

Wykłady z Matematyki Dyskretnej

dla kierunku Informatyka

dr Adam Marszałek

Instytut Informatyki
Politechnika Krakowska

Wykłady na bazie materiałów:
dra hab. Andrzeja Karafiata
dr hab. Joanny Kołodziej, prof. PK

Siecią przepływową (G, s, t, c) nazywamy graf skierowany $G = \langle V, E \rangle$, w którym są wyróżnione wierzchołki: **źródło** $s \in V$ i **ujście** $t \in V$ oraz z każdą krawędzią związana jest **funkcja przepustowości** $c: V \times V \rightarrow \mathbb{R}^+ \cup \{0\}$, taka, że

$$c(u, v) = \begin{cases} c(u, v) > 0 & \text{dla } (u, v) \in E \\ c(u, v) = 0 & \text{dla } (u, v) \notin E \end{cases}$$

W sieci przepływowej zakłada się też, że każdy wierzchołek u leży na pewnej drodze od źródła s do ujścia t .

$$s \rightsquigarrow u \rightsquigarrow t$$

Przepływem w sieci (G, s, t, c) nazywamy funkcję $f: V \times V \rightarrow \mathbb{R}$ spełniającą warunki:

- $$f(u, v) \leq c(u, v)$$

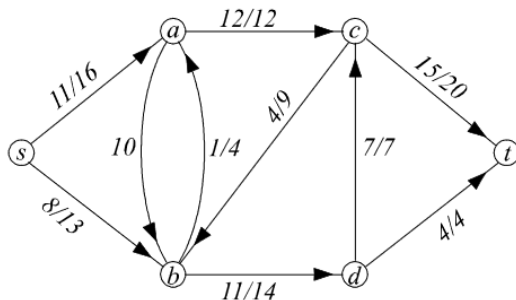
(warunek ograniczenia przepustowości)

- $$f(u, v) = -f(v, u)$$

(warunek skośnej symetrii)

- $$\sum_{v \in V} f(u, v) = 0$$

(warunek zachowania przepływu)

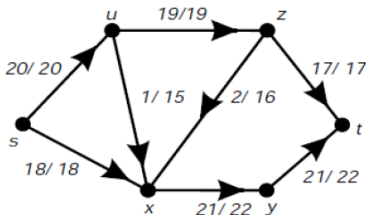


$$\begin{aligned} c(s, a) &= 16, & c(a, s) &= 0, & c(a, b) &= 10, & c(b, a) &= 4, & c(s, c) &= c(c, s) = 0, \\ f(s, a) &= 11, & f(a, s) &= -11, & f(a, b) &= -1, & f(b, a) &= 1, & f(s, c) &= f(c, s) = 0. \end{aligned}$$

Definicja: Wartość przepływu

Wartość przepływu f , oznaczamy $|f|$ i definiujemy jako sumaryczną wielkość przepływu wypływającego ze źródła s z wszystkimi krawędziami, tzn.

$$|f| = \sum_{u \in V} f(s, u).$$



$|f| = 38$

Niech f będzie przepływem w sieci (G, s, t, c) wtedy

- 1 $f(v, v) = 0$ dla każdego $v \in V$,
- 2 dla dowolnych wierzchołków $u, v \in V$ takich, że $(u, v), (v, u) \notin E$ mamy $f(u, v) = f(v, u) = 0$,
- 3 dla dowolnego wierzchołka $u \in V \setminus \{s, t\}$, mamy

$$\sum_{v \in V: f(u,v) > 0} f(u,v) = - \sum_{v \in V: f(u,v) < 0} f(u,v).$$

Przyjmijmy następującą notację sumacyjną, w której argument funkcji $f: X \rightarrow Y$ może być podzbiorem A dziedziny X . W takim wypadku przyjmujemy, że wartością funkcji f jest suma jej wartości dla zbioru A , tzn.

$$f(X, Y) = \sum_{x \in X} \sum_{y \in Y} f(x, y).$$

Używając tej notacji, możemy zapisać warunek zachowania przepływu dla wierzchołka v jako $f(v, V) = 0$.

