4b. Grafy skierowane: Drogi minimalne

Grzegorz Kosiorowski

Uniwersytet Ekonomiczny w Krakowie

Grafy skierowane - uwaga początkowa

Znajdowanie optymalnych tras przejścia to typowe zagadnienie związane z grafami skierowanymi. Jednak wszystkie tego typu algorytmy można też realizować dla grafów nieskierowanych. Wystarczy za każdym razem traktować graf nieskierowany jako graf skierowany, w którym każdą krawędź grafu nieskierowanego zastępujemy dwiema krawędziami, prowadzącymi w przeciwne strony, o końcach w tych samych wierzchołkach.
Zagadnienie wyznaczania dróg minimalnych dla niekierowanych grafów bez wag rozważaliśmy już w rozdziale o przeszukiwaniach grafu, a odpowiedni algorytm opierał się na przeszukiwaniu wszerz.

Motywacja

Typowym zagadnieniem związanym z grafami (zwłaszcza skierowanymi i z wagami) jest znalezienie minimalnej drogi (czyli drogi o minimalnej wadze) z jednego wierzchołka do drugiego.

- Znajdowanie najkrótszych odległościowo lub czasowo tras w sieci dróg np. nawigacja samochodowa.
- Procesy produkcyjne: trasa o najmniejszym koszcie w grafie, który przedstawia alternatywne sposoby dotarcia do wyznaczonego celu.
- Przesyłanie infomacji drogą najmniej obciążającą sieć.

Algorytm Dijkstry - wstęp

Wyznaczanie dróg minimalnych algorytmem przeszukiwania wszerz ma ograniczony zakres stosowalności. Nieczęsto się zdarza, by w rzeczywistym problemie wszystkie krawędzie grafu (drogi między skrzyżowaniami, procesy pomiędzy kolejnymi etapami produkcji) miały tę samą wagę (tj. odległość, koszt, czas konieczny do przebycia itp.).
Dany jest graf prosty (znów obecność pętli i krawędzi wielokrotnych nic nie zmienia, ale zapis algorytmu jest prostszy bez nich), skierowany $G = (V, E)$. Każdej jego krawędzi e przypisujemy wagę $W(e)$.
Jak poprzednio, zaczynamy od ponumerowania w dowolny sposób wierzchołków takiego grafu $V = \{1, \ldots, n\}$. Poszukujemy drogi minimalnej (czyli o najmniejszej sumie wag krawędzi) z wierzchołka 1 do każdego innego wierzchołka j. Problem ten rozwiązuje algorytm Dijkstry (ze wskaźnikami).

Algorytm Dijkstry ze wskaźnikami - część I

Algorytm Dijkstry

Dane: Graf $G = (V(G), E(G))$ prosty skierowany ze zbiorem wierzchołków $V(G) = \{1, \ldots, n\}$ i funkcją W wag („długości") krawędzi o wartościach nieujemnych (jeśli krawędź (v, w) nie istnieje to podstawiamy $W(v, w) = \infty$).
Zmienne: L i V - zbiory wierzchołków , d, p - funkcje.

- I. $L := \{1\}$, $V := \{2, \ldots n\}$.
- II. Dla $i \in V$ wykonuj:
- II.1. $d(i) := W(1, i)$.
- II.2. Jeśli $W(1, i) = \infty$ to $p(i) := 0$, w innym wypadku $p(i) := 1$.

Algorytm Dijkstry ze wskaźnikami - część II

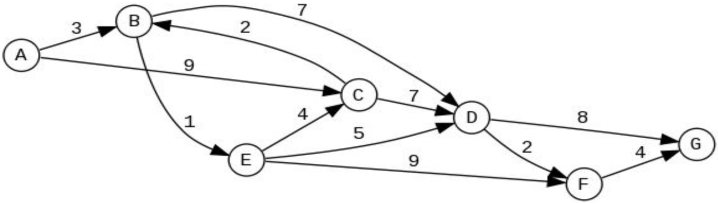
Algorytm Dijkstry

- III. Dopóki $V \setminus L \neq \emptyset$, wykonuj:
- III.1. wybierz $k \in V \setminus L$ takie, że $d(k)$ przyjmuje najmniejszą możliwą wartość.
- III.2. dołącz k do zbioru L.
- III.3. dla każdego $j \in V \setminus L$ wykonuj: jeśli $d(j) > d(k) + W(k,j)$ to zastąp $d(j)$ sumą $d(k) + W(k,j)$ i zastąp $p(j)$ liczbą k.
- **Rezultat**: $d(j)$ to minimalna waga drogi z 1 do j, $p(j)$ to wskaźnik, który pokazuje poprzedni etap drogi z 1 do j.

