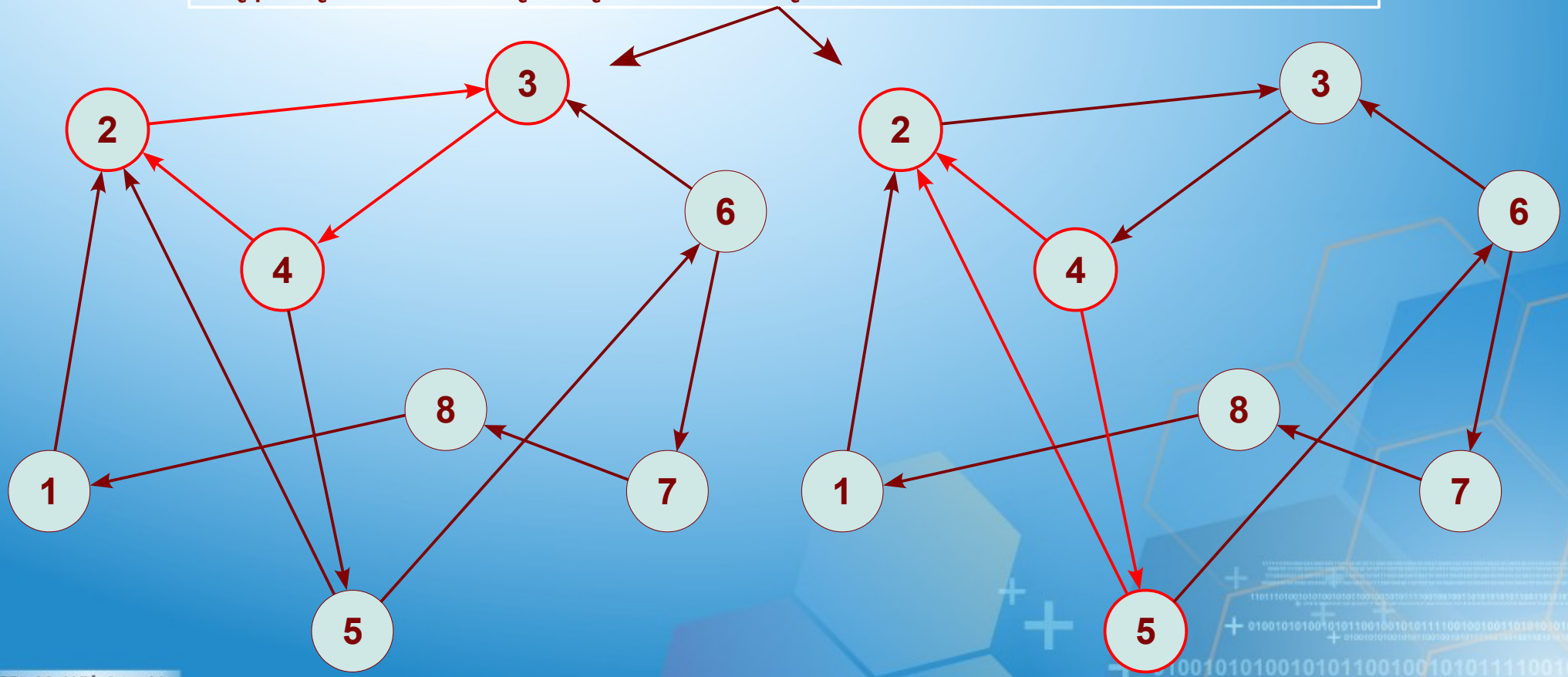
Jeśli krawędź ma początek i koniec w tym samym węźle to mówi się o pętli.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Pojęcia związane z budową grafu:

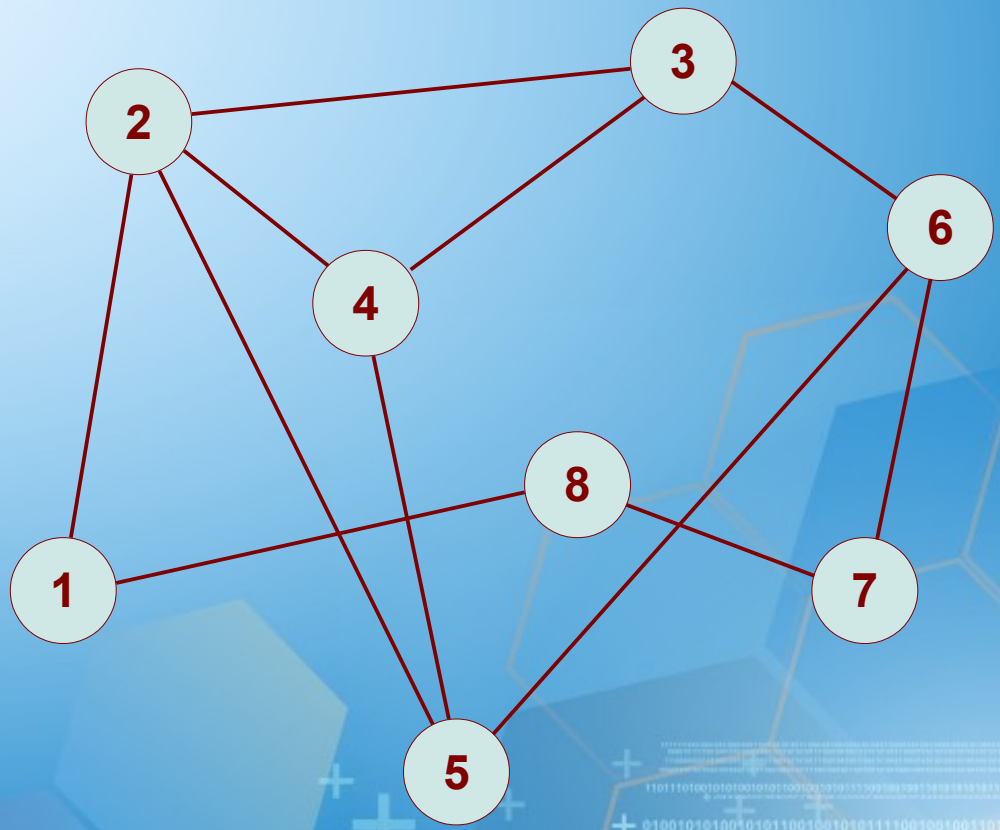
Jeśli w grafie można wskazać zbiór węzłów, w którym każde dwa węzły są połączone krawędzią, to mówi się o istnieniu **kliki**.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Drzewo rozpinające grafu:

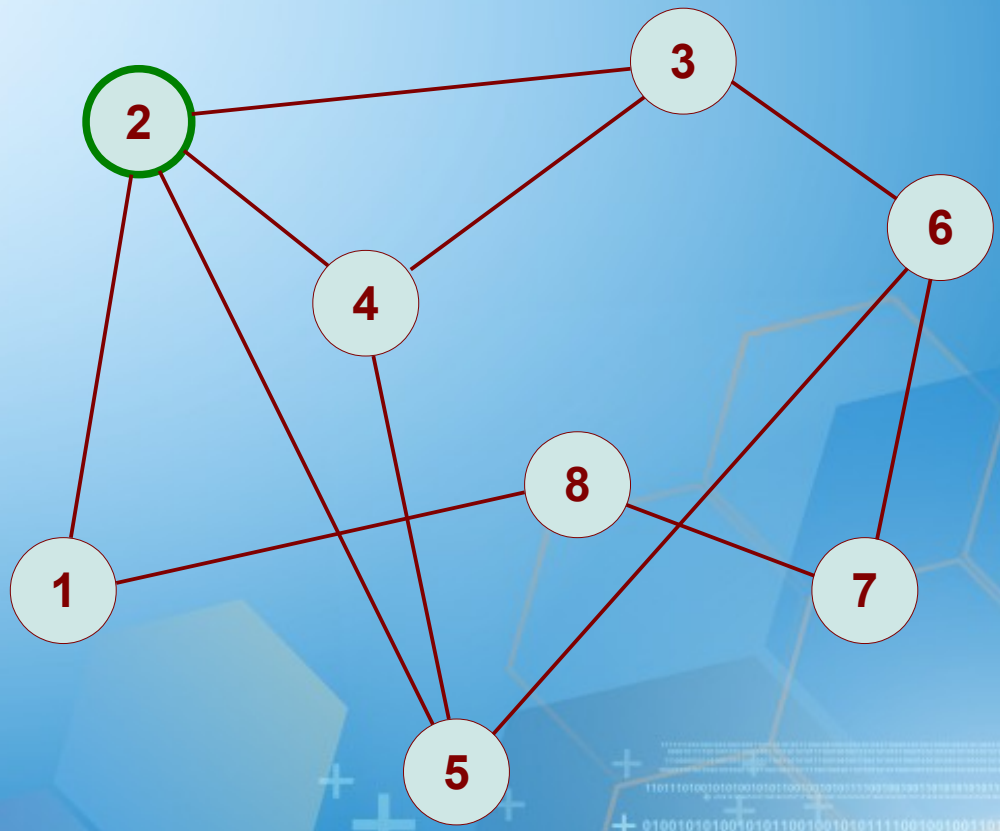
- zawiera wszystkie węzły grafu,
- zbiór krawędzi drzewa jest podzbiorem krawędzi grafu,
- powstaje przez usunięcie cykli z grafu.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Drzewo rozpinające grafu:

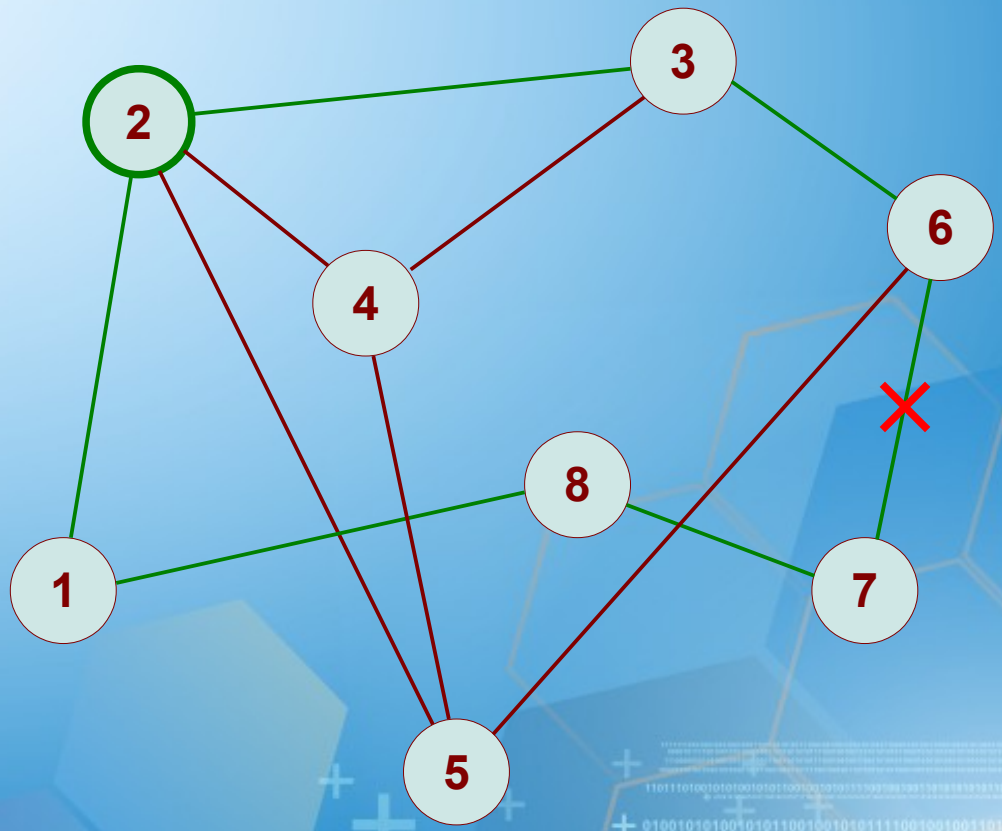
- zawiera wszystkie węzły grafu,
- zbiór krawędzi drzewa jest podzbiorem krawędzi grafu,
- powstaje przez usunięcie cykli z grafu.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Drzewo rozpinające grafu:

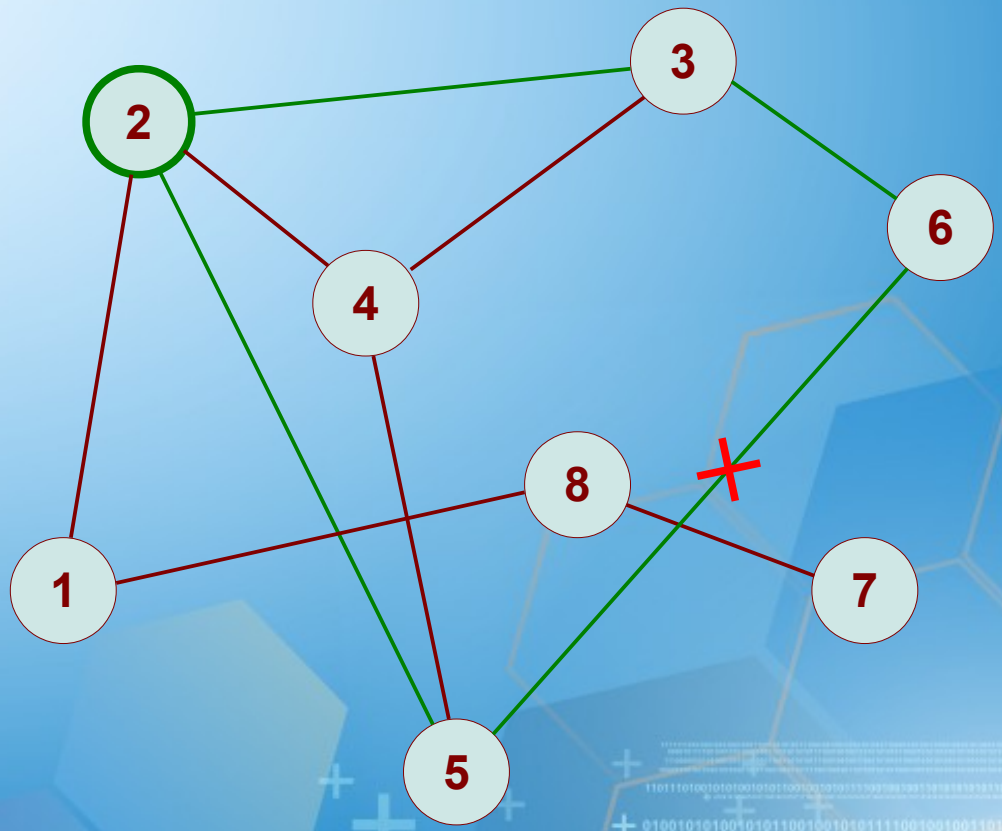
- zawiera wszystkie węzły grafu,
- zbiór krawędzi drzewa jest podzbiorem krawędzi grafu,
- powstaje przez usunięcie cykli z grafu.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Drzewo rozpinające grafu:

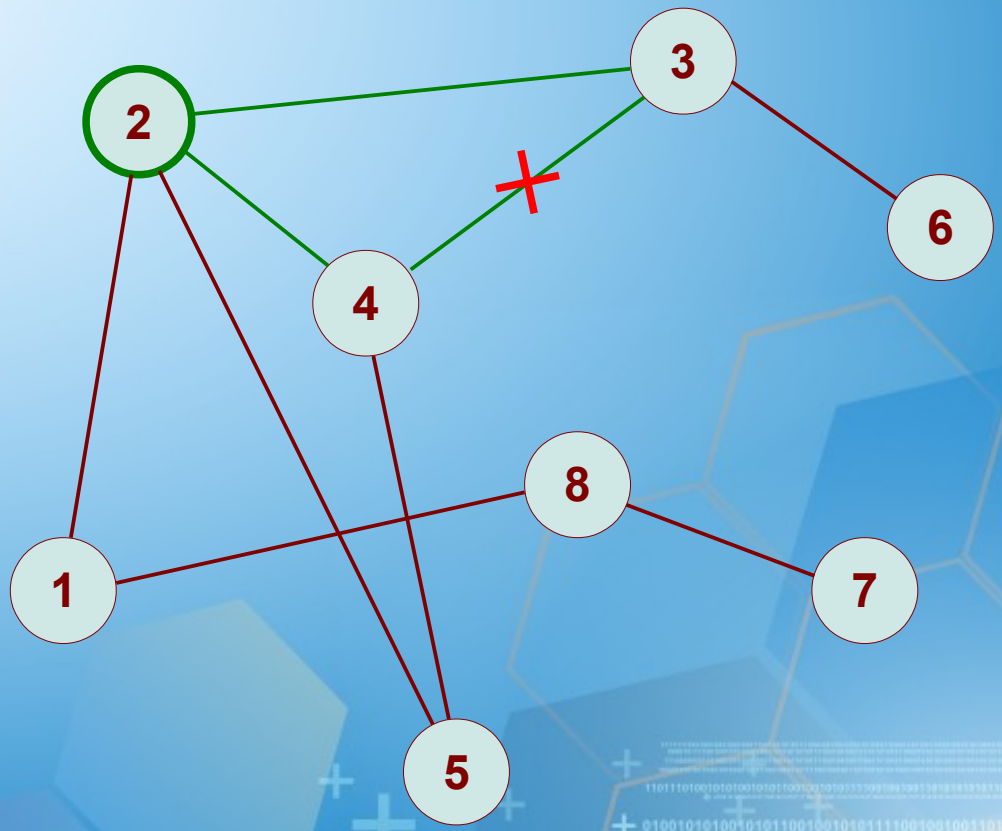
- zawiera wszystkie węzły grafu,
- zbiór krawędzi drzewa jest podzbiorem krawędzi grafu,
- powstaje przez usunięcie cykli z grafu.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Drzewo rozpinające grafu:

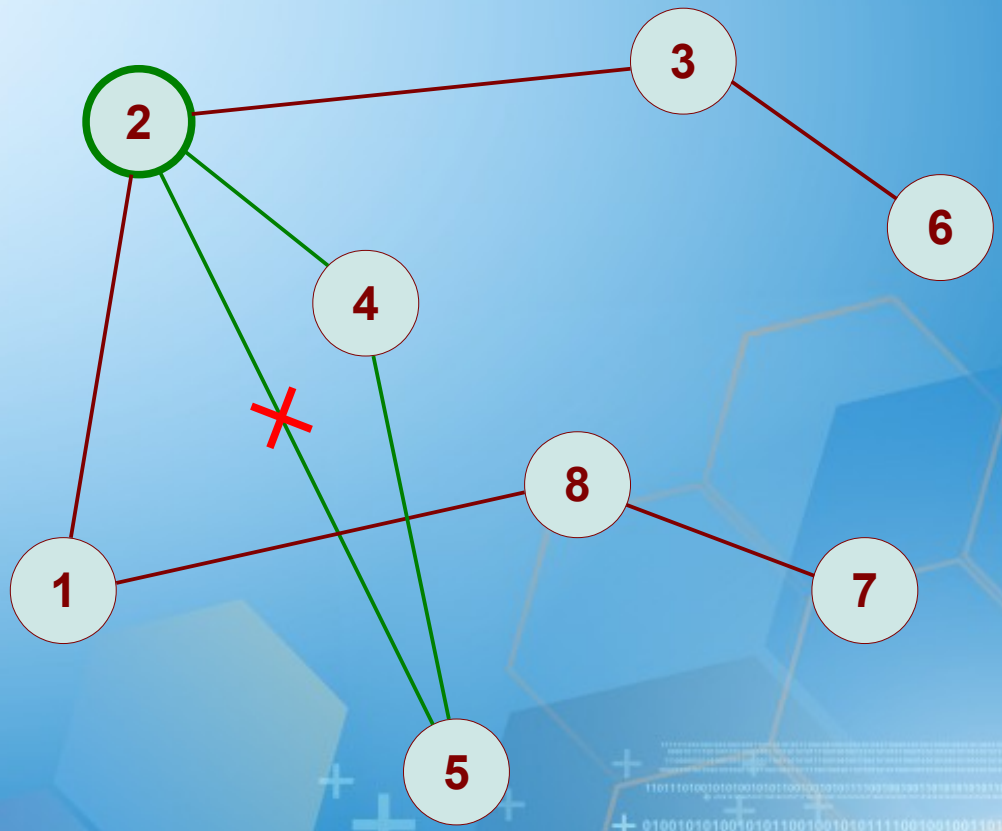
- zawiera wszystkie węzły grafu,
- zbiór krawędzi drzewa jest podzbiorem krawędzi grafu,
- powstaje przez usunięcie cykli z grafu.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

Drzewo rozpinające grafu:

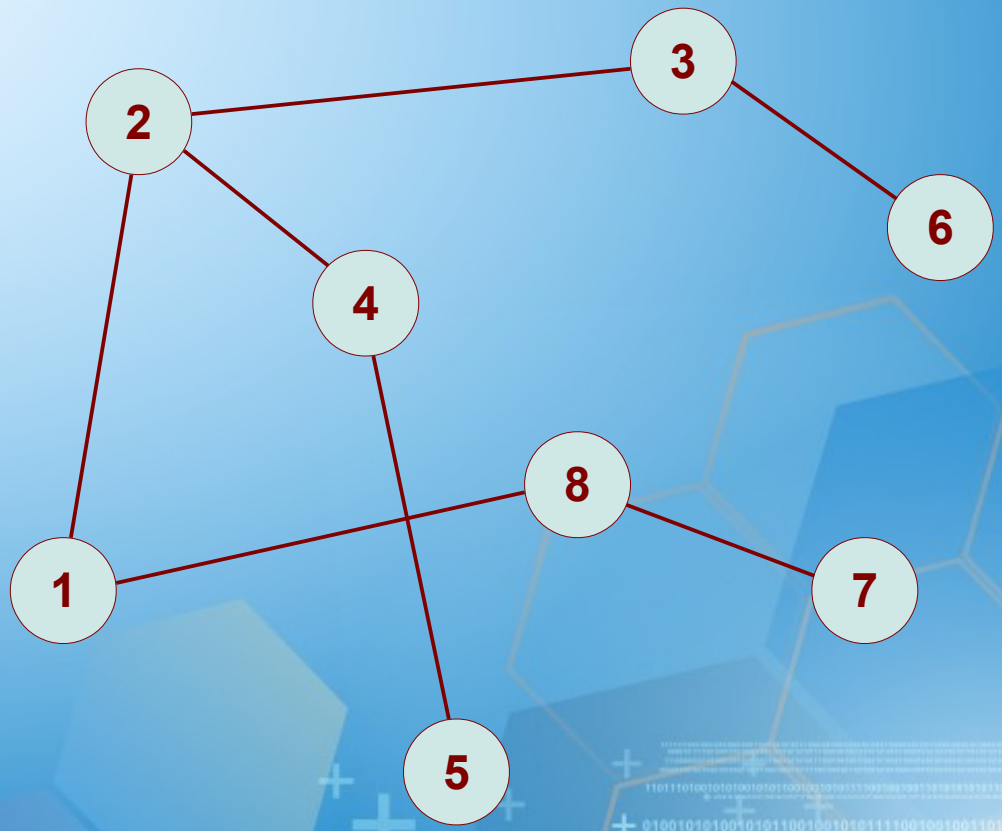
- zawiera wszystkie węzły grafu,
- zbiór krawędzi drzewa jest podzbiorem krawędzi grafu,
- powstaje przez usunięcie cykli z grafu.



Graf – to rodzaj kontenera danych, składającego się z węzłów i łączących je krawędzi w taki sposób, że każda krawędź zaczyna się i kończy w jednym z węzłów grafu.

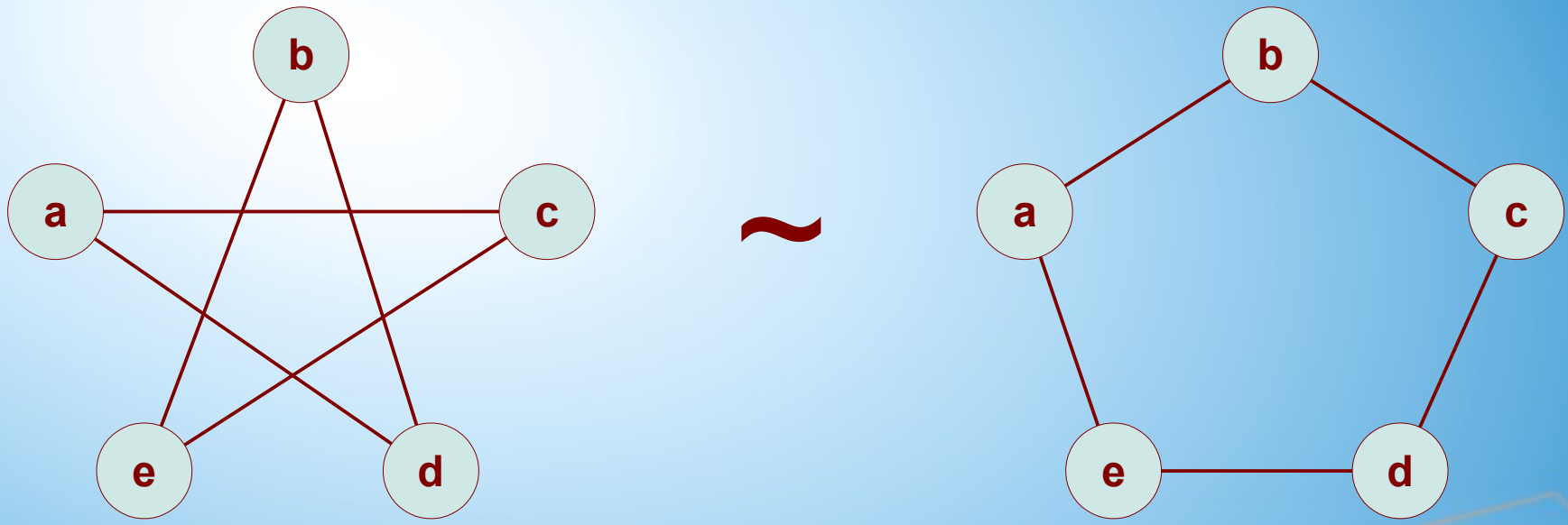
Drzewo rozpinające grafu:

- zawiera wszystkie węzły grafu,
- zbiór krawędzi drzewa jest podzbiorem krawędzi grafu,
- powstaje przez usunięcie cykli z grafu.



Klasy grafów

Grafy izomorficzne – są to grafy dla których istnieje funkcja przekształcająca zbiór węzłów jednego grafu w drugi, zachowująca strukturę grafu.

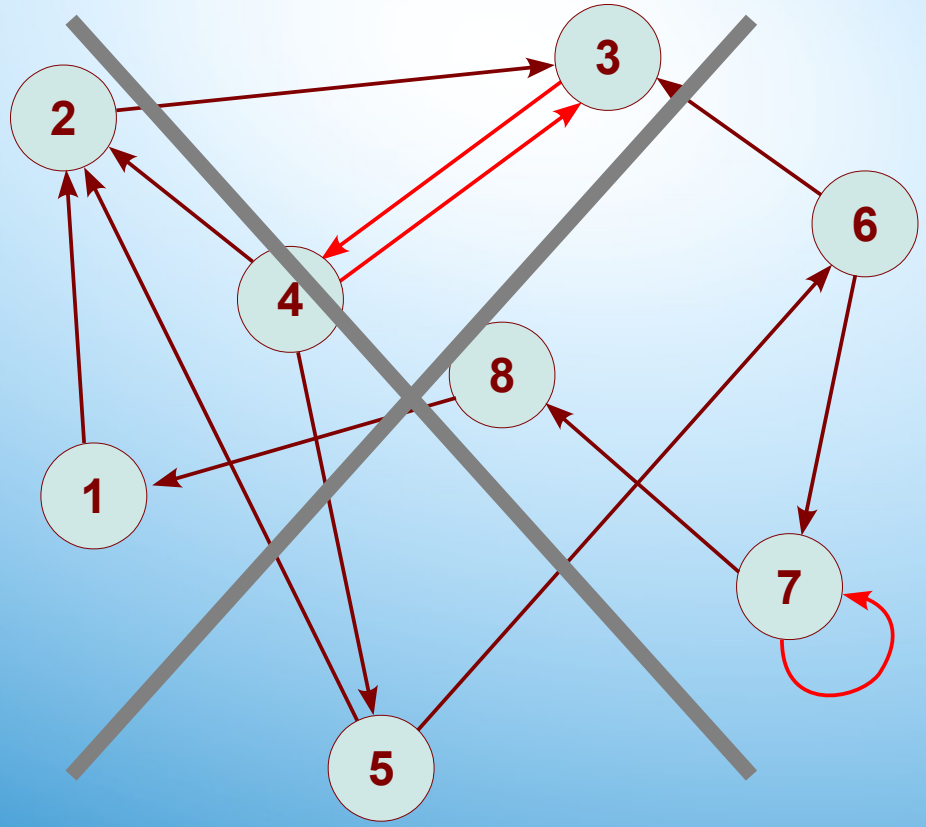


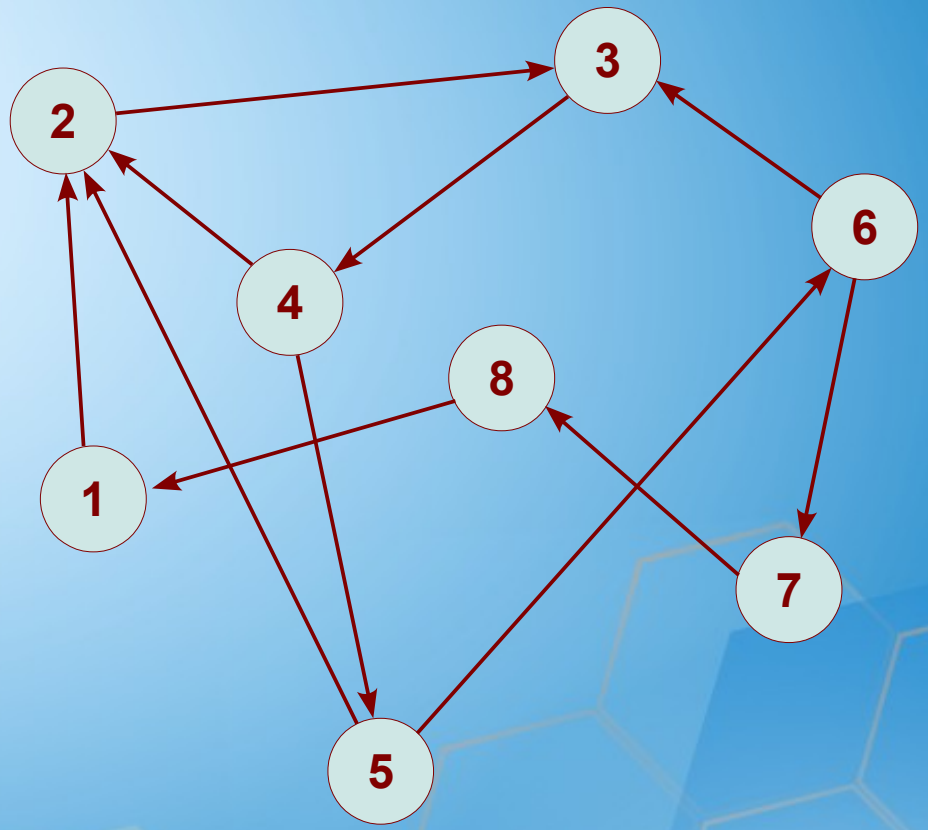
Funkcja przekształcająca: f(a) = a, f(b) = d, f(c) = b, f(d) = e, f(e) = c.

Uwagi:

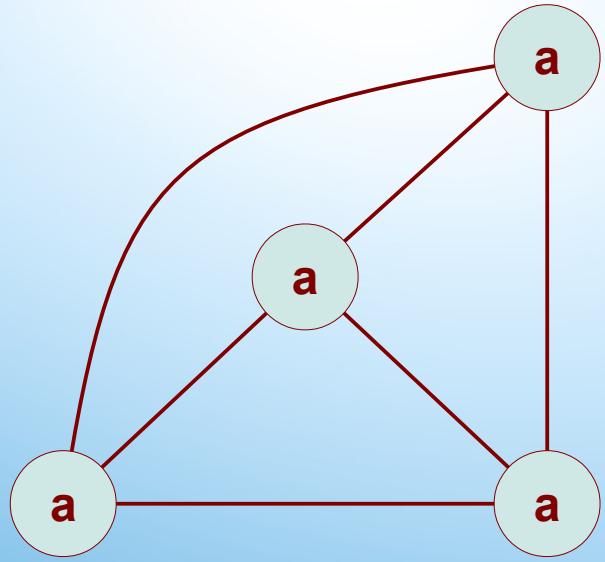
- problem rozstrzygania czy grafy są izomorficzne należy do klasy problemów NP,
- uogólnienie na problem rozstrzygający o izomorfizmie podgrafu jest problemem NP-zupełnym.

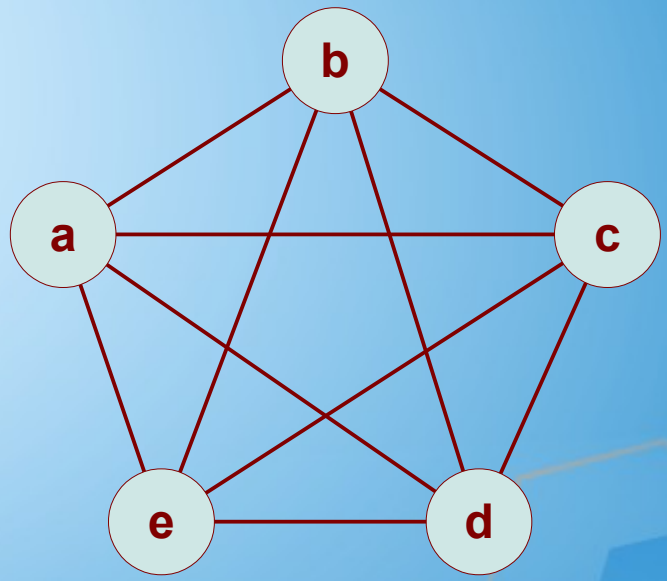
Graf prosty – jest to graf który nie zawiera pętli ani krawędzi wielokrotnych.



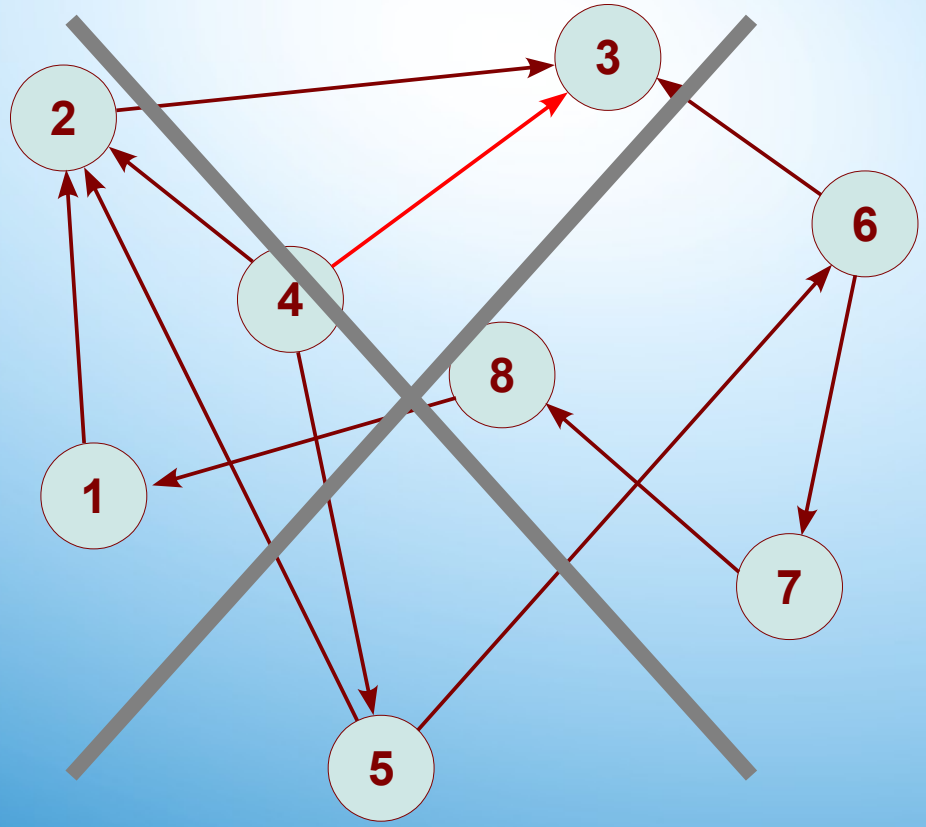


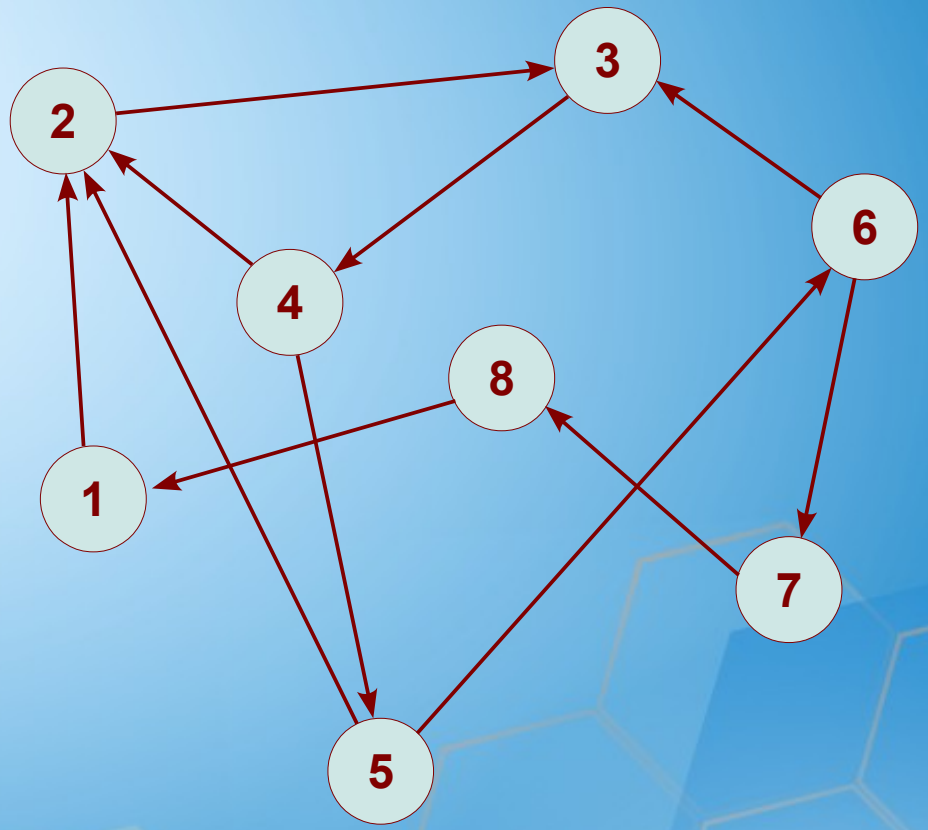
Graf pełny – jest to graf w którym każdy węzeł jest bezpośrednio połączony krawędzią z każdym innym węzłem.



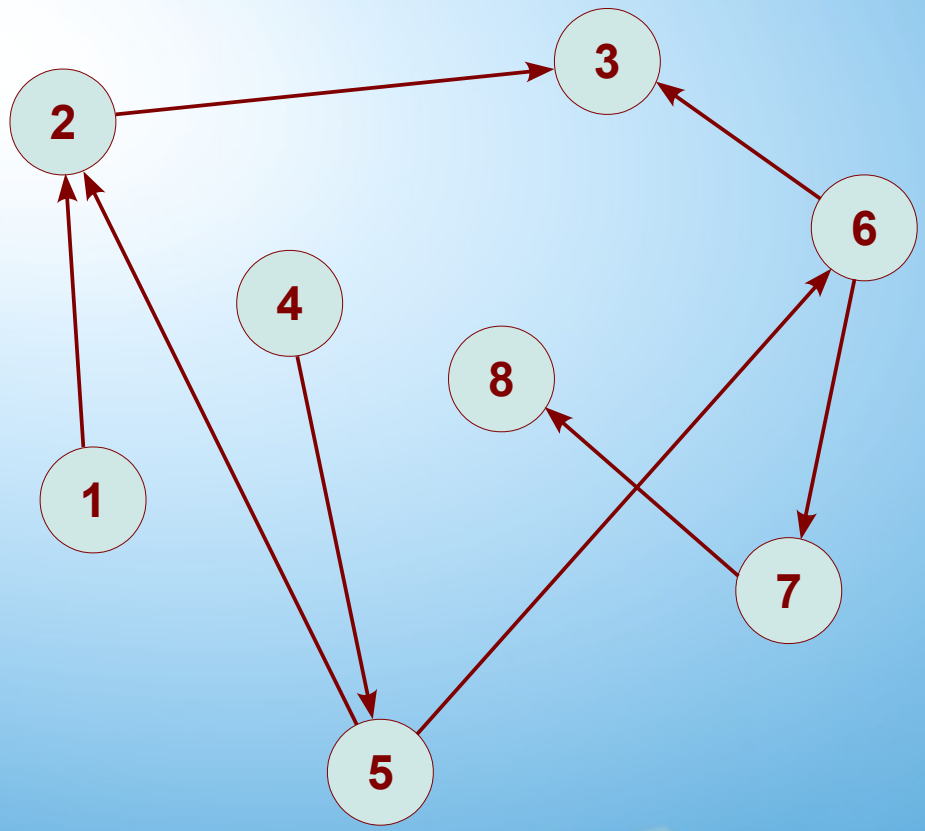


Graf spójny – jest to graf w którym dla każdego węzła istnieje droga do dowolnego innego węzła.

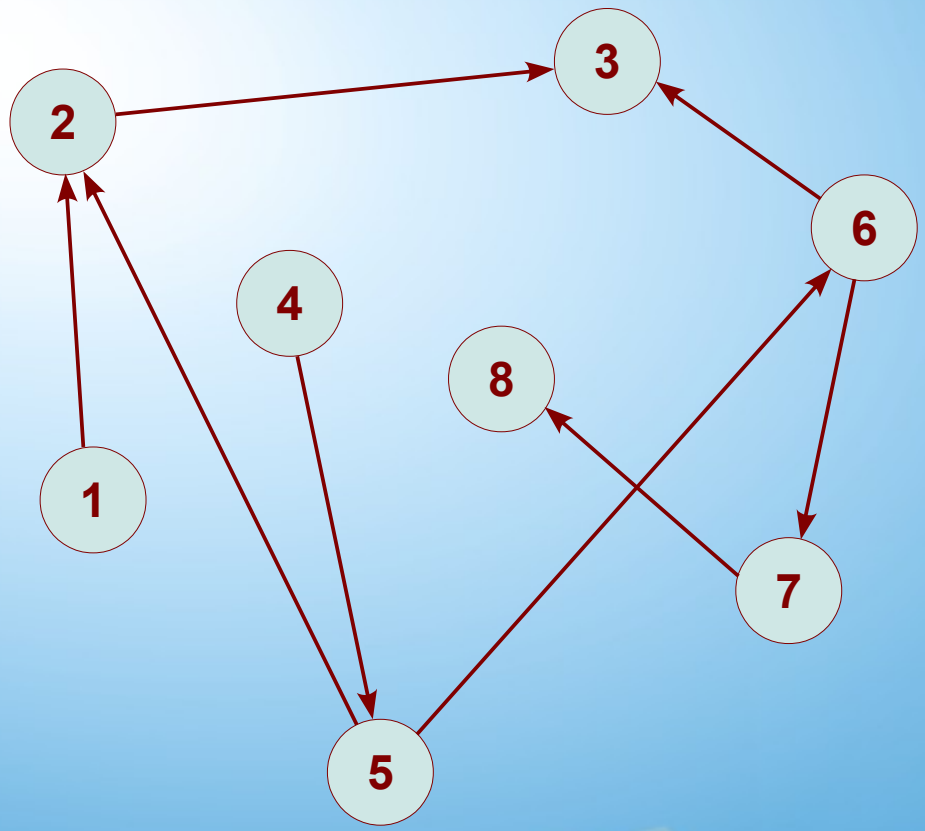




Graf acykliczny – jest to graf w którym nie istnieje droga zamknięta.

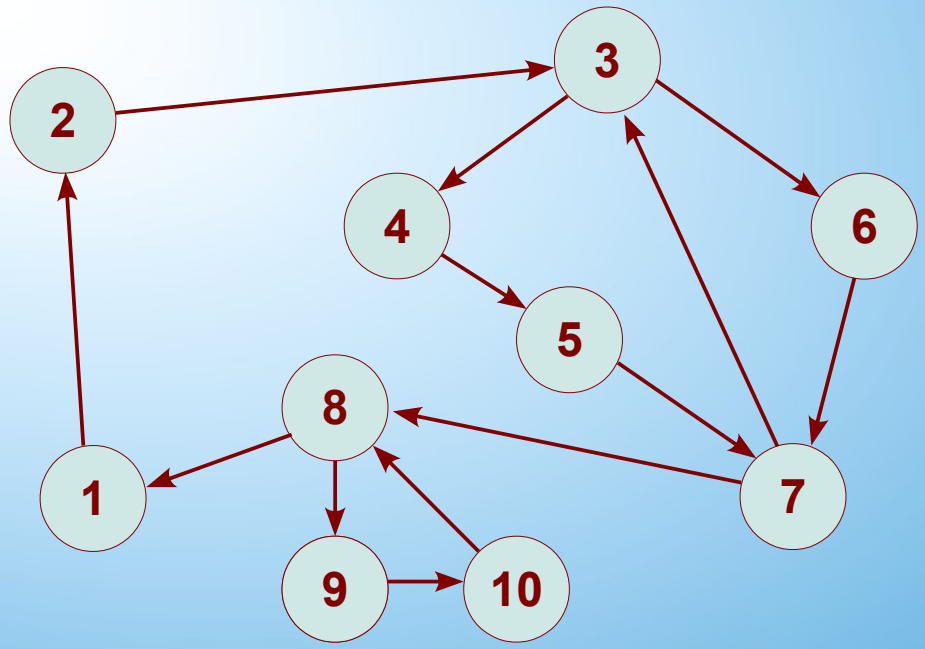


Graf acykliczny – jest to graf w którym nie istnieje droga zamknięta.

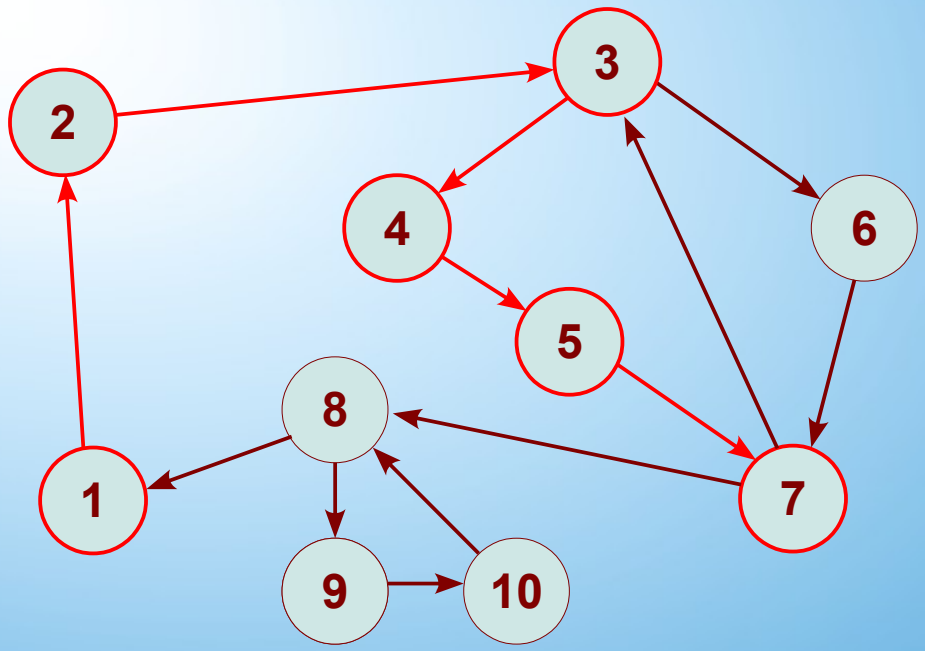


Drzewo jest grafem spójnym, acyklicznym.

Graf eulerowski – jest to graf w którym można wyznaczyć cykl, w którym przez każdą krawędź cyklu przechodzi się dokładnie jeden raz.

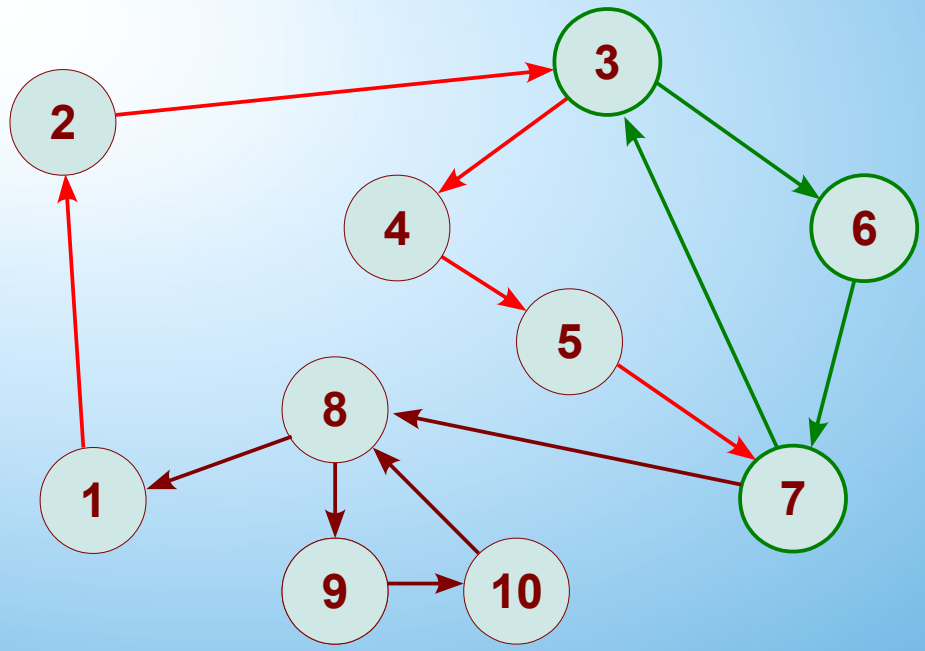


Graf eulerowski – jest to graf w którym można wyznaczyć cykl, w którym przez każdą krawędź cyklu przechodzi się dokładnie jeden raz.



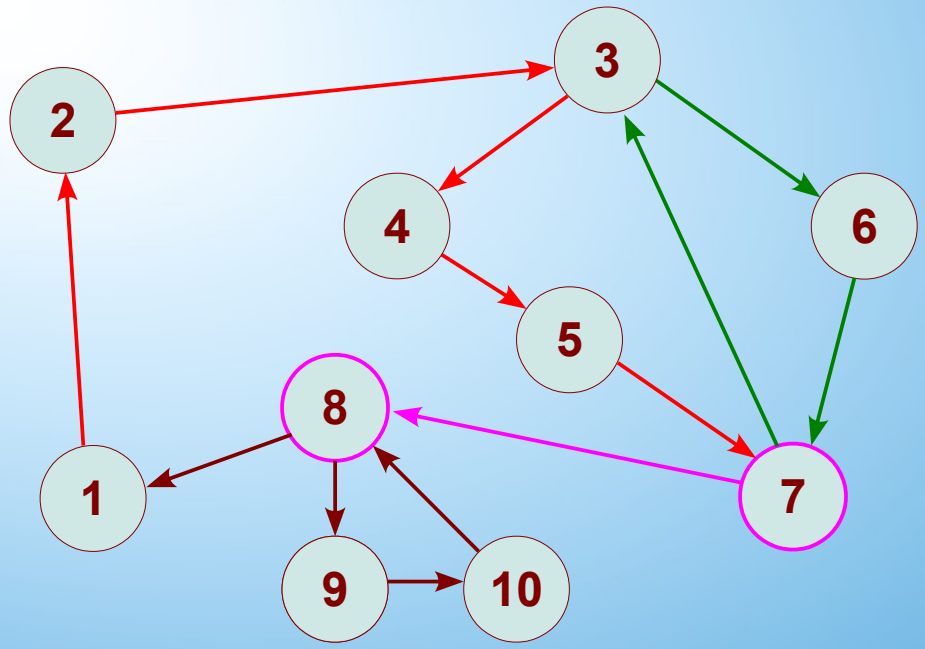
Cykl Eulera: 1-(1,2)-2-(2,3)-3-(3,4)-4-(4,5)-5-(5,7)-7...

Graf eulerowski – jest to graf w którym można wyznaczyć cykl, w którym przez każdą krawędź cyklu przechodzi się dokładnie jeden raz.



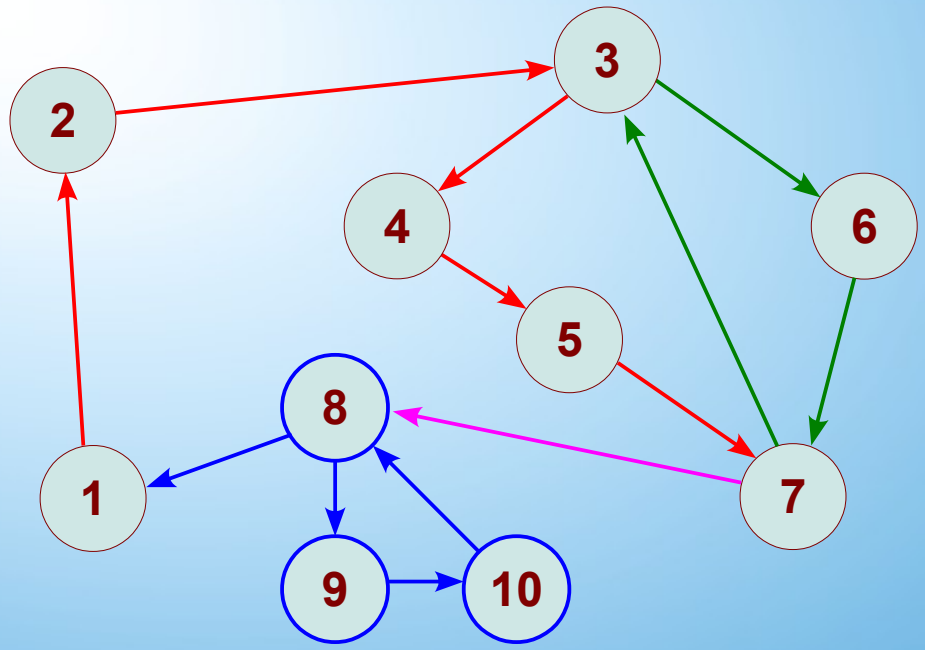
Cykl Eulera: 1-(1,2)-2-(2,3)-3-(3,4)-4-(4,5)-5-(5,7)-7-(7,3)-3-(3,6)-6-(6,7)-7...

Graf eulerowski – jest to graf w którym można wyznaczyć cykl, w którym przez każdą krawędź cyklu przechodzi się dokładnie jeden raz.



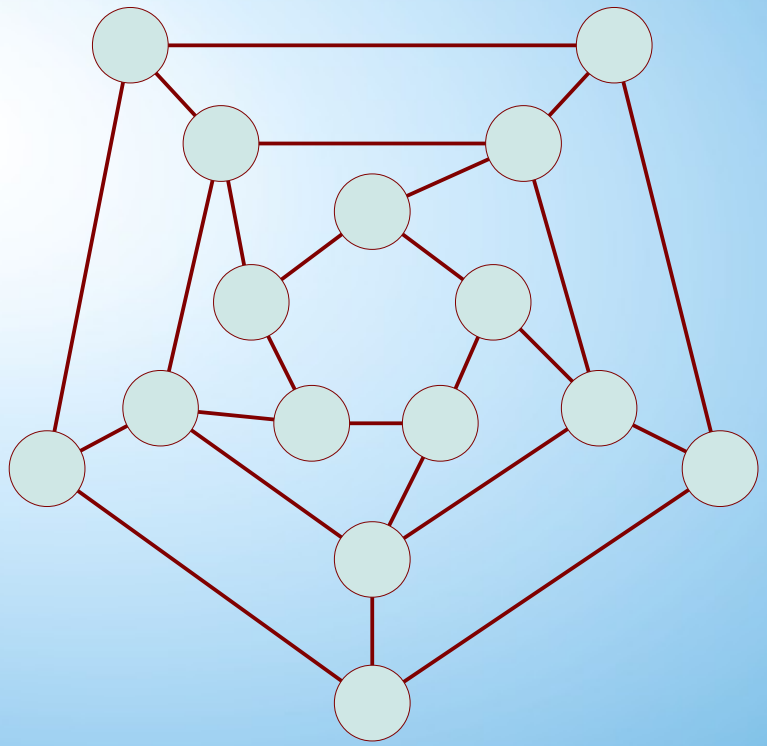
Cykl Eulera: 1-(1,2)-2-(2,3)-3-(3,4)-4-(4,5)-5-(5,7)-7-(7,3)-3-(3,6)-6-(6,7)-7-(7,8)-8..

Graf eulerowski – jest to graf w którym można wyznaczyć cykl, w którym przez każdą krawędź cyklu przechodzi się dokładnie jeden raz.

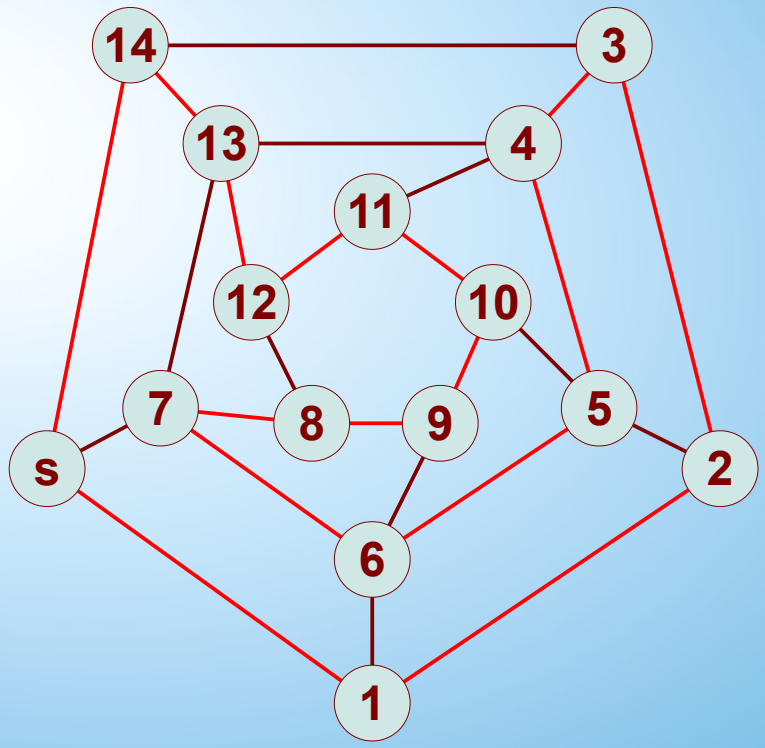


Cykl Eulera: 1-(1,2)-2-(2,3)-3-(3,4)-4-(4,5)-5-(5,7)-7-(7,3)-3-(3,6)-6-(6,7)-7-(7,8)-8-
-(8,9)-9-(9,10)-10-(10,8)-8-(8,1)-1

Graf hamiltonowski – jest to graf zawierający ścieżkę w której przez każdy z węzłów przechodzi się dokładnie jeden raz.

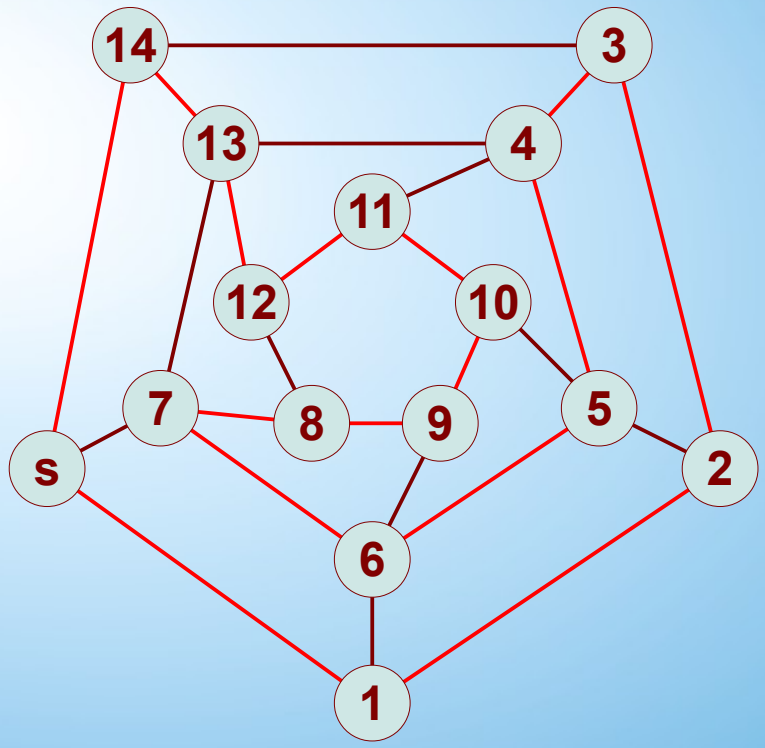


Graf hamiltonowski – jest to graf zawierający ścieżkę w której przez każdy z węzłów przechodzi się dokładnie jeden raz.



Cykl Hamiltona: s-1-2-3-4-5-6-7-8-9-10-11-12-13-14-s

Graf hamiltonowski – jest to graf zawierający ścieżkę w której przez każdy z węzłów przechodzi się dokładnie jeden raz.

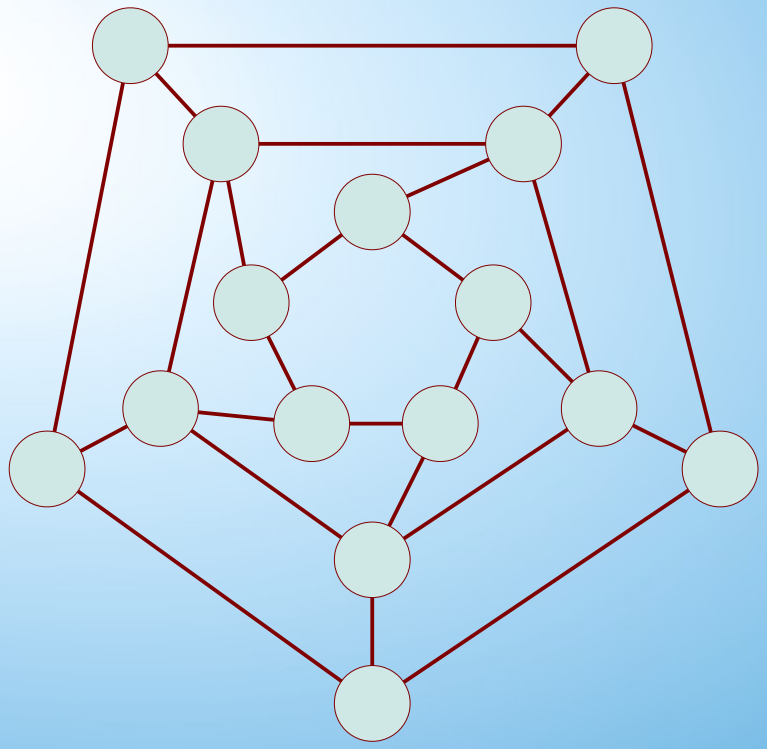


Cykl Hamiltona: s-1-2-3-4-5-6-7-8-9-10-11-12-13-14-s

Uwagi:

- graf pełny i opisujący wielościan foremny są grafami hamiltonowskimi,
- graf niespójny nie jest hamiltonowski,
- znalezienie cyklu Hamiltona jest równoważne rozwiązaniu problemu komiwojażera.

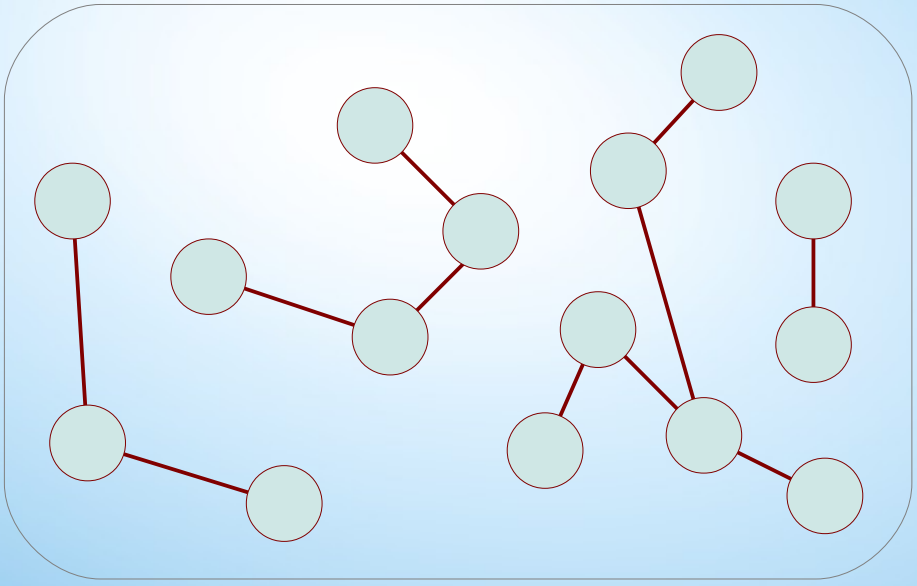
Graf planarny – jest to graf, który można przedstawić na płaszczyźnie w taki sposób, że żadna krawędź nie przecina się z inną krawędzią.



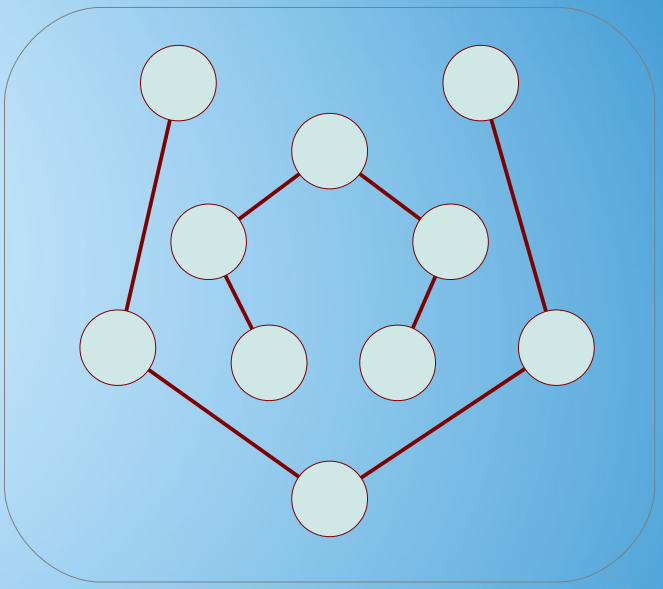
Uwagi:

- problem rozstrzygnięcia czy graf jest planarny występuje podczas projektowania topologi układów scalonych,
- graf jest planarny jeśli do jego pokolorowania nie potrzeba więcej niż czterech kolorów.

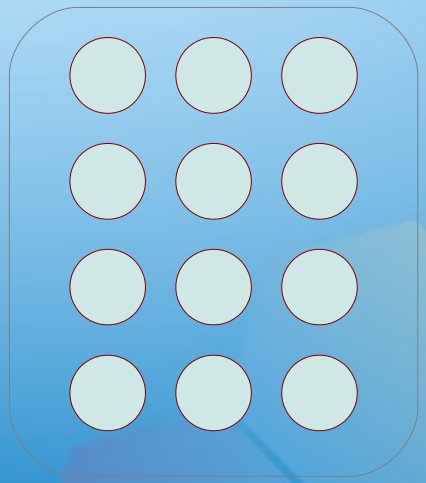
Las – jest to graf, którego wszystkie spójne składowe (podgrafy wydzielone z grafu bez usuwania krawędzi) są drzewami.



Graf 1 – las 4 drzew



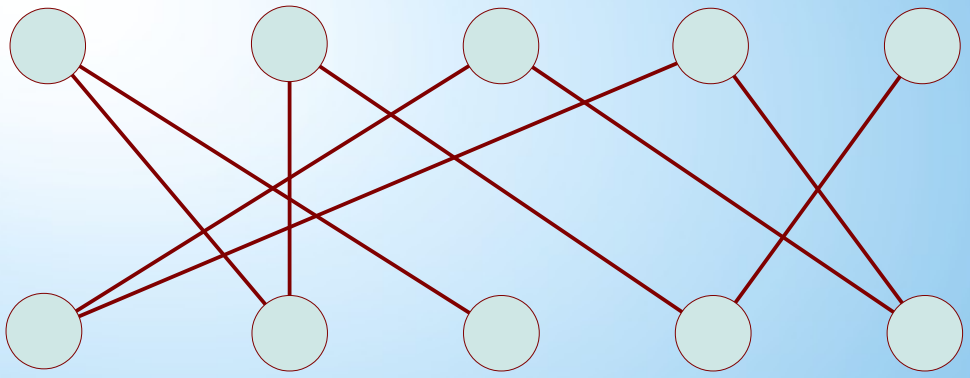
Graf 2 – las 2 drzew



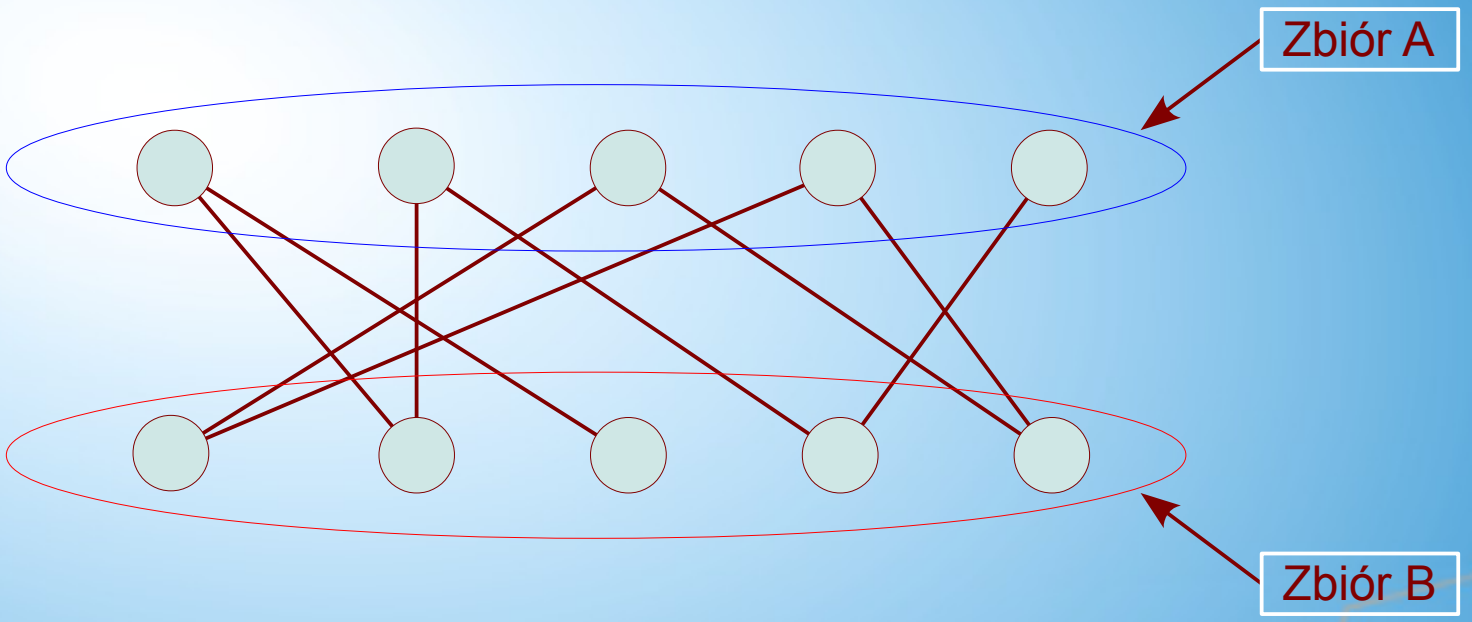
Graf 3 – las 12 drzew

Graf dwudzielny – jest to graf, w którym węzły można podzielić na dwa rozłączne zbiory, przy czym każda krawędź łączy węzły z różnych zbiorów.

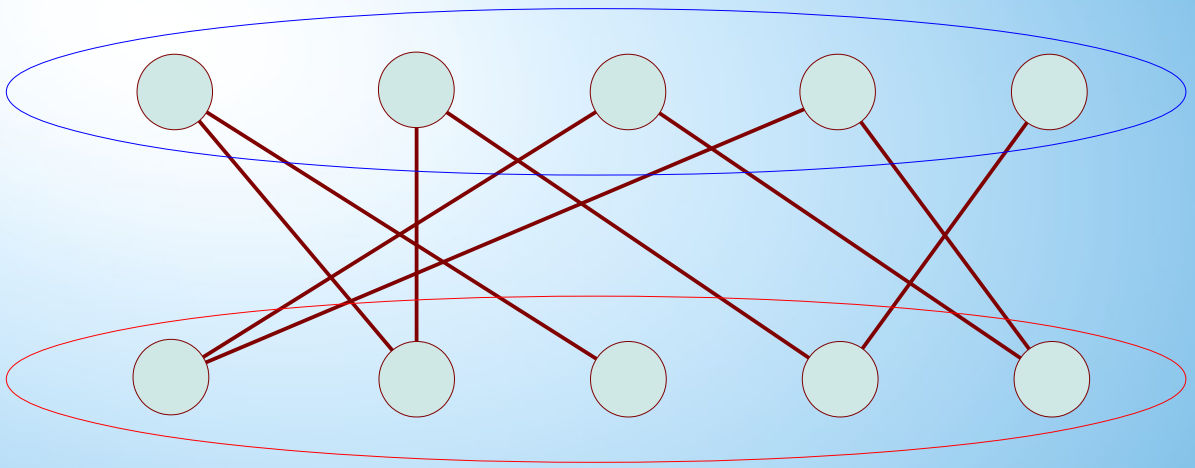
Graf dwudzielny – jest to graf, w którym węzły można podzielić na dwa rozłączne zbiory, przy czym każda krawędź łączy węzły z różnych zbiorów.



Graf dwudzielny – jest to graf, w którym węzły można podzielić na dwa rozłączne zbiory, przy czym każda krawędź łączy węzły z różnych zbiorów.

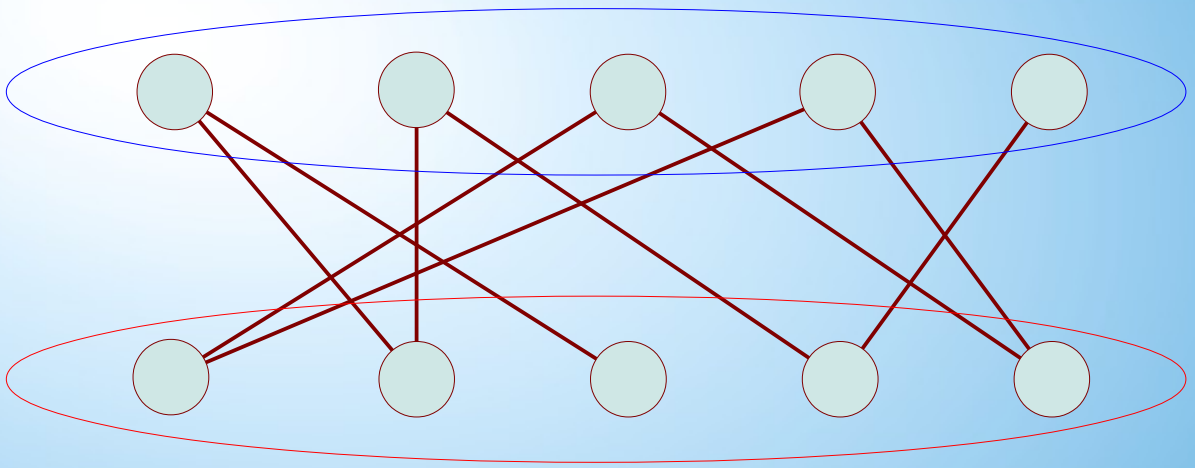


Graf dwudzielny – jest to graf, w którym węzły można podzielić na dwa rozłączne zbiory, przy czym każda krawędź łączy węzły z różnych zbiorów.



Graf stosowany jest do rozwiązywania problemów skojarzeń par obiektów.

Graf dwudzielny – jest to graf, w którym węzły można podzielić na dwa rozłączne zbiory, przy czym każda krawędź łączy węzły z różnych zbiorów.



Graf stosowany jest do rozwiązywania problemów skojarzeń par obiektów.

Zadanie:
Jaką maksymalną liczbę wież można postawić na szachownicy, aby nie mogły się atakować ?

Rozwiązanie:
Wyznaczenie maksymalnego skojarzenia w grafie dwudzielnym, gdzie jeden zbiór węzłów to wiersze szachownicy, a drugi zbiór węzłów to jej kolumny.

Reprezentacja grafów w pamięci komputera

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

A[]	1	2	3	4	5	...
1	0	1	0	0	0	...
2	1	0	1	1	1	...
3	0	1	0	1	0	...
4	0	1	1	0	1	...
5	0	1	0	1	0	...
...	...	...	...	...	...	...

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

A[]	1	2	3	4	5	...
1	0	1	0	0	0	...
2	1	0	1	1	1	...
3	0	1	0	1	0	...
4	0	1	1	0	1	...
5	0	1	0	1	0	...
...	...	...	...	...	...	...

Kolumny i wiersze tabeli A odpowiadają węzłom grafu.

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

A[]	1	2	3	4	5	...
1	0	1	0	0	0	...
2	1	0	1	1	1	...
3	0	1	0	1	0	...
4	0	1	1	0	1	...
5	0	1	0	1	0	...
...	...	...	...	...	...	...

Kolumny i wiersze tabeli A odpowiadają węzłom grafu.

Zawartość tabeli określa zależności między węzłami.

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

A[]	1	2	3	4	5	...
1	0	1	0	0	0	...
2	1	0	1	1	1	...
3	0	1	0	1	0	...
4	0	1	1	0	1	...
5	0	1	0	1	0	...
...	...	...	...	...	...	...

Kolumny i wiersze tabeli A odpowiadają węzłom grafu.

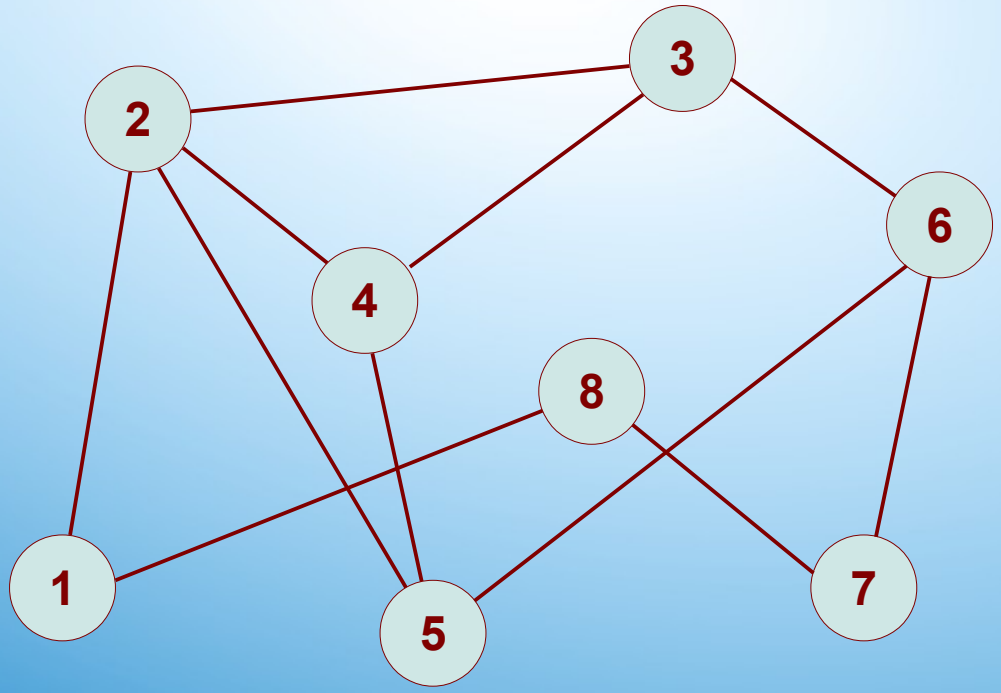
Zawartość tabeli określa zależności między węzłami.

Uwagi:

- złożoność pamięciowa wynosi $O(n^2)$,
- złożoność dla operacji sprawdzenia, czy dane węzły są połączone wynosi $O(1)$.

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

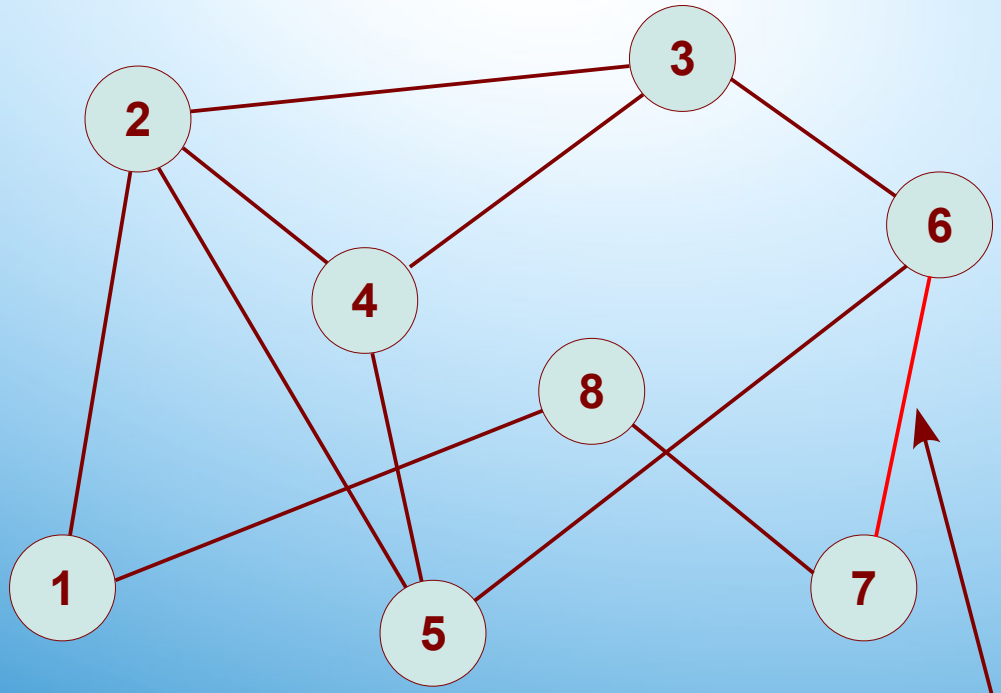
Graf nieskierowany:



A[]	1	2	3	4	5	6	7	8
1	0	1	0	0	0	0	0	1
2	1	0	1	1	1	0	0	0
3	0	1	0	1	0	1	0	0
4	0	1	1	0	1	0	0	0
5	0	1	0	1	0	1	0	0
6	0	0	1	0	1	0	1	0
7	0	0	0	0	0	1	0	1
8	1	0	0	0	0	0	1	0

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

Graf nieskierowany:

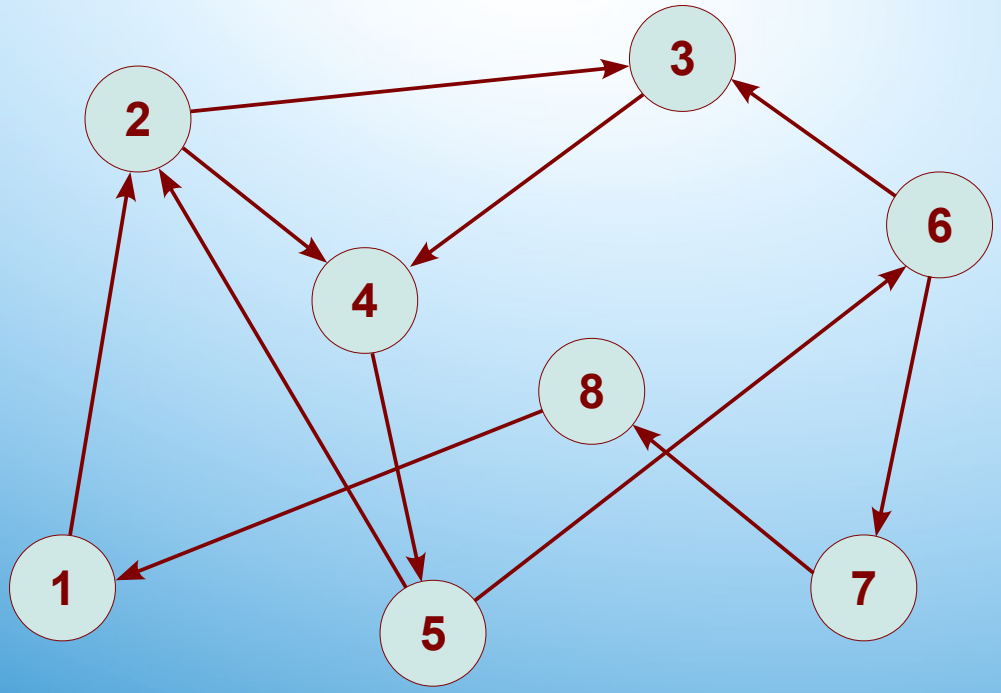


A[]	1	2	3	4	5	6	7	8
1	0	1	0	0	0	0	0	1
2	1	0	1	1	1	0	0	0
3	0	1	0	1	0	1	0	0
4	0	1	1	0	1	0	0	0
5	0	1	0	1	0	1	0	0
6	0	0	1	0	1	0	1	0
7	0	0	0	0	0	1	0	1
8	1	0	0	0	0	0	1	0

Połączenie węzła 6 z węzłem 7 jest zaznaczane w tablicy sąsiedztwa we właściwych pozycjach.

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

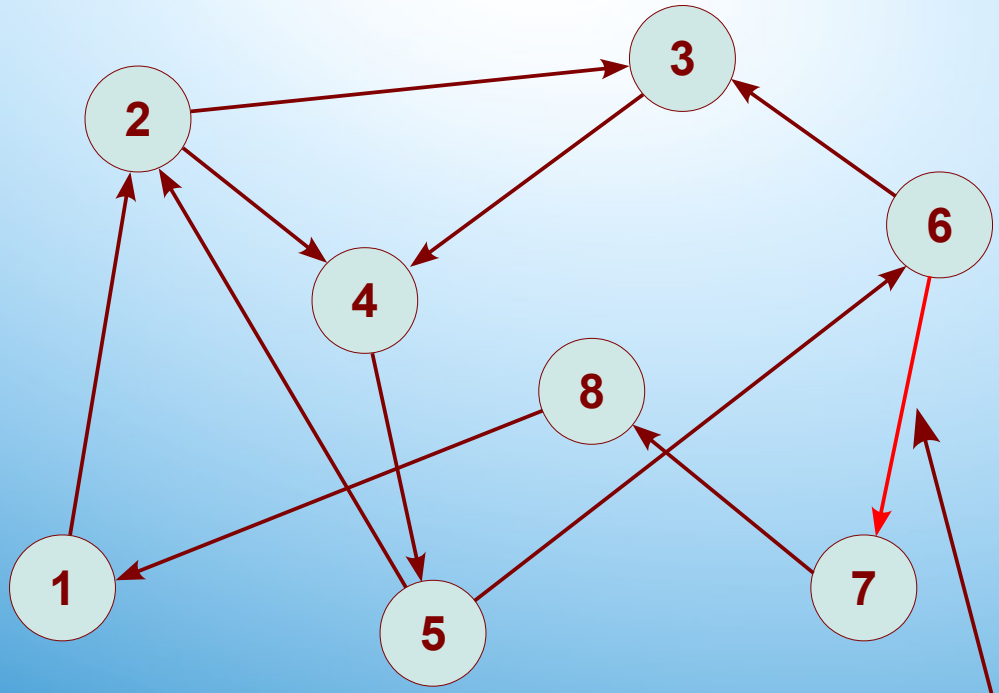
Graf skierowany:



A[]	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	1
2	1	0	0	0	1	0	0	0
3	0	1	0	0	0	1	0	0
4	0	1	1	0	0	0	0	0
5	0	0	0	1	0	0	0	0
6	0	0	0	0	1	0	0	0
7	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	1	0

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

Graf skierowany:

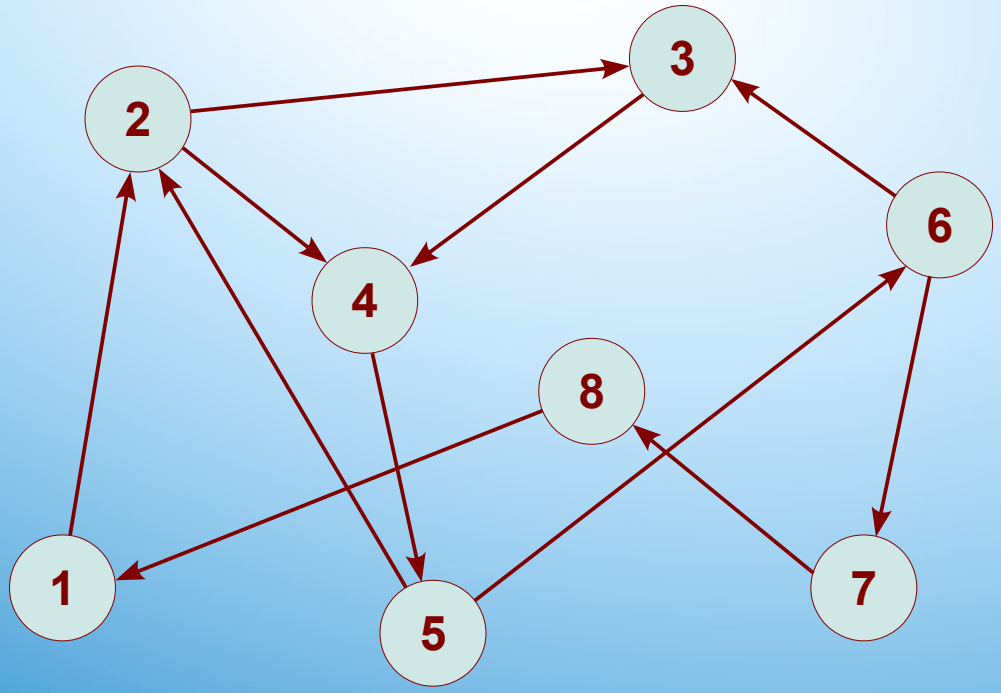


A[]	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	1
2	1	0	0	0	1	0	0	0
3	0	1	0	0	0	1	0	0
4	0	1	1	0	0	0	0	0
5	0	0	0	1	0	0	0	0
6	0	0	0	0	1	0	0	0
7	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	1	0

Istniejące połączenie węzła 6 z węzłem 7 jest zaznaczane w tablicy sąsiedztwa

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

Graf skierowany:

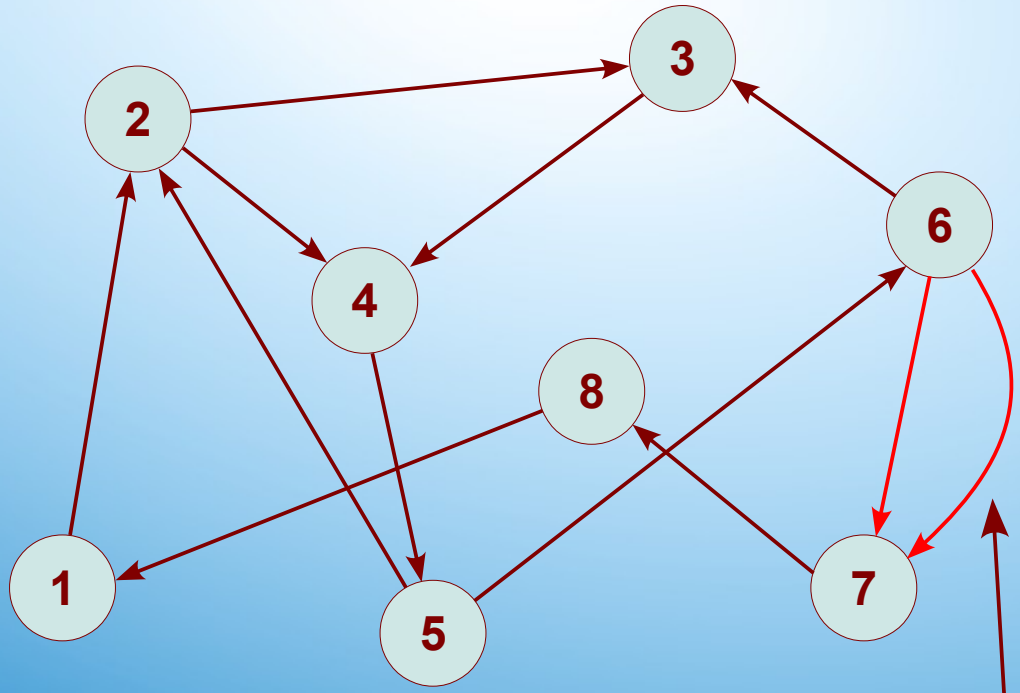


A[]	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	1
2	1	0	0	0	1	0	0	0
3	0	1	0	0	0	1	0	0
4	0	1	1	0	0	0	0	0
5	0	0	0	1	0	0	0	0
6	0	0	0	0	1	0	0	0
7	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	1	0

Połączenie węzła 7 z węzłem 6 nie istnieje.

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

Graf skierowany:

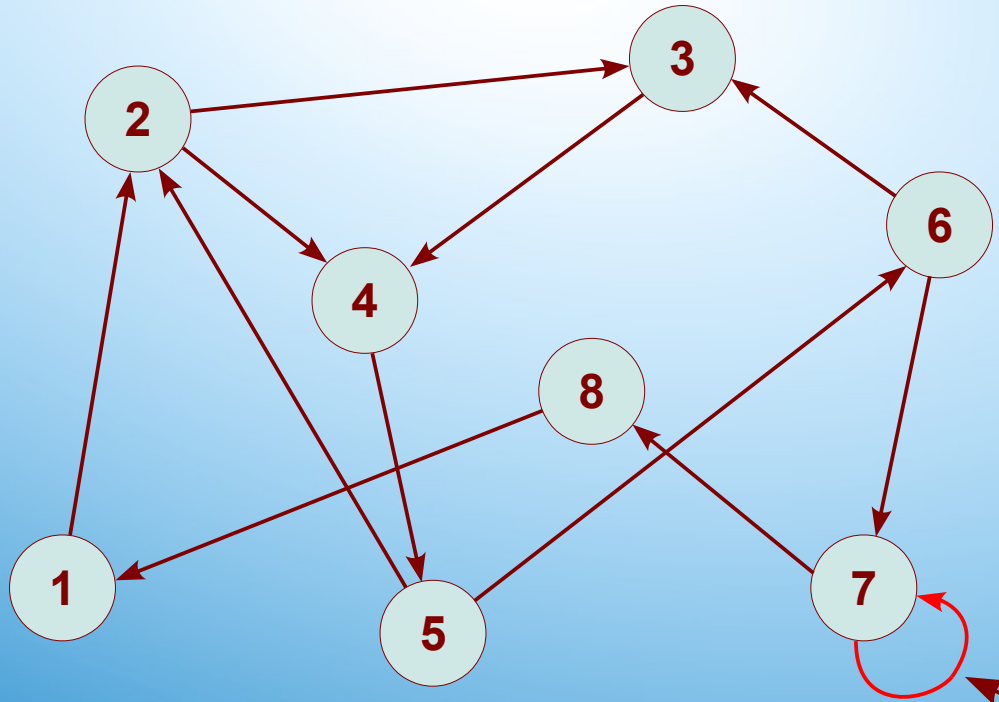


A[]	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	1
2	1	0	0	0	1	0	0	0
3	0	1	0	0	0	1	0	0
4	0	1	1	0	0	0	0	0
5	0	0	0	1	0	0	0	0
6	0	0	0	0	1	0	0	0
7	0	0	0	0	0	2	0	0
8	0	0	0	0	0	0	1	0

Dla krawędzi wielokrotnych w tablicy sąsiedztwa można wpisać liczbę krawędzi łączących dane węzły.

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

Graf skierowany:

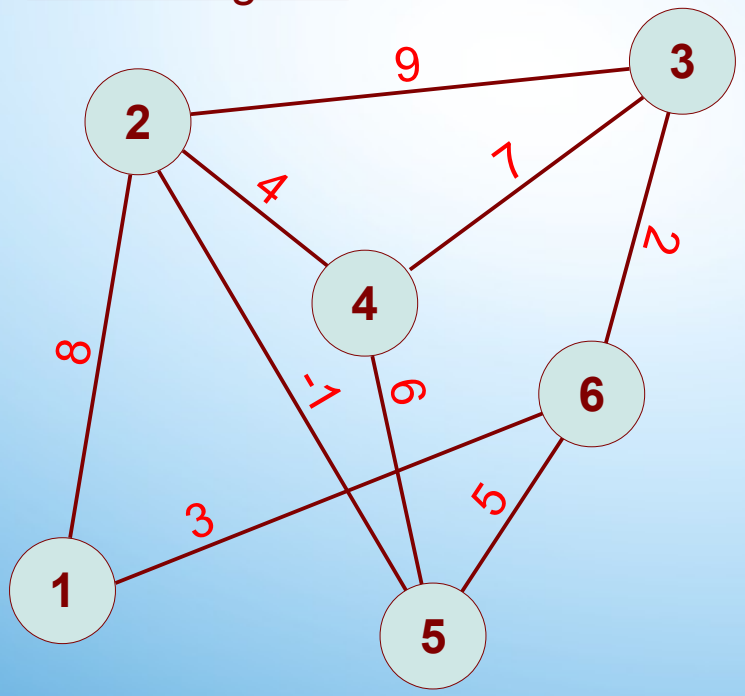


A[]	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	1
2	1	0	0	0	1	0	0	0
3	0	1	0	0	0	1	0	0
4	0	1	1	0	0	0	0	0
5	0	0	0	1	0	0	0	0
6	0	0	0	0	1	0	0	0
7	0	0	0	0	0	1	1	0
8	0	0	0	0	0	0	1	0

Pętle, jeśli występują w grafie, zapisuje się na przekątnej macierzy sąsiedztwa.

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

Graf z wagami:

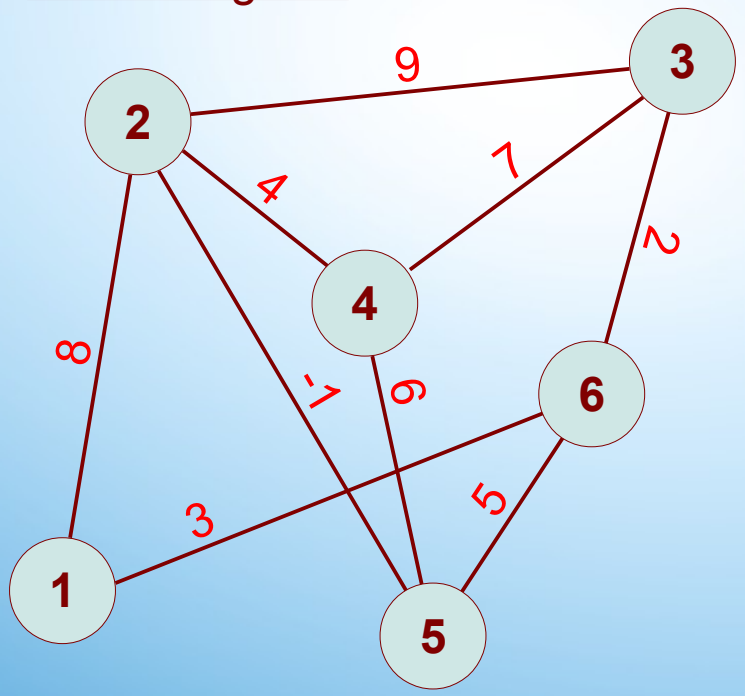


A[]	1	2	3	4	5	6
1	0	1	0	0	0	1
2	1	0	1	1	1	0
3	0	1	0	1	0	1
4	0	1	1	0	1	0
5	0	1	0	1	0	1
6	1	0	1	0	1	0

W[]	1	2	3	4	5	6
1	0	8	0	0	0	3
2	8	0	9	4	-1	0
3	0	9	0	7	0	2
4	0	4	7	0	6	0
5	0	-1	0	6	0	5
6	3	0	2	0	5	0

Macierz sąsiedztwa – jest to tablica, w której przechowuje się informacje o połączeniach węzłów ze sobą.

Graf z wagami:



A[]	1	2	3	4	5	6
1	0	1	0	0	0	1
2	1	0	1	1	1	0
3	0	1	0	1	0	1
4	0	1	1	0	1	0
5	0	1	0	1	0	1
6	1	0	1	0	1	0

W[]	1	2	3	4	5	6
1	0	8	0	0	0	3
2	8	0	9	4	-1	0
3	0	9	0	7	0	2
4	0	4	7	0	6	0
5	0	-1	0	6	0	5
6	3	0	2	0	5	0

Uwagi:

- jeśli wagi nigdy nie przyjmują wartości 0 to można je zapamiętać w tabeli A,
- elementy tabeli A mogą być strukturami dwupolowymi, np. w pierwszym można zapamiętać informacje o krawędzi, a w drugim jej wagę.

Lista sąsiedztwa – jest strukturą danych, w której dla każdego węzła grafu zapamiętuje się listę węzłów z którymi jest połączony.

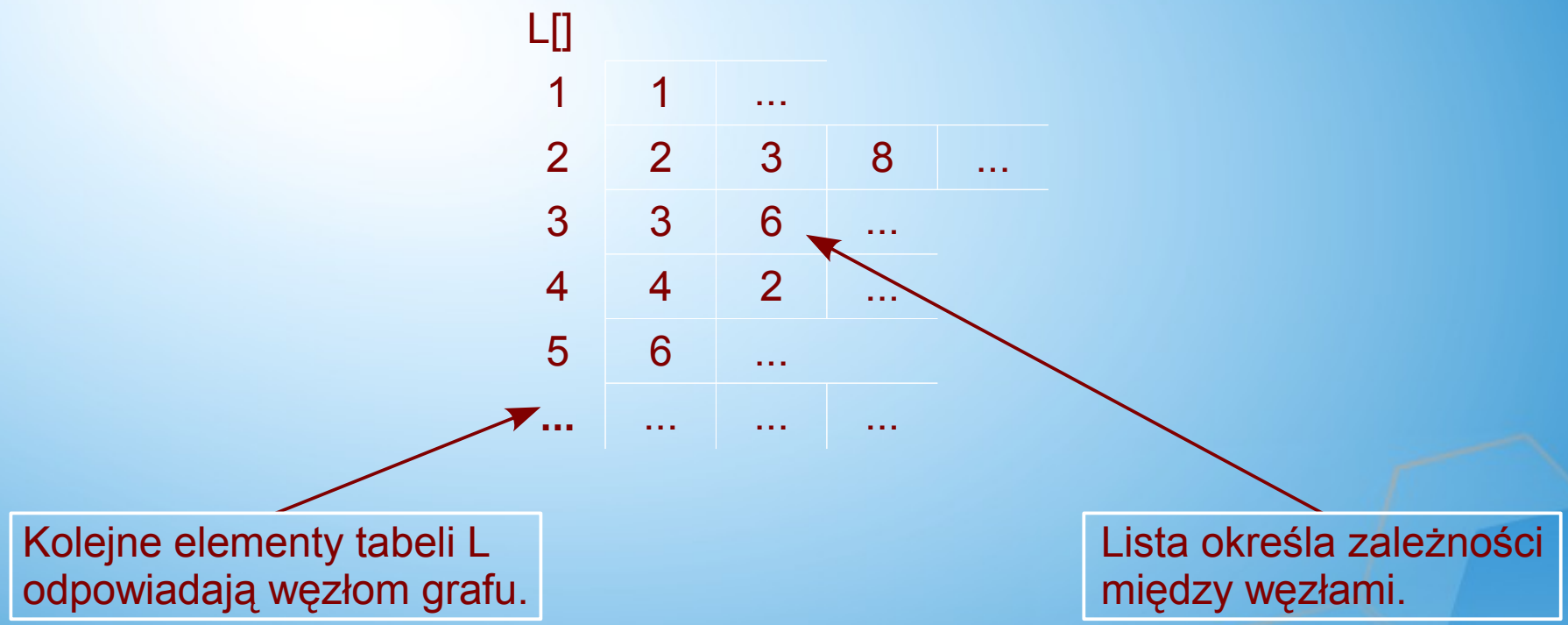
L[]				
1	1	...		
2	2	3	8	...
3	3	6	...	
4	4	2	...	
5	6	...		
...	...	...	...	

Lista sąsiedztwa – jest strukturą danych, w której dla każdego węzła grafu zapamiętuje się listę węzłów z którymi jest połączony.

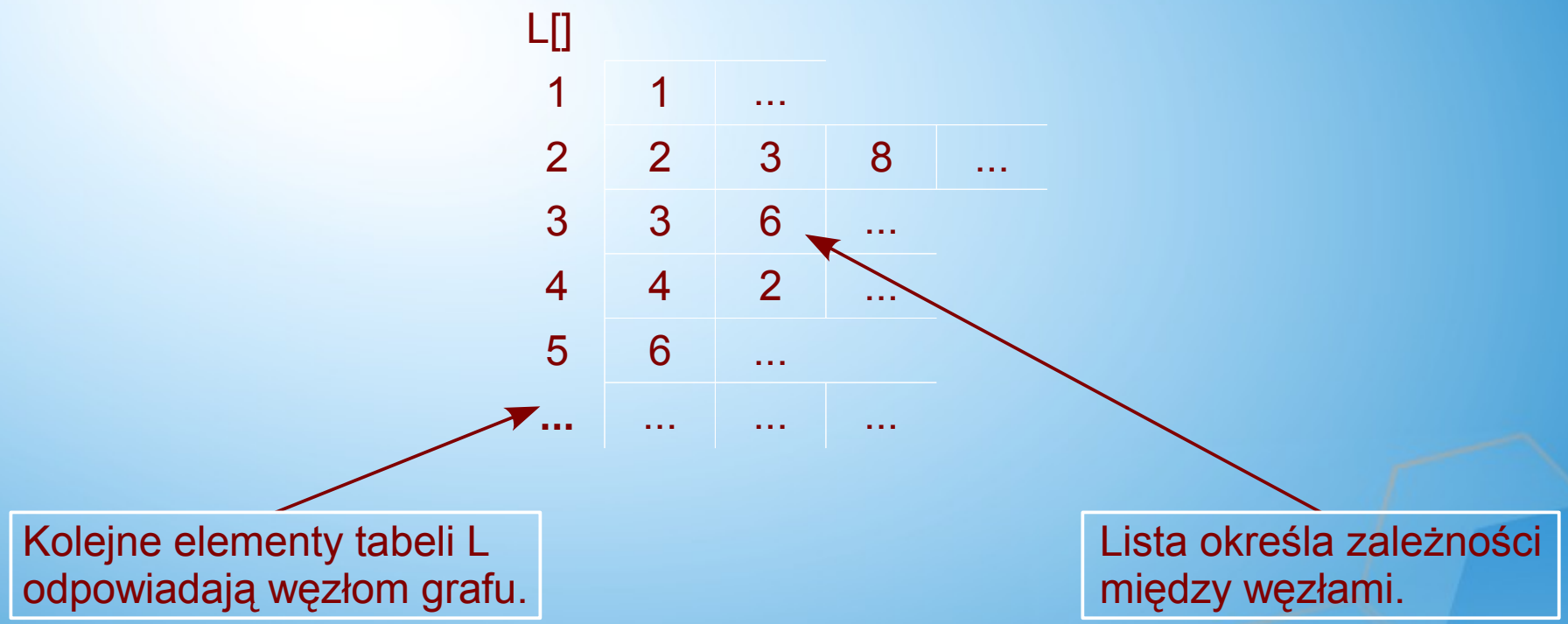
L[]				
1	1	...		
2	2	3	8	...
3	3	6	...	
4	4	2	...	
5	6	...		
...	...	...	...	

Kolejne elementy tabeli L odpowiadają węzłom grafu.

Lista sąsiedztwa – jest strukturą danych, w której dla każdego węzła grafu zapamiętuje się listę węzłów z którymi jest połączony.



Lista sąsiedztwa – jest strukturą danych, w której dla każdego węzła grafu zapamiętuje się listę węzłów z którymi jest połączony.

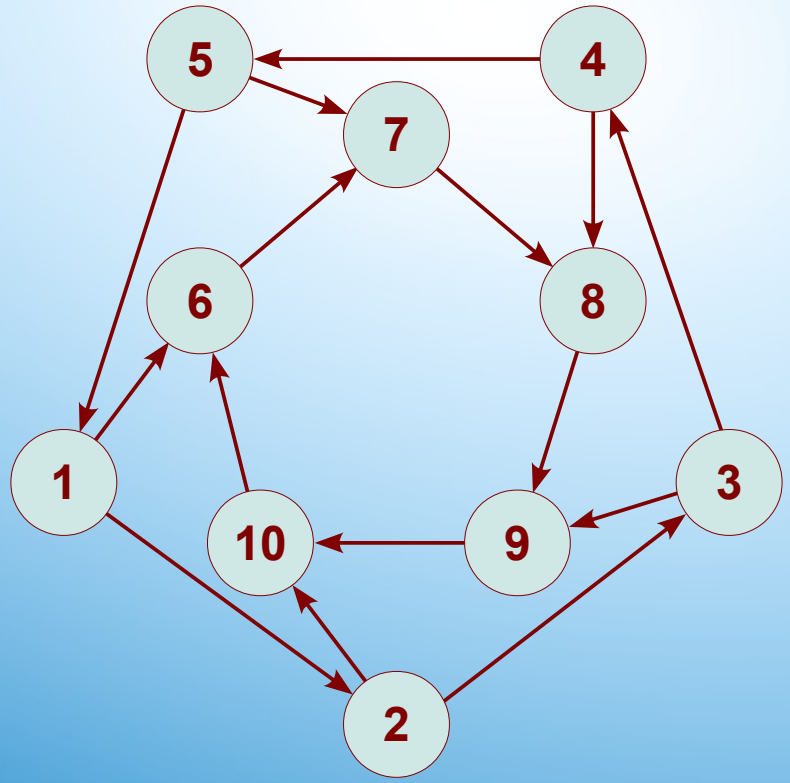


Uwagi:

- lista sąsiedztwa pozwala szybko określić, czy dane węzły łączy jakaś krawędź,
- zazwyczaj lista sąsiedztwa jest strukturą listową, węzły w L także mogą tworzyć listę,
- brak listy odpowiadającej danemu elementowi w L oznacza, że jest to węzeł wyizolowany z grafu.

Lista sąsiedztwa – jest strukturą danych, w której dla każdego węzła grafu zapamiętuje się listę węzłów z którymi jest połączony.

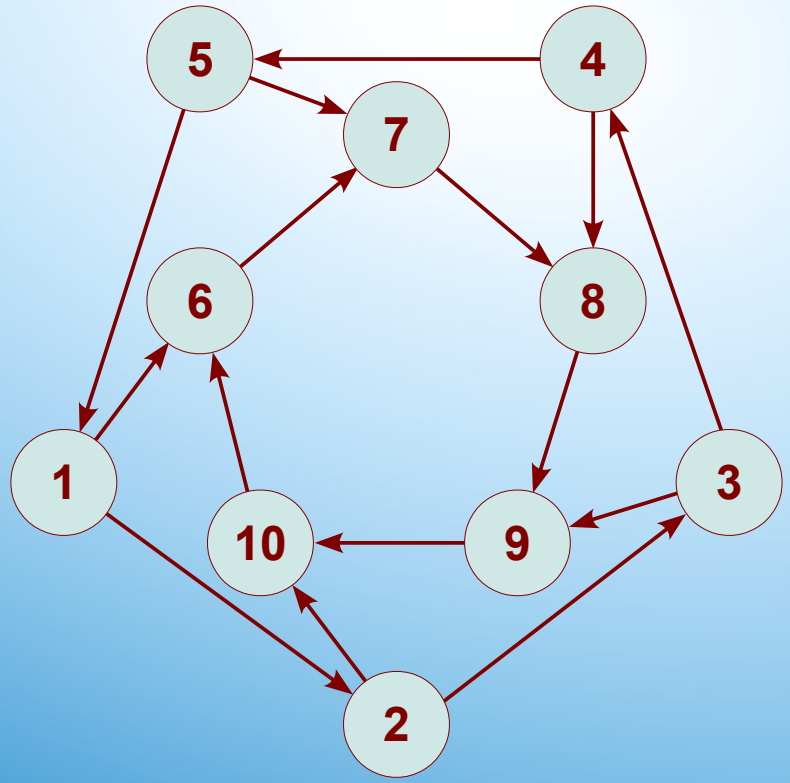
Graf skierowany:



L[]		
1	2	6
2	3	10
3	4	9
4	5	8
5	1	7
6	7	
7	8	
8	9	
9	10	
10	6	

Lista sąsiedztwa – jest strukturą danych, w której dla każdego węzła grafu zapamiętuje się listę węzłów z którymi jest połączony.

Graf skierowany:

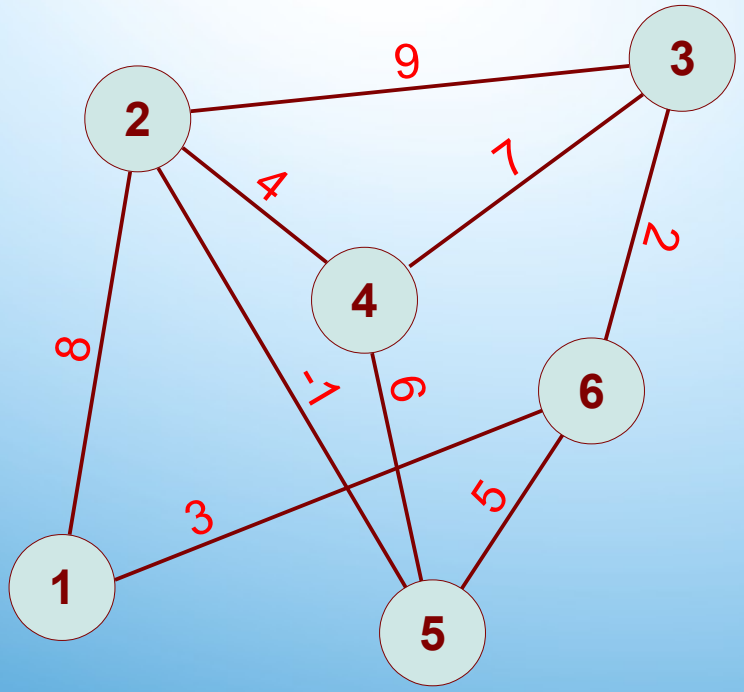


L[]		
1	2	6
2	3	10
3	4	9
4	5	8
5	1	7
6	7	
7	8	
8	9	
9	10	
10	6	

Elementy w wierszach są zazwyczaj listami.

Lista sąsiedztwa – jest strukturą danych, w której dla każdego węzła grafu zapamiętuje się listę węzłów z którymi jest połączony.

Graf z wagami:



L[]				
1	2,8	6,3		
2	1,8	3,9	4,4	5,-1
3	2,9	4,7	6,2	
4	2,4	3,7	5,6	
5	2,-1	4,6	6,5	
6	1,3	3,2	5,5	

Macierz incydencji – jest to tablica, w której przechowuje się informacje o krawędziach grafu i połączeniach między jego węzłami.

M[]	1	2	3	4	5	...
1	0	-1	0	0	0	...
2	1	0	1	0	-1	...
3	0	0	0	1	0	...
4	0	0	1	0	1	...
5	0	1	0	-1	0	...
...	...	...	...	...	...	...

Macierz incydencji – jest to tablica, w której przechowuje się informacje o krawędziach grafu i połączeniach między jego węzłami.

Wiersze tabeli M odpowiadają i-temu węzłowi grafu.

M[]	1	2	3	4	5	...
1	0	-1	0	0	0	...
2	1	0	1	1	-1	...
3	0	0	0	1	0	...
4	0	1	1	0	1	...
...	...	...	...	...	...	...

Macierz incydencji – jest to tablica, w której przechowuje się informacje o krawędziach grafu i połączeniach między jego węzłami.

M[]	1	2	3	4	5	...
1	0	-1	0	0	0	...
2	1	0	1	1	-1	...
3	0	0	0	1	0	...
4	0	1	1	0	1	...
...	...	...	...	...	...	...

Wiersze tabeli M odpowiadają i-temu węzłowi grafu.

Kolumny tabeli M odpowiadają j-tej krawędzi grafu.

Macierz incydencji – jest to tablica, w której przechowuje się informacje o krawędziach grafu i połączeniach między jego węzłami.

M[]	1	2	3	4	5	...
1	0	-1	0	0	0	...
2	1	0	1	1	-1	...
3	0	0	0	1	0	...
4	0	1	1	0	1	...
...	...	...	...	...	...	...

Wiersze tabeli M odpowiadają i-temu węzłowi grafu.

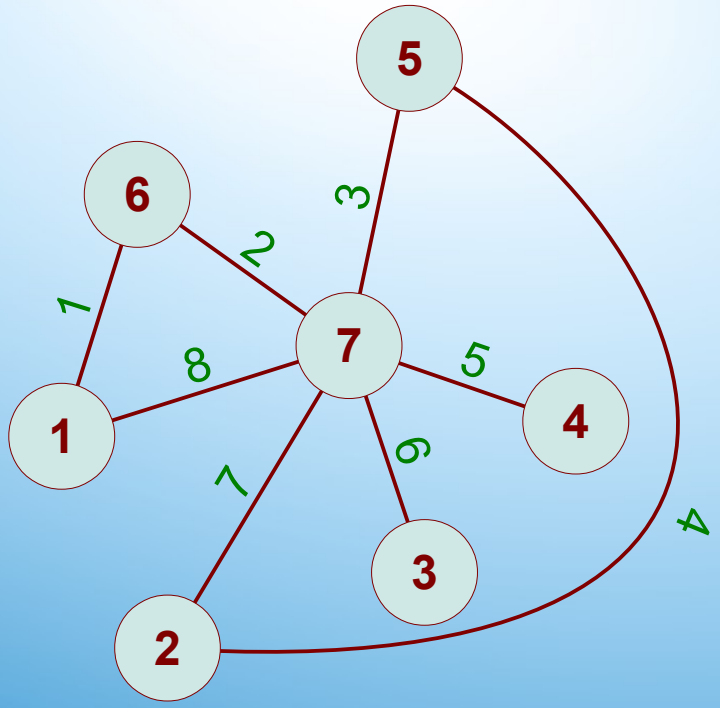
Kolumny tabeli M odpowiadają j-tej krawędzi grafu.

Zawartość tabeli określa zależności między węzłami, wg poniższego wzoru:

$$M_{ij} = \begin{cases} 1 & - \text{jeśli } v_i \text{ jest początkiem krawędzi } k_j, \\ -1 & - \text{jeśli } v_i \text{ jest końcem krawędzi } k_j, \\ 0 & - \text{jeśli węzeł } v_i \text{ i krawędź } k_j \text{ nie są połączone.} \end{cases}$$

Macierz incydencji – jest to tablica, w której przechowuje się informacje o krawędziach grafu i połączeniach między jego węzłami.

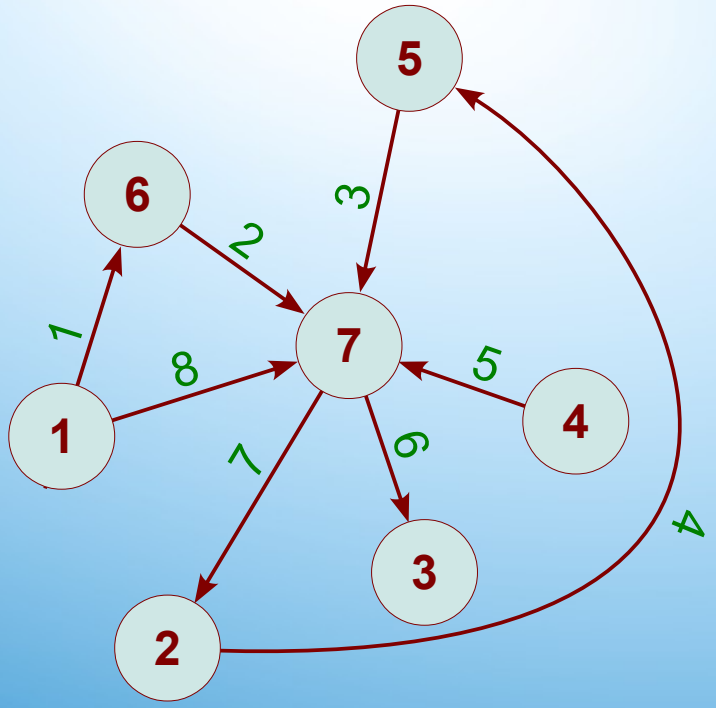
Graf nieskierowany:



M[]	1	2	3	4	5	6	7	8
1	1	0	0	0	0	0	0	1
2	0	0	0	1	0	0	1	0
3	0	0	0	0	0	1	0	0
4	0	0	0	0	1	0	0	0
5	0	0	1	1	0	0	0	0
6	1	1	0	0	0	0	0	0
7	0	1	1	0	1	1	1	1

Macierz incydencji – jest to tablica, w której przechowuje się informacje o krawędziach grafu i połączeniach między jego węzłami.

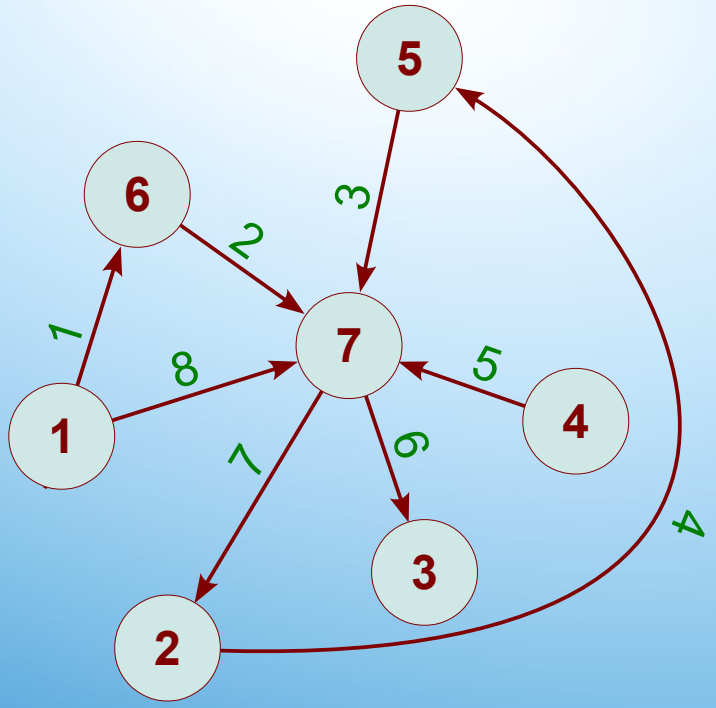
Graf skierowany:



M[]	1	2	3	4	5	6	7	8
1	1	0	0	0	0	0	0	1
2	0	0	0	1	0	0	-1	0
3	0	0	0	0	0	-1	0	0
4	0	0	0	0	1	0	0	0
5	0	0	1	-1	0	0	0	0
6	-1	1	0	0	0	0	0	0
7	0	-1	-1	0	-1	1	1	-1

Macierz incydencji – jest to tablica, w której przechowuje się informacje o krawędziach grafu i połączeniach między jego węzłami.

Graf skierowany:



M[]	1	2	3	4	5	6	7	8
1	1	0	0	0	0	0	0	1
2	0	0	0	1	0	0	-1	0
3	0	0	0	0	0	-1	0	0
4	0	0	0	0	1	0	0	0
5	0	0	1	-1	0	0	0	0
6	-1	1	0	0	0	0	0	0
7	0	-1	-1	0	-1	1	1	-1

Uwagi:

- wagi krawędzi można zapamiętać w dodatkowym polu w odpowiednim miejscu macierzy,
- macierz incydencji pozwala na szybkie stwierdzenie liczby krawędzi połączonych z wybranym węzłem grafu.

Algorytmy grafowe I

Przeszukiwanie w szerz – ang. breadth-first search (BFS) – służy do przeszukiwania grafu (drzewa) o V węzłach i E krawędziach.

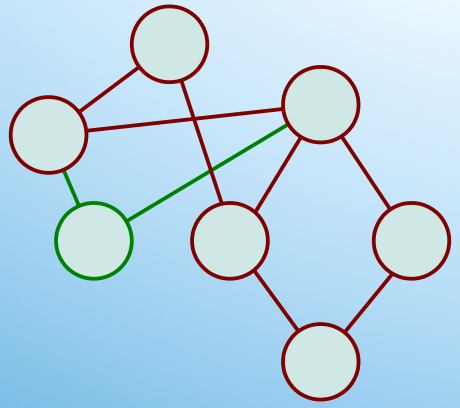
Zasada działania:

Algorytm rozpoczyna poszukiwania od wybranego węzła i przechodzi jeden raz przez każdy z jego sąsiadów. Następnie przechodzi przez kolejne węzły, będące w poprzednim kroku sąsiadami węzła pierwszego oraz ich sąsiadów, itd. – złożoność czasowa i pamięciowa wynosi O(|V| + |E|).

Przeszukiwanie w szerz – ang. breadth-first search (BFS) – służy do przeszukiwania grafu (drzewa) o V węzłach i E krawędziach.

Zasada działania:

Algorytm rozpoczyna poszukiwania od wybranego węzła i przechodzi jeden raz przez każdy z jego sąsiadów. Następnie przechodzi przez kolejne węzły, będące w poprzednim kroku sąsiadami węzła pierwszego oraz ich sąsiadów, itd. – złożoność czasowa i pamięciowa wynosi O(|V| + |E|).

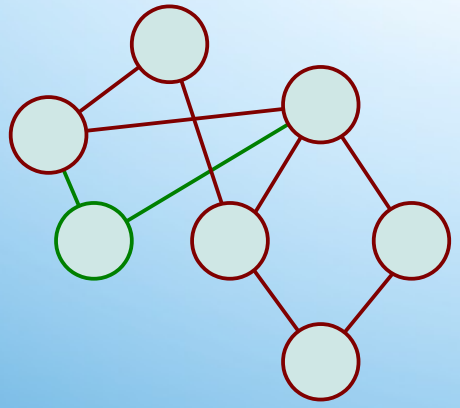


etap 1

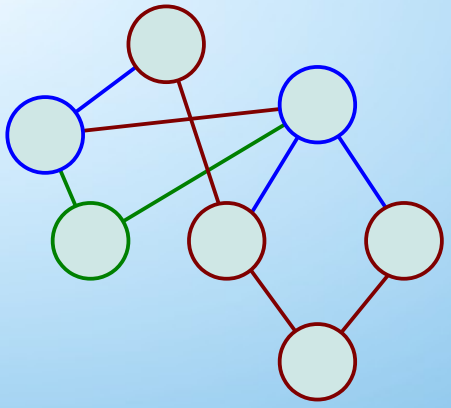
Przeszukiwanie w szerz – ang. breadth-first search (BFS) – służy do przeszukiwania grafu (drzewa) o V węzłach i E krawędziach.