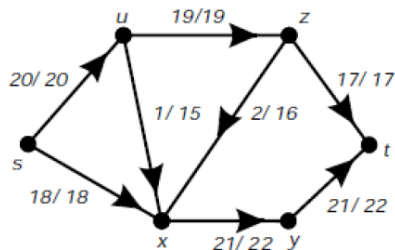
Przepływy w sieciach

Własność

Niech f będzie przepływem w sieci (G, s, t, c) wtedy

- ❶ dla dowolnego $X \subseteq V$ mamy $f(X, X) = 0$,
- ❷ dla dowolnych $X, Y \subseteq V$ mamy $f(X, Y) = -f(Y, X)$,
- ❸ dla dowolnych $X, Y, Z \subseteq V$, mamy
$$f(X \cup Y, Z) = f(X, Z) + f(Y, Z) - f(X \cap Y, Z)$$
oraz
$$f(X, Y \cup Z) = f(X, Y) + f(X, Z) - f(X, Y \cap Z),$$
- ❹ $|f| = f(s, V) = f(V, t)$.

Przykład



	c	f		f
(s, u)	20	20	(u, s)	-20
(s, x)	18	18	(x, s)	-18
(u, z)	19	19	(z, u)	-19
(u, x)	15	1	(x, u)	-1
(z, x)	16	2	(x, z)	-2
(x, y)	22	21	(y, x)	-21
(y, t)	22	21	(t, y)	-21
(z, t)	17	17	(t, z)	-17

$$\sum_{v \in V} f(z, v) = f(z, t) + f(z, u) + f(z, x) = 17 - 19 + 2 = 0$$

Problem maksymalnego przepływu

Problem maksymalnego przepływu

Dla danej sieci (G, s, t, c) znaleźć taki przepływ f , którego wartość będzie maksymalna.

$$|f| = \sum_{u \in V} f(s, u) \rightarrow \max.$$

Taki przepływ nazywamy **maksymalnym przepływem w sieci** (G, s, t, c) .

Definicja

Niech (G, s, t, c) będzie siecią, f przepływem w tej sieci. **Przepustowością residualną pary wierzchołków** (u, v) sieci (G, s, t, c) nazywamy liczbę

$$c_f(u, v) = c(u, v) - f(u, v).$$

Problem maksymalnego przepływu

Definicja: Sieć residualna

Siecią residualną dla sieci (G, s, t, c) indukowaną przez przepływ f nazywamy sieć (G_f, s, t, c_f) , w której $G_f = \langle V, E_f \rangle$, przy czym

$$E_f = \{(u, v) \in V \times V : c_f(u, v) > 0\}.$$

Krawędzie sieci residualnej nazywamy **krawędziami residualnymi**.

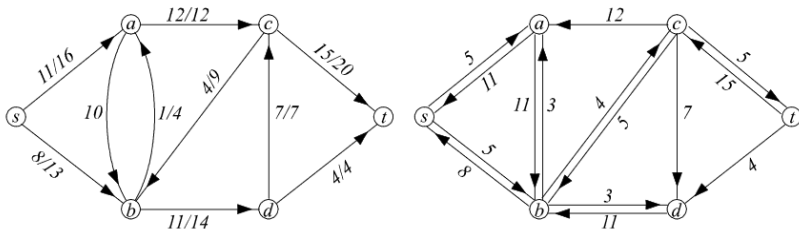
Sieć residualna jest siecią przeplywową.

Definicja: Ścieżka powiększająca

Dla danej sieci (G, s, t, c) i przepływu f **ścieżką powiększającą p** nazywamy każdą ścieżkę ze źródła s do ujścia t w sieci residualnej (G_f, s, t, c_f)

Przykład

Poniżej przedstawiona jest sieć z przepływem f oraz sieć residualna indukowana przez f .