Algorytm Dijkstry - wstępne uwagi

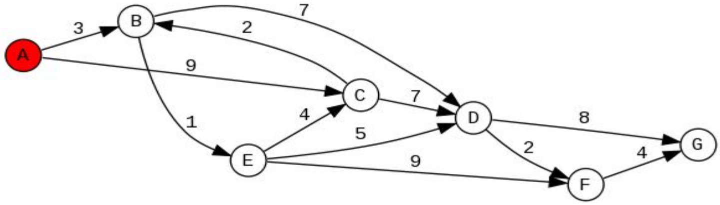
Przebieg minimalnej drogi odczytujemy ze wskaźników, jak w wypadku przeszukiwania wszerz.
Potocznie: W każdym kroku zawsze patrzymy na ostatni wierzchołek (nazwijmy go k), który dołączyliśmy do zbioru L, czyli do zbioru wierzchołków, dla których już wszystko obliczyliśmy. Następnie sprawdzamy, do jakich wierzchołków, które nie są w zbiorze L prowadzą krawędzie z naszego wierzchołka i dla każdego z tych wierzchołków sprawdzamy, czy droga, która prowadzi do niego przez k nie jest krótsza od dotychczas znalezionych dróg. Jeśli tak, zapisujemy jej nową wagę. Gdy się skończą wierzchołki do zbadania, wybieramy ten, który ma w tym momencie najmniejszą odległość od startu i jest poza L i on się staje nowym wierzchołkiem k.
Powtarzamy, aż się skończą wierzchołki.

Algorytm Dijkstry - przykład



Spróbujemy zastosować algorytm Dijkstry dla powyższego grafu skierowanego. Przedstawię tutaj sposób postępowania przyjęty w ramach tego kursu - obowiązujący na sprawdzianie/egzaminie. Algorytm zaczynamy od ponumerowania wierzchołków w kolejności alfabetycznej: A-1, B-2 itd... Będziemy obliczać minimalną wagę drogi pomiędzy A i każdym z pozostałych wierzchołków.

Algorytm Dijkstry - przykład

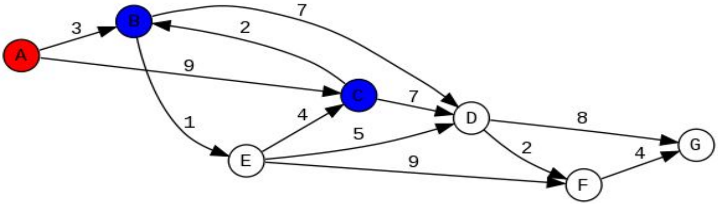


Zaczynamy od wierzchołka A. Na zielono będę zaznaczać wierzchołki będące w L od poprzedniego obiegu pętli, na czerwono wierzchołek ostatnio dołączony do L (w algorytmie - k), a na niebiesko wierzchołki, których dane właśnie korygujemy.

Działanie algorytmu Dijkstry zapisujemy w takiej tabeli:

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
1							
2							

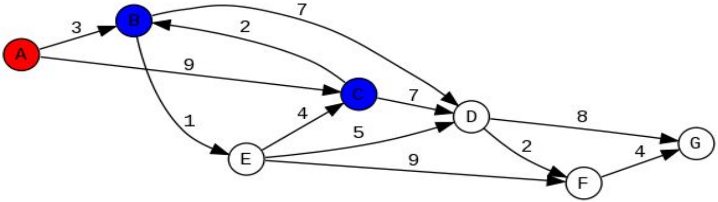
Algorytm Dijkstry - przykład - krok 1



Na początku wszystkie wagi (odległości) są ustalone na $+\infty$, a wskaźniki mają wartość 0, póki te odległości się nie zmniejszą. Badamy dwa wierzchołki, do których prowadzