Teoria grafów - grafy z wagami

Aleksandra Gorzkowska

Elektronika i Telekomunikacja
Wydział Informatyki, Elektroniki i Telekomunikacji

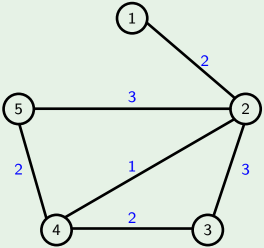
Definicja

Grafem z wagami nazywamy trójkę (V, E, w), gdzie $G = (V, E)$ jest grafem prostym a $w\colon E \to \mathbb{R}_+$ jest funkcją wagową, która każdej krawędzi przyporządkowuje wagę.

Definicja

Grafem z wagami nazywamy trójkę (V, E, w), gdzie $G = (V, E)$ jest grafem prostym a $w\colon E \to \mathbb{R}_+$ jest funkcją wagową, która każdej krawędzi przyporządkowuje wagę.

Przykład



Dany jest graf ważony $G = (V, E)$ z wagą w.

Problem: Znaleźć najkrótsze ścieżki z danego wierzchołka grafu G do wszystkich pozostałych wierzchołków.

Dany jest graf ważony $G = (V, E)$ z wagą w.

Problem: Znaleźć najkrótsze ścieżki z danego wierzchołka grafu G do wszystkich pozostałych wierzchołków.

Algorytm Dijkstry

Niech v_0 - wybrany wierzchołek, z którego szukamy ścieżek do pozostałych

Ponadto wprowadzany:

$d[v]$ - waga aktualnej ścieżki z v_0 do v;
$p[v]$ - poprzednik wierzchołka v na ścieżce z v_0 do v, która ma najmniejszą wagę;

Dany jest graf ważony $G = (V, E)$ z wagą w.

Problem: Znaleźć najkrótsze ścieżki z danego wierzchołka grafu G do wszystkich pozostałych wierzchołków.

Algorytm Dijkstry

Niech v_0 - wybrany wierzchołek, z którego szukamy ścieżek do pozostałych

Ponadto wprowadzany:

$d[v]$ - waga aktualnej ścieżki z v_0 do v;
$p[v]$ - poprzednik wierzchołka v na ścieżce z v_0 do v, która ma najmniejszą wagę;

Dla każdego wierzchołka $v \in V$ ustawiamy: $d[v] = \infty$; $p[v] =$ NULL;
Dodatkowo $d[v_0] = 0$.

Dany jest graf ważony $G = (V, E)$ z wagą w.

Problem: Znaleźć najkrótsze ścieżki z danego wierzchołka grafu G do wszystkich pozostałych wierzchołków.

Algorytm Dijkstry

Niech v_0 - wybrany wierzchołek, z którego szukamy ścieżek do pozostałych

Ponadto wprowadzany:

$d[v]$ - waga aktualnej ścieżki z v_0 do v;
$p[v]$ - poprzednik wierzchołka v na ścieżce z v_0 do v, która ma najmniejszą wagę;

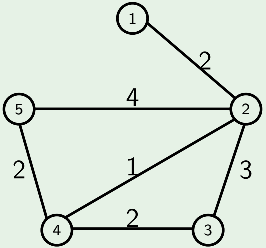
Dla każdego wierzchołka $v \in V$ ustawiamy: $d[v] = \infty$; $p[v] =$ NULL;
Dodatkowo $d[v_0] = 0$.

S - zbiór rozpatrzonych wierzchołków; Ustawiamy $S = \emptyset$

Algorytm Dijkstry

DIJKSTRA(v_0) {
dopóki $S \neq V$ {
wybierz wierzchołek $u \notin S$, taki że $d[u]$ jest najmniejsze;
$S = S \cup \{u\}$; // dodaj u do S
dla każdego $v \in V \setminus S$ sąsiada u
jeśli $d[v] > d[u] + w(uv)$ {
$d[v] = d[u] + w(uv)$;
$p[v] = u$;
}
}
}