Zasada działania:

Algorytm rozpoczyna poszukiwania od wybranego węzła i przechodzi jeden raz przez każdy z jego sąsiadów. Następnie przechodzi przez kolejne węzły, będące w poprzednim kroku sąsiadami węzła pierwszego oraz ich sąsiadów, itd. – złożoność czasowa i pamięciowa wynosi O(|V| + |E|).



etap 1

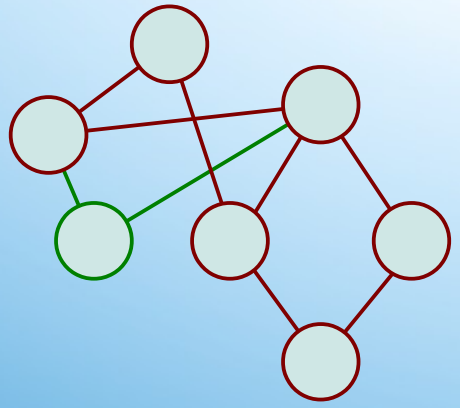


etap 2

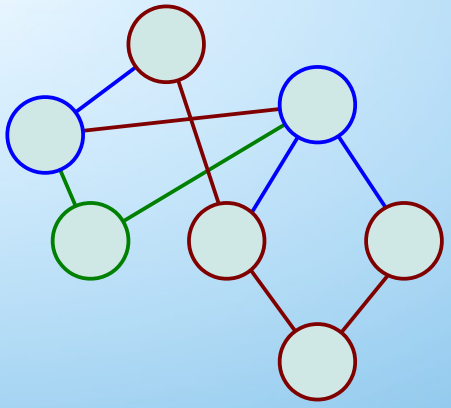
Przeszukiwanie w szerz – ang. breadth-first search (BFS) – służy do przeszukiwania grafu (drzewa) o V węzłach i E krawędziach.

Zasada działania:

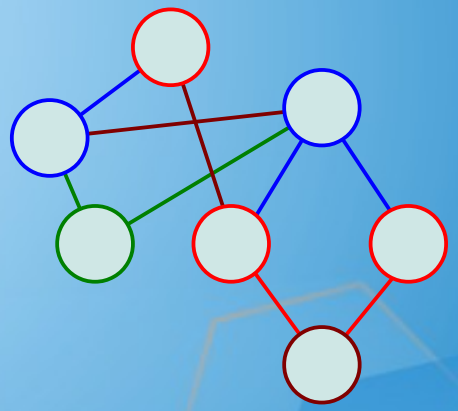
Algorytm rozpoczyna poszukiwania od wybranego węzła i przechodzi jeden raz przez każdy z jego sąsiadów. Następnie przechodzi przez kolejne węzły, będące w poprzednim kroku sąsiadami węzła pierwszego oraz ich sąsiadów, itd. – złożoność czasowa i pamięciowa wynosi O(|V| + |E|).



etap 1



etap 2

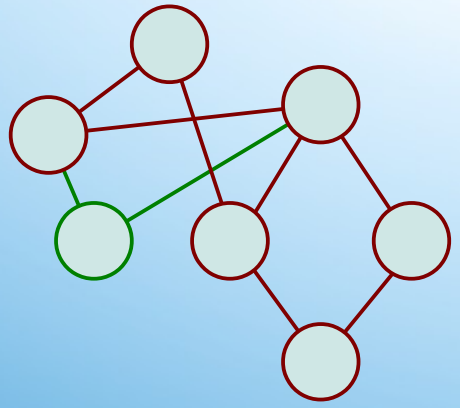


etap 3

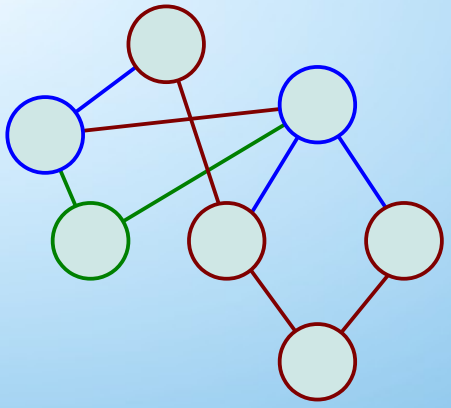
Przeszukiwanie w szerz – ang. breadth-first search (BFS) – służy do przeszukiwania grafu (drzewa) o V węzłach i E krawędziach.

Zasada działania:

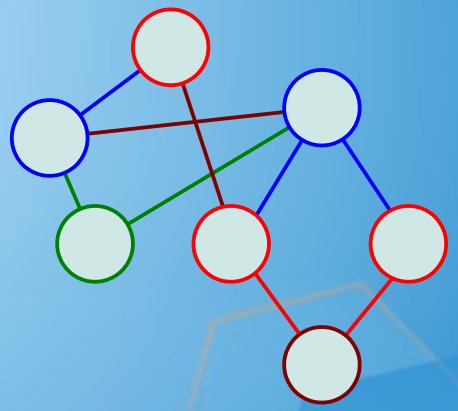
Algorytm rozpoczyna poszukiwania od wybranego węzła i przechodzi jeden raz przez każdy z jego sąsiadów. Następnie przechodzi przez kolejne węzły, będące w poprzednim kroku sąsiadami węzła pierwszego oraz ich sąsiadów, itd. – złożoność czasowa i pamięciowa wynosi O(|V| + |E|).



etap 1



etap 2



etap 3

Zastosowanie:

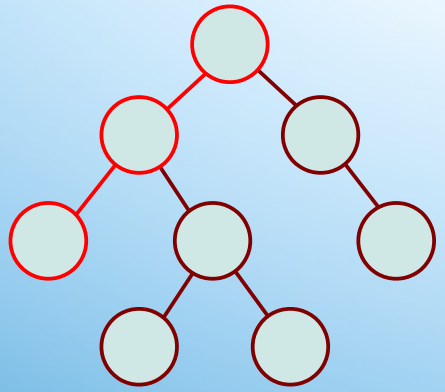
- odnalezienie wszystkich połączonych węzłów w grafie,
- odnalezienie najkrótszej ścieżki między dwoma węzłami w grafie nieważonym,
- rozstrzygnięcie dwudzielności grafu.

Przeszukiwanie w głąb – ang. depth-first search (DFS) – jest to algorytm przeszukujący graf (drzewo) poprzez odwiedzenie wszystkich węzłów połączonych z rozpatrywanym węzłem.

Zasada działania:
Przeszukiwanie zaczyna się od wybranego węzła (korzenia w drzewie) i porusza się w dół do krańcowej gałęzi ścieżki lub ostatniej gałęzi drzewa, po czym wraca się o jeden poziom i próbuje kolejne krawędzie itd. – złożoność czasowa i pamięciowa wynosi O(|V| + |E|).

Przeszukiwanie w głąb – ang. depth-first search (DFS) – jest to algorytm przeszukujący graf (drzewo) poprzez odwiedzenie wszystkich węzłów połączonych z rozpatrywanym węzłem.

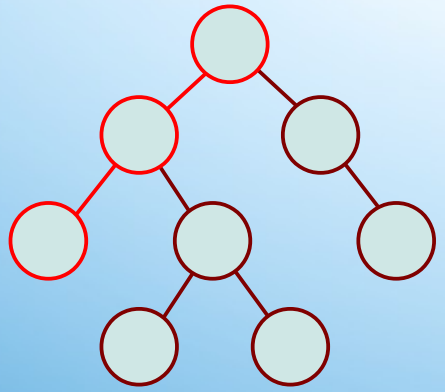
Zasada działania:
Przeszukiwanie zaczyna się od wybranego węzła (korzenia w drzewie) i porusza się w dół do krańcowej gałęzi ścieżki lub ostatniej gałęzi drzewa, po czym wraca się o jeden poziom i próbuje kolejne krawędzie itd. – złożoność czasowa i pamięciowa wynosi O(|V| + |E|).



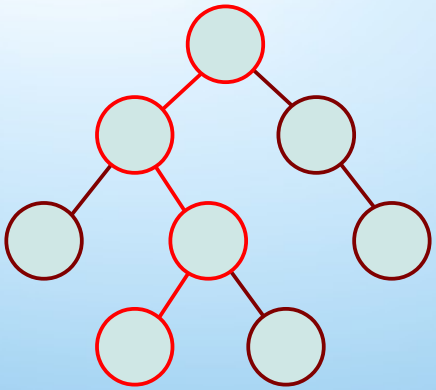
etap 1

Przeszukiwanie w głąb – ang. depth-first search (DFS) – jest to algorytm przeszukujący graf (drzewo) poprzez odwiedzenie wszystkich węzłów połączonych z rozpatrywanym węzłem.

Zasada działania:
Przeszukiwanie zaczyna się od wybranego węzła (korzenia w drzewie) i porusza się w dół do krańcowej gałęzi ścieżki lub ostatniej gałęzi drzewa, po czym wraca się o jeden poziom i próbuje kolejne krawędzie itd. – złożoność czasowa i pamięciowa wynosi O(|V| + |E|).



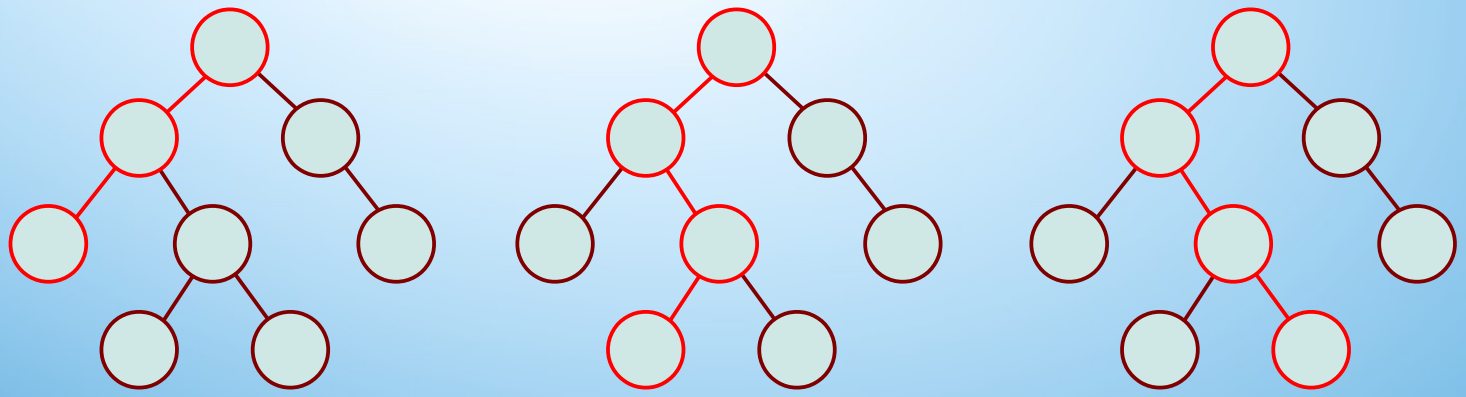
etap 1



etap 2

Przeszukiwanie w głąb – ang. depth-first search (DFS) – jest to algorytm przeszukujący graf (drzewo) poprzez odwiedzenie wszystkich węzłów połączonych z rozpatrywanym węzłem.

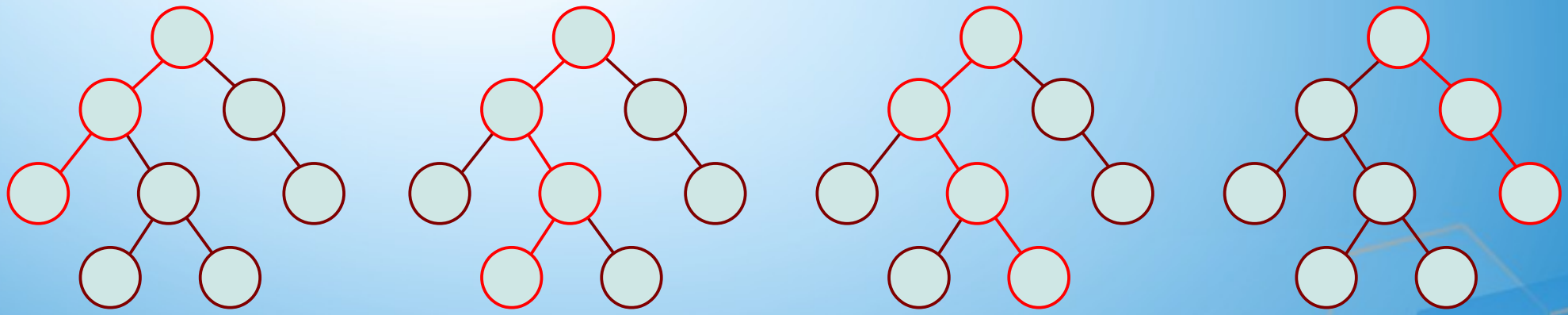
Zasada działania:
Przeszukiwanie zaczyna się od wybranego węzła (korzenia w drzewie) i porusza się w dół do krańcowej gałęzi ścieżki lub ostatniej gałęzi drzewa, po czym wraca się o jeden poziom i próbuje kolejne krawędzie itd. – złożoność czasowa i pamięciowa wynosi O(|V| + |E|).



etap 1 etap 2 etap 3

Przeszukiwanie w głąb – ang. depth-first search (DFS) – jest to algorytm przeszukujący graf (drzewo) poprzez odwiedzenie wszystkich węzłów połączonych z rozpatrywanym węzłem.

Zasada działania:
Przeszukiwanie zaczyna się od wybranego węzła (korzenia w drzewie) i porusza się w dół do krańcowej gałęzi ścieżki lub ostatniej gałęzi drzewa, po czym wraca się o jeden poziom i próbuje kolejne krawędzie itd. – złożoność czasowa i pamięciowa wynosi O(|V| + |E|).

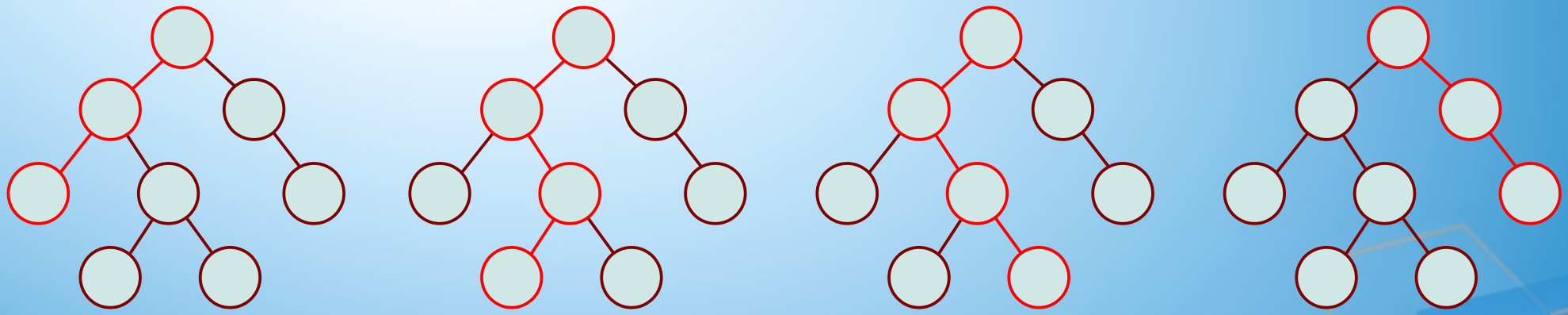


etap 1 etap 2 etap 3 etap 4

Przeszukiwanie w głąb – ang. depth-first search (DFS) – jest to algorytm przeszukujący graf (drzewo) poprzez odwiedzenie wszystkich węzłów połączonych z rozpatrywanym węzłem.

Zasada działania:

Przeszukiwanie zaczyna się od wybranego węzła (korzenia w drzewie) i porusza się w dół do krańcowej gałęzi ścieżki lub ostatniej gałęzi drzewa, po czym wraca się o jeden poziom i próbuje kolejne krawędzie itd. – złożoność czasowa i pamięciowa wynosi O(|V| + |E|).



etap 1 etap 2 etap 3 etap 4

Zastosowanie:

- znajdywania najkrótszych ścieżek między dwoma węzłami w drzewie,
- sprawdzania, czy istnieje ścieżka między dwoma węzłami – badanie spójności grafu,
- wyznaczania spójnych składowych – podgrafów, które można wydzielić z grafu bez usuwania krawędzi

Minimalne drzewo rozpinające – ang. minimum spanning tree (MST), jest to drzewo rozpinające grafu o sumie wag najmniejszej z możliwych.

Algorytm Prima:

1. Utworzenie drzewa zawierającego jeden węzeł, dowolnie wybrany z grafu.
2. Utworzenie kolejki priorytetowej, zawierającej węzły znajdujące się już z MST (na początku będzie to jeden węzeł, więc na początku w kolejce będą sąsiedzi początkowego węzła), o priorytecie najmniejszego kosztu dotarcia do danego węzła z MST.
3. Powtarzanie, dopóki drzewo nie obejmuje wszystkich węzłów grafu:
   - wśród nieprzetworzonych węzłów (spoza obecnego MST) wybiera się ten, dla którego koszt dojścia z obecnego MST jest najmniejszy,
   - dodanie tego węzła do obecnego MST,
   - aktualizacja kolejki z uwzględnieniem nowych krawędzi wychodzących z dodanego węzła.

Uwagi:

- złożoność obliczeniowa zależy od sposobu implementacji kolejki priorytetowej, np. dla kolejki opartej na stercie wynosi O(|E| log (|V|)).

Minimalne drzewo rozpinające – ang. minimum spanning tree (MST), jest to drzewo rozpinające grafu o sumie wag najmniejszej z możliwych.

Algorytm Kruskala:

1. Utworzenie lasu L z węzłów grafu wejściowego – każdy węzeł jest na początku osobnym drzewem.
2. Utworzenie zbioru S zawierającego wszystkie krawędzie grafu wejściowego.
3. Dopóki zbiór S nie jest pusty:
   - wybranie i usunięcie z S krawędzi o minimalnej wadze,
   - jeśli wybrana krawędź łączyła dwa różne drzewa, to dodanie jej do lasu L, tak aby połączyła dwa odpowiadające drzewa w jedno, w przeciwnym wypadku usunięcie krawędzi.
4. Jeśli usunięcie krawędzi spowoduje, że zbiór S jest pusty to L zawiera minimalne drzewo rozpinające.

Uwagi:

- złożoność obliczeniowa wynosi O(|E| log |V|),
- zbiór S może być tablicą krawędzi posortowaną wg wag.

Koniec wykładu