Przepustowość residualna krawędzi pokazuje o ile można zwiększyć przepływ przez tę krawędź w sieci oryginalnej lub o ile można zmniejszyć przepływ przez krawędź przeciwną. Przykładowo:

- $c_f(s, a) = 5$ - w sieci oryginalnej można zwiększyć przepływ przez (s, a) o 5,
- $c_f(a, s) = 11$ - w sieci oryginalnej można zmniejszyć przepływ przez (s, a) o 11,
- $c_f(a, b) = 11$ - w sieci oryginalnej można zwiększyć przepływ przez (a, b) o 10 oraz zmniejszyć przepływ przez (b, a) o 1.

Problem maksymalnego przepływu

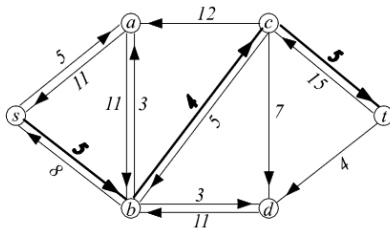
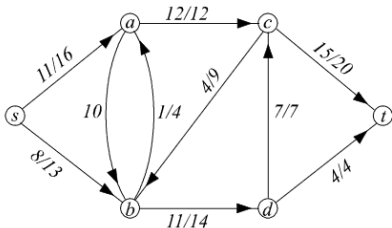
Twierdzenie

Jeżeli f jest przepływem w sieci (G, s, t, c) , zaś f' przepływem w sieć residualnej (G_f, s, t, c_f) , to funkcja $f + f'$ jest przepływem w sieci (G, s, t, c) oraz $|f + f'| = |f| + |f'|$.

Twierdzenie

Jeżeli f jest przepływem w sieci (G, s, t, c) , oraz p jest ścieżką powiększającą, to funkcja $f + f^p$ jest przepływem w sieci (G, s, t, c) takim, że $|f + f^p| > |f|$.

O przepływie $f + f^p$ mówimy, że jest to przepływ powiększony wzdłuż ścieżki p .



Metoda Forda-Fulkersona

Poprzednie twierdzenie jest podstawą działania **metody Forda-Fulkersona**, pierwszego opisanego sposobu szukania maksymalnego przepływu.

Ford-Fulkerson(N)

$f := 0$

while istnieje ścieżka powiększająca w N_f **do**

$p :=$ ścieżka powiększająca

$f := f + f^p$

$f_{\max} := f$

gdzie N to sieć oryginalna, a N_f to sieć residualna indukowana przez f .

Twierdzenie

Jeżeli przepustowość w sieci (G, s, t, c) przyjmuje tylko wartości wymierne, to wynikiem działania procedury Ford-Fulkerson jest przepływ maksymalny.

Przepływ przez przekrój

Definicja: Przekrój

Niech (G, s, t, c) będzie siecią przepływową. Dowolną parę zbiorów (A, B) takich, że $A \subset V, B = V \setminus A, s \in A$ i $t \in B$ nazywamy **przekrojem** w tej sieci. **Przepustowością przekroju** (A, B) nazywamy liczbę $c(A, B)$.

Definicja: Przekrój minimalny

Przekrojem minimalnym nazywamy przekrój o najmniejszej przepustowości. Ponieważ liczba wszystkich przekrojów jest skończona, więc przekrój minimalny zawsze istnieje.

Definicja: Przepływ przez przekrój

Jeżeli f jest przepływem w (G, s, t, c) , to liczbę $f(A, B)$ nazywamy **przepływem przez przekrój** (A, B) .

Przepływ przez przekrój

Własność

Dla dowolnego przepływu f i dowolnego przekroju (A, B) zachodzi

$$|f| = f(A, B) \leq c(A, B).$$

Twierdzenie

Niech f będzie przepływem w sieci (G, s, t, c) . Następujące warunki są równoważne:

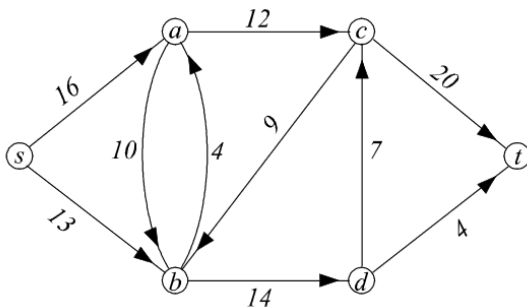
- 1 Przepływ f jest maksymalny.
- 2 Sieć residualna (G_f, s, t, c_f) nie zawiera ścieżek powiększających.
- 3 Istnieje przekrój (A, B) taki, że $|f| = c(A, B)$.

Twierdzenie minimaksowe

Twierdzenie: O maksymalnym przepływie i minimalnym przekroju

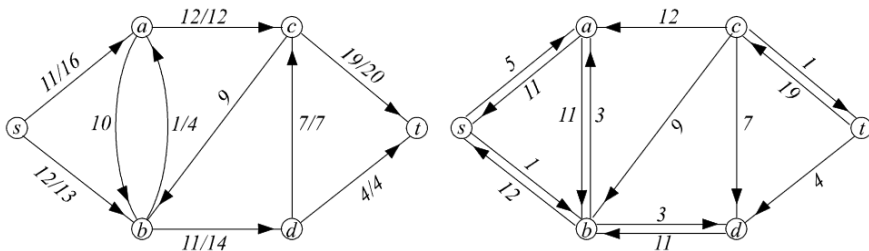
Wartość maksymalnego przepływu w sieci przepływowej jest równa przepustowości minimalnego przekroju w tej sieci.

Rozważmy sieć:



Przykład

Znaleźliśmy przepływ f generujący następującą sieć residualną:

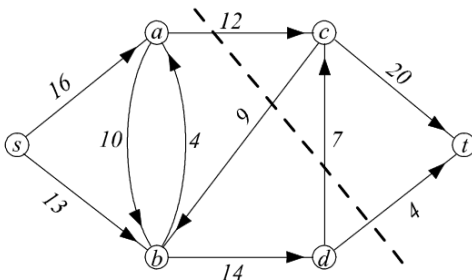


Zauważmy, że w sieci residualnej nie istnieje ścieżka powiększająca, więc f jest przepływem minimalnym.

Przykład

Wyznamy teraz przekrój minimalny, wystarczy przyjąć jako zbiór A wszystkie wierzchołki osiągalne ze źródła w sieci residualnej. May więc

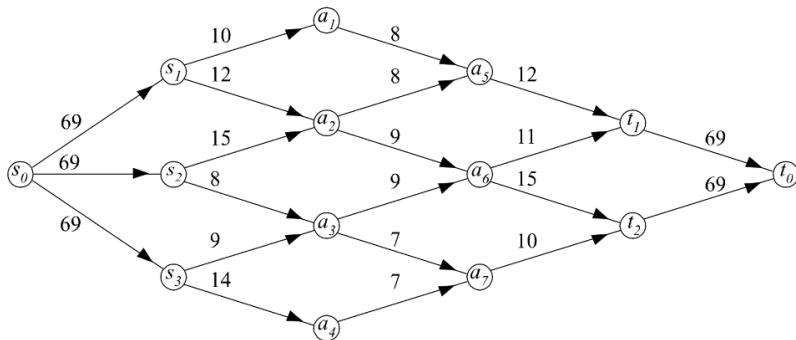
$$A = \{s, a, b, d\}, \quad B = \{c, t\}.$$



Ponieważ $c(A, B) = 12 + 7 + 4 = 23 = |f|$, więc na podstawie twierdzenia przekrój (A, B) jest przekrojem minimalnym.

Sieci przepływowe z wieloma źródłami i ujściami

Problem szukania maksymalnego przepływu w sieciach przepływowych z wieloma źródłami i ujściami można sprowadzić do szukania maksymalnego przepływu w pewnej sieci z jednym źródłem i jednym ujściem. Sieć taką stworzymy poprzez dodanie wierzchołka s_0 zwanego **superźródłem**, wierzchołka t_0 zwanego **superujściem** oraz krawędzi łączących superźródło ze źródłami i krawędzi łączących ujścia z superujściem.



Drzewo najkrótszych ścieżek

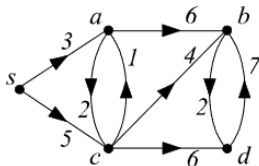
Definicja

Niech $G = \langle V, E \rangle$ będzie digrafem z wagami, zaś s ustalonym wierzchołkiem w G . **Drzewem najkrótszych ścieżek o korzeniu s** nazywamy podgraf $G_0 = \langle V_0, E_0 \rangle$ digrafu G taki, że

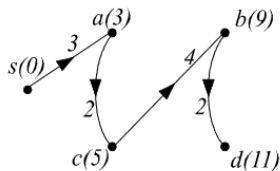
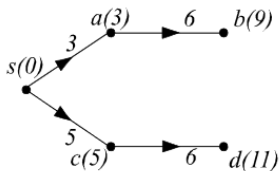
- V_0 składa się ze wszystkich wierzchołków osiągalnych z s .
- G_0 jest drzewem skierowanym z korzeniem s .
- Dla dowolnego wierzchołka $a \in V_0$, jedyna ścieżka z s do a w G_0 jest najkrótszą ścieżką z s do a w G .

Przykład

Rozważmy ważony graf skierowany:



Wagi najkrótszych ścieżek z wierzchołka s wynoszą: $m(s, s) = 0$, $m(s, a) = 3$, $m(s, b) = 9$, $m(s, c) = 5$, $m(s, d) = 11$. Drzewami najkrótszych ścieżek są np.



Algorytmy

- Omówimy dwa algorytmy rozwiązujące problem najkrótszych ścieżek z jednym źródłem w prostym digrafie z wagami: **algorytm Dijkstry** i **algorytm Bellmana-Forda**.
- Istota działania obu algorytmów jest podobna. Dla ustalonego wierzchołka s , budują one dwa wektory $p = [p(u)]_{u \in V}$ i $d = [d(u)]_{u \in V}$.
- Pierwszy jest wektorem poprzedników drzewa najkrótszych ścieżek o korzeniu s . Jego znajomość pozwala na wyznaczenie najkrótszych ścieżek z s do pozostałych wierzchołków.
- Drugi wektor przechowuje wagi najkrótszych ścieżek o początku s .
- Wektory są stopniowo poprawiane w trakcie analizy kolejnych fragmentów grafu.

Initializacja

- Na początku jedyną znaną nam najkrótszą ścieżką jest ścieżka trywialna z s do s . Zatem przyjmujemy $d(s) = 0$ i $p(s) = \emptyset$ (s nie ma na tej ścieżce poprzednika). Dla pozostałych wierzchołków najkrótsze ścieżki nie są znane. Przyjmujemy więc $p(u) = \emptyset$ i $d(u) = \infty$ dla $u \neq s$.
- Ustawienie wartości początkowych jest realizowane przez procedurę Init:

```
Init( $G, s$ )  
for każdy  $u \in V_G$  do  
     $d(u) := \infty$   
     $p(u) := \emptyset$   
 $d(s) := 0$ 
```

Relaksacja

- Po zainicjowaniu algorytmu rozpoczynają poprawianie wektorów p i d .
- Polega to na badaniu kolejnych krawędzi i sprawdzaniu czy uwzględnienie badanej krawędzi nie zmniejszy wagi którejś ze znalezionych do tej pory najkrótszych ścieżek.
- Operacja ta nazywana jest relaksacją i realizowana jest przez procedurę Relax:

Relax(a, b)

if $d(a) + w(a, b) < d(b)$ **then**

$d(b) := d(a) + w(a, b)$

$p(b) := a$

Algorytm Dijkstry

Niech (G, w) będzie prostym grafem skierowanym z wagami, s dowolnym wierzchołkiem G . Algorytm Dijkstry wykorzystuje zmienną V_0 , w której pamiętany jest zbiór przetworzonych wierzchołków czyli takich, dla których znaleźliśmy rzeczywiste najkrótsze ścieżki:

```

Dijkstra( $G, w, s$ )
Init( $G, s$ )
 $V_0 := \emptyset$ 
while  $V_0 \neq V$  do
    znajdź  $a \in V \setminus V_0$  o minimalnej etykiecie  $d(a)$ 
     $V_0 := V_0 \cup \{a\}$ 
    for każdy wierzchołek  $b \in V \setminus V_0$  do
        if  $(a, b) \in E$  then Relax( $a, b$ )
  
```

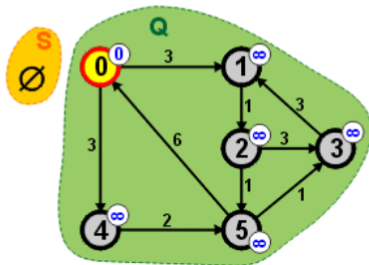
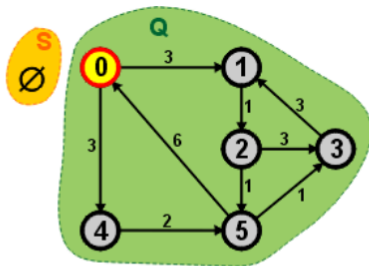
Algorytm Dijkstry

Twierdzenie

Niech (G, w) będzie prostym grafem skierowanym z wagami oraz s wierzchołkiem G . Wynikiem działania procedury Dijkstra są wektory d, p takie, że

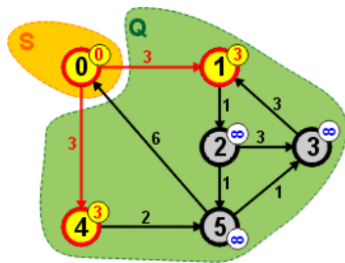
- 1 $d(a) = m(s, a)$ dla $a \in V$, czyli d składa się z wag najkrótszych ścieżek o początku s .
- 2 Wektor p jest wektorem poprzedników w G .
- 3 G_p jest drzewem najkrótszych ścieżek o korzeniu s .