Przykład

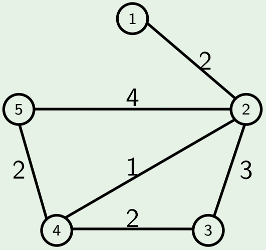


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	
2^o						
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o					
3^o					
4^o					
5^o					

Przykład

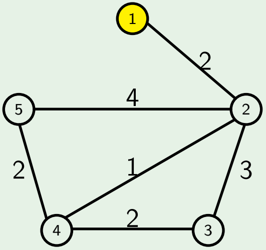


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o						
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o					
3^o					
4^o					
5^o					

Przykład

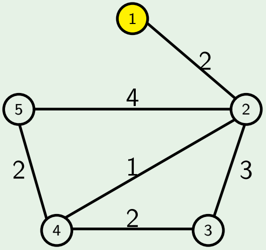


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o						
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o					
3^o					
4^o					
5^o					

Przykład

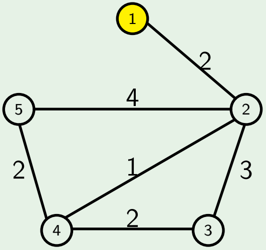


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2				
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1			
3^o					
4^o					
5^o					

Przykład

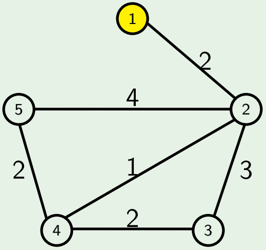


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o					
4^o					
5^o					

Przykład

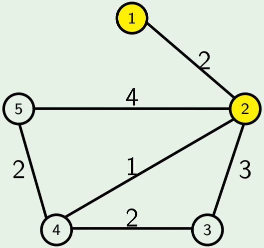


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o					
4^o					
5^o					

Przykład

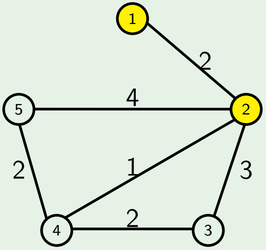


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o					
4^o					
5^o					

Przykład

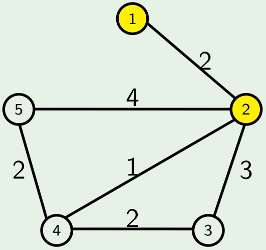


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o			5			
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o			2		
4^o					
5^o					

Przykład

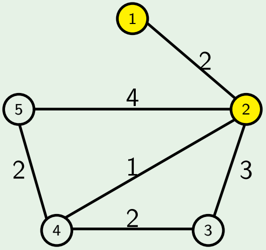


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o			5	3		
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o			2	2	
4^o					
5^o					

Przykład

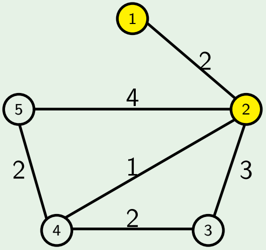


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o			5	3	6	
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o			2	2	2
4^o					
5^o					

Przykład

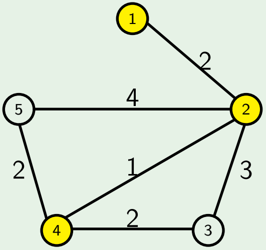


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o			5	3	6	4
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o			2	2	2
4^o					
5^o					

Przykład

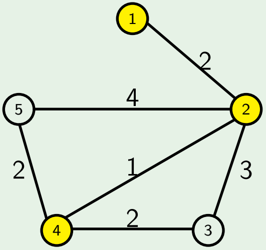


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o			5	3	6	4
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o			2	2	2
4^o					
5^o					

Przykład

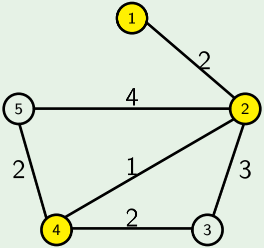


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o			5	3	6	4
4^o			5			
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o			2	2	2
4^o			2		
5^o					

Przykład

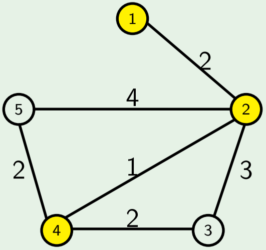


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o			5	3	6	4
4^o			5		5	
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o			2	2	2
4^o			2		4
5^o					

Przykład

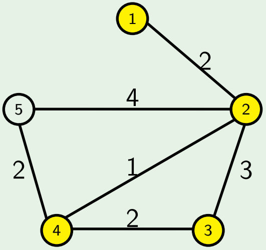


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o			5	3	6	4
4^o			5		5	3
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o			2	2	2
4^o			2		4
5^o					

Przykład

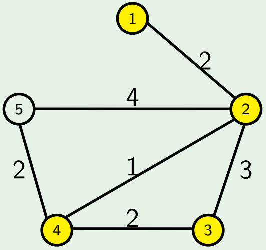


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o			5	3	6	4
4^o			5		5	3
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o			2	2	2
4^o			2		4
5^o					

Przykład

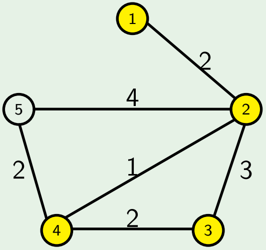


$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o			5	3	6	4
4^o			5		5	3
5^o					5	

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o			2	2	2
4^o			2		4
5^o					4

Przykład



$v_0 = 1$

d[v]	1	2	3	4	5	+S
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	∞	2
3^o			5	3	6	4
4^o			5		5	3
5^o					5	5

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	NULL
3^o			2	2	2
4^o			2		4
5^o					4

Wynik działania algorytmu Dijkstry

najkrótsze ścieżki w grafie G z wierzchołka 1 do wszystkich pozostałych:

z 1 do 2: $1 \to 2$ waga 2;
z 1 do 3: $1 \to 2 \to 3$ waga 5;
z 1 do 4: $1 \to 2 \to 4$ waga 3;
z 1 do 5: $1 \to 2 \to 4 \to 5$ waga 5;

Dany jest spójny graf ważony $G = (V, E)$ z wagą w.

Problem: Znaleźć drzewo rozpinające grafu G o najmniejszej sumarycznej wadze.

Dany jest spójny graf ważony $G = (V, E)$ z wagą w.

Problem: Znaleźć drzewo rozpinające grafu G o najmniejszej sumarycznej wadze.

Algorytm Prima

Wprowadzamy:

$k[v]$ - minimalna waga krawędzi łączącej v z drzewem;
$p[v]$ - sąsiad wierzchołka v w drzewie, realizujący połączenie o koszcie $k[v]$;

Dany jest spójny graf ważony $G = (V, E)$ z wagą w.

Problem: Znaleźć drzewo rozpinające grafu G o najmniejszej sumarycznej wadze.

Algorytm Prima

Wprowadzamy:

$k[v]$ - minimalna waga krawędzi łączącej v z drzewem;
$p[v]$ - sąsiad wierzchołka v w drzewie, realizujący połączenie o koszcie $k[v]$;

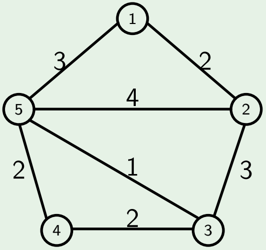
Dla każdego wierzchołka $v \in V$ ustawiamy: $k[v] = \infty$; $p[v] =$ NULL;

Wybieramy dowolny wierzchołek v_0 i ustawiamy $k[v] = 0$.
Tworzymy drzewo $T = (V, F)$ i ustawiamy $F = \emptyset$.