Algorytm Dijkstry - Przykład

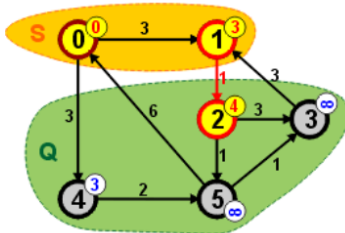


u	0	1	2	3	4	5
$d[u]$	0	∞	∞	∞	∞	∞
$p[u]$	-1	-1	-1	-1	-1	-1

Algorytm Dijkstry - Przykład

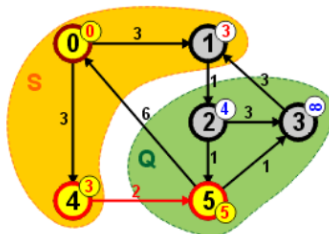


u	0	1	2	3	4	5
$d[u]$	0	3	∞	∞	3	∞
$p[u]$	-1	0	-1	-1	0	-1

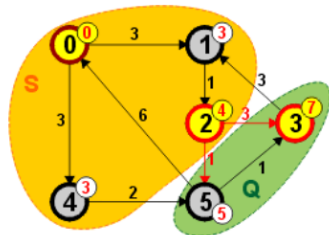


u	0	1	2	3	4	5
$d[u]$	0	3	4	∞	3	∞
$p[u]$	-1	0	1	-1	0	-1

Algorytm Dijkstry - Przykład

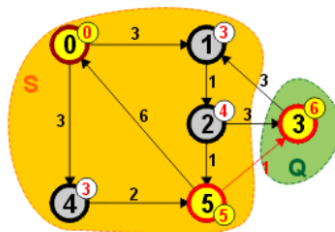


u	0	1	2	3	4	5
$d[u]$	0	3	4	∞	3	5
$p[u]$	-1	0	1	-1	0	4

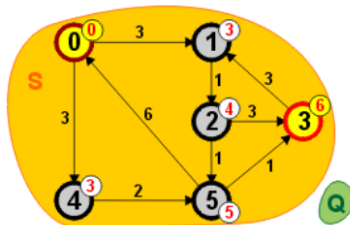


u	0	1	2	3	4	5
$d[u]$	0	3	4	7	3	5
$p[u]$	-1	0	1	2	0	4

Algorytm Dijkstry - Przykład



u	0	1	2	3	4	5
$d[u]$	0	3	4	6	3	5
$p[u]$	-1	0	1	5	0	4



u	0	1	2	3	4	5
$d[u]$	0	3	4	6	3	5
$p[u]$	-1	0	1	5	0	4

Algorytm Bellmana-Forda

Niech (G, w) będzie prostym grafem skierowanym z wagami, s dowolnym wierzchołkiem G oraz niech n oznacza ilość wierzchołków tego grafu. Algorytm Bellmana-Forda polega na wykonaniu $n - 1$ relaksacji wszystkich krawędzi grafu:

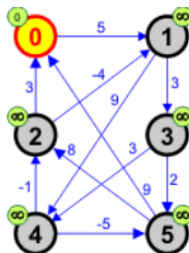
```

Bellman-Ford( $G, w, s$ )
  Init( $G, s$ )
  for  $i := 1$  to  $n - 1$  do
    for każda krawędź  $(a, b) \in E$  do
      Relax( $a, b$ )
  
```

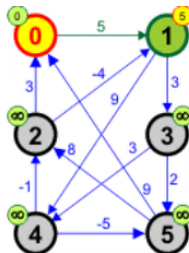
Twierdzenie

Wynikiem działania procedury Bellman-Ford są wektory d, p takie, że

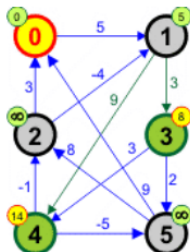
- ❶ $d(a) = m(s, a)$ dla $a \in V$, czyli d składa się z wag najkrótszych ścieżek o początku s .
- ❷ Wektor p jest wektorem poprzedników w G .
- ❸ G_p jest drzewem najkrótszych ścieżek o korzeniu s .



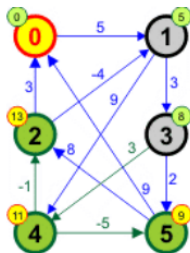
i	0	1	2	3	4	5
$d[i]$	0	∞	∞	∞	∞	∞
$p[i]$	-1	-1	-1	-1	-1	-1



i	0	1	2	3	4	5
$d[i]$	0	5	∞	∞	∞	∞
$p[i]$	-1	0	-1	-1	-1	-1

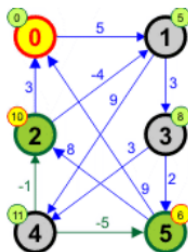


i	0	1	2	3	4	5
$d[i]$	0	5	∞	8	14	∞
$p[i]$	-1	0	-1	1	1	-1

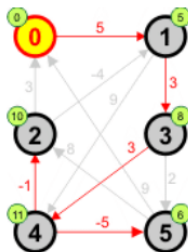


i	0	1	2	3	4	5
$d[i]$	0	5	13	8	11	9
$p[i]$	-1	0	4	1	3	4

Algorytm Bellmana-Forda - Przykład



i	0	1	2	3	4	5
$d[i]$	0	5	10	8	11	6
$p[i]$	-1	0	4	1	3	4



i	0	1	2	3	4	5
$d[i]$	0	5	10	8	11	6
$p[i]$	-1	0	4	1	3	4

Dijkstra vs Bellman-Ford

- Algorytm Dijkstry można stosować tylko przy założeniu że wagi w grafie są nieujemne.
- Algorytm Dijkstry jest szybszy niż algorytm Bellmana-Forda.
- Złożoność obliczeniowa algorytmu Bellmana-Forda: $O(|V| \cdot |E|)$.
- Złożoność obliczeniowa algorytmu Dijkstry: $O(|V|^2)$. Jeżeli stosujemy go do grafów wystarczająco rzadkich, to przy odpowiedniej implementacji można uzyskać $O(\frac{|V|^2}{\ln V})$.
- Algorytm Bellmana-Forda można stosować również do grafów z wagami ujemnymi (o ile nie istnieje w nim cykl ujemny).