Algorytm Prima

PRIM(v_0) $\{$
    dopóki $V \neq \emptyset$ $\{$
        wybierz wierzchołek $u \in V$, taki że $k[u]$ jest najmniejsze;
        $V = V \setminus \{u\}$; // usuń u z V
        jeśli $u \neq v_0$, to dodaj $up[u]$ do drzewa T;
        dla każdego $v \in V$ sąsiada u
            jeśli $k[v] > w(uv)$ $\{$
                $k[v] = w(uv)$;
                $p[v] = u$;
            $\}$
    $\}$
$\}$

Przykład

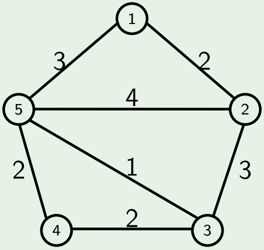


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	
2^o						
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o					
3^o					
4^o					
5^o					

Przykład

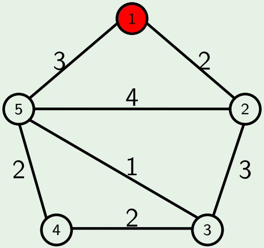


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o						
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o					
3^o					
4^o					
5^o					

Przykład

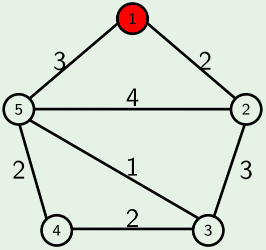


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o						
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o					
3^o					
4^o					
5^o					

Przykład

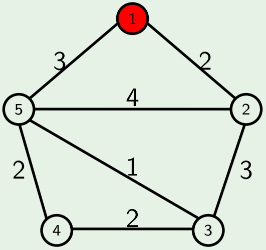


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2				
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1			
3^o					
4^o					
5^o					

Przykład

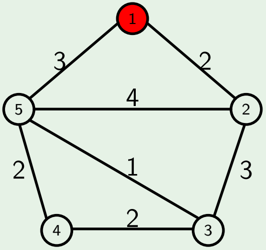


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2			3	
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1			1
3^o					
4^o					
5^o					

Przykład

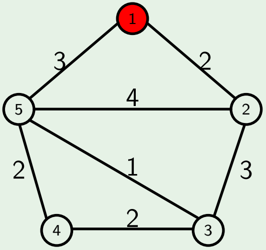


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o					
4^o					
5^o					

Przykład

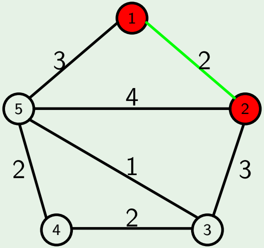


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o					
4^o					
5^o					

Przykład

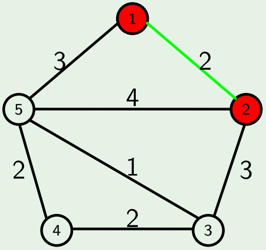


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o						
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o					
4^o					
5^o					

Przykład

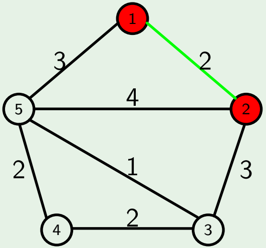


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3			
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2		
4^o					
5^o					

Przykład

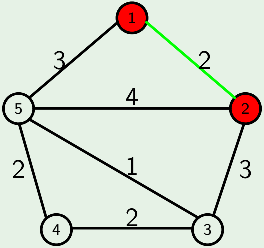


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3		3	
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2		1
4^o					
5^o					

Przykład

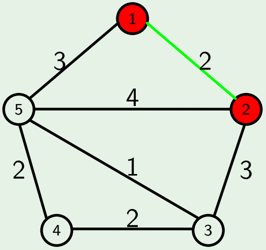


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3	∞	3	
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2	NULL	1
4^o					
5^o					

Przykład

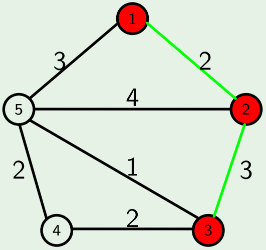


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3	∞	3	3
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2	NULL	1
4^o					
5^o					

Przykład

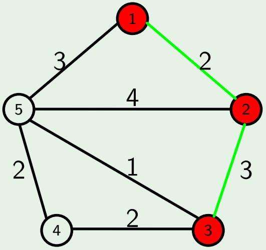


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3	∞	3	3
4^o						
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2	NULL	1
4^o					
5^o					

Przykład

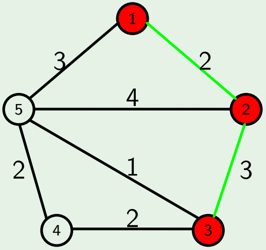


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3	∞	3	3
4^o				2		
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2	NULL	1
4^o				3	
5^o					

Przykład

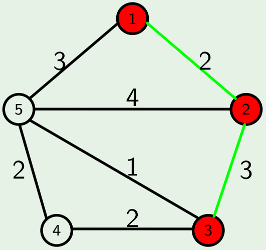


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3	∞	3	3
4^o				2	1	
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2	NULL	1
4^o				3	3
5^o					

Przykład

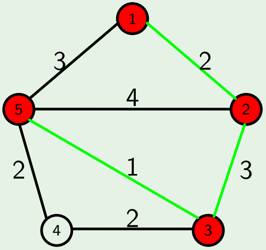


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3	∞	3	3
4^o				2	1	5
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2	NULL	1
4^o				3	3
5^o					

Przykład

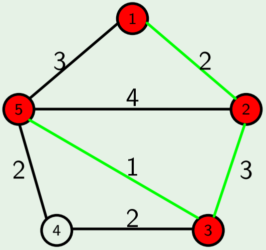


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3	∞	3	3
4^o				2	1	4
5^o						

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2	NULL	1
4^o				3	3
5^o					

Przykład

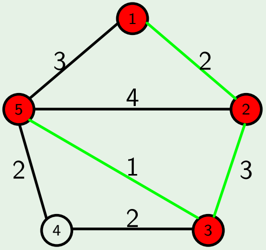


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3	∞	3	3
4^o				2	1	4
5^o				2		

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2	NULL	1
4^o				3	3
5^o				3	

Przykład

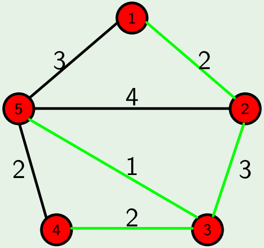


$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3	∞	3	3
4^o				2	1	4
5^o				2		5

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2	NULL	1
4^o				3	3
5^o				3	

Przykład



$v_0 = 1$

k[v]	1	2	3	4	5	-V
1^o	0	∞	∞	∞	∞	1
2^o		2	∞	∞	3	2
3^o			3	∞	3	3
4^o				2	1	5
5^o				2		4

p[v]	1	2	3	4	5
1^o	NULL	NULL	NULL	NULL	NULL
2^o		1	NULL	NULL	1
3^o			2	NULL	1
4^o				3	3
5^o				3	

Waga drzewa rozpinającego T to $2 + 3 + 1 + 2 = 8$.

Dany jest spójny graf ważony $G = (V, E)$ z wagą w.

Problem: Znaleźć drzewo rozpinające grafu G o najmniejszej sumarycznej wadze.

Dany jest spójny graf ważony $G = (V, E)$ z wagą w.

Problem: Znaleźć drzewo rozpinające grafu G o najmniejszej sumarycznej wadze.

Algorytm Kruskala

Wprowadzamy:

P -ciąg krawędzi grafu ustawionych w kolejności niemalejącej względem wag; $T = (V, F)$, $F = \emptyset$ - las wierzchołków izolowanych;

Algorytm Kruskala

KRUSKAL(G**)** $\{$

    dopóki T nie jest spójny $\{$

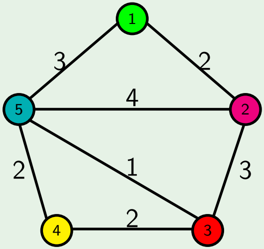
        pobierz wyraz uv ciągu P;

        jeśli u oraz v są w rożnych składowych T, to dodaj uv do lasu T;

    $\}$

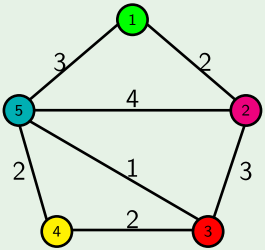
$\}$

Przykład



Ciąg krawędzi:
35, 12, 34, 45, 15, 23, 25

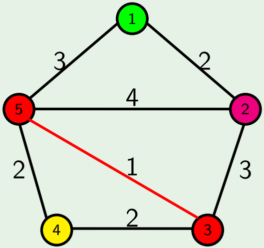
Przykład



Ciąg krawędzi:
35, 12, 34, 45, 15, 23, 25

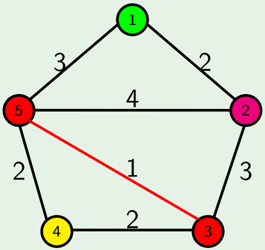
Pobieramy krawędź 35,
3 oraz 5 są w różnych składowych

Przykład



Ciąg krawędzi:
35, 12, 34, 45, 15, 23, 25

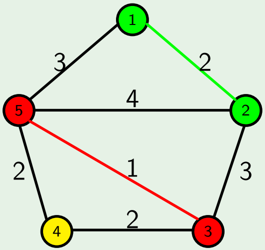
Przykład



Ciąg krawędzi:
35, 12, 34, 45, 15, 23, 25

Pobieramy krawędź 12,
1 oraz 2 są w różnych składowych

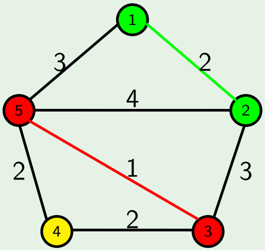
Przykład



Ciąg krawędzi:

35, 12, 34, 45, 15, 23, 25

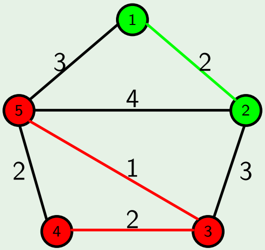
Przykład



Ciąg krawędzi:
35, 12, 34, 45, 15, 23, 25

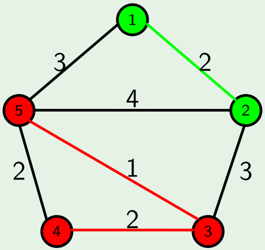
Pobieramy krawędź 34,
3 oraz 4 są w różnych składowych

Przykład



Ciąg krawędzi:
35, 12, 34, 45, 15, 23, 25

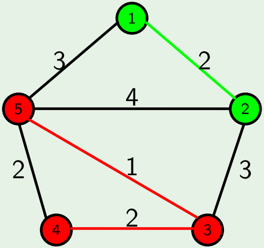
Przykład



Ciąg krawędzi:
35, 12, 34, 45, 15, 23, 25

Pobieramy krawędź 45,
4 oraz 5 są w tej samej składowej

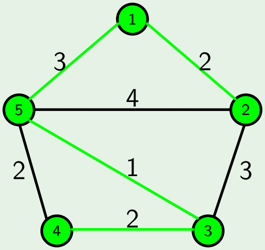
Przykład



Ciąg krawędzi:
35, 12, 34, 45, 15, 23, 25

Pobieramy krawędź 15,
1 oraz 5 są w różnych składowych

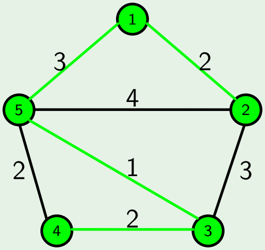
Przykład



Ciąg krawędzi:

35, 12, 34, 45, 15, 23, 25

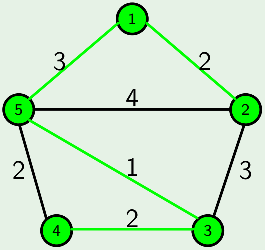
Przykład



Ciąg krawędzi:
35, 12, 34, 45, 15, 23, 25

Graf T jest drzewem, jest spójny. Algorytm kończy działanie.

Przykład



Ciąg krawędzi:
35, 12, 34, 45, 15, 23, 25

Graf T jest drzewem, jest spójny. Algorytm kończy działanie.

Waga drzewa rozpinającego T to $1 + 2 + 2 + 3 = 8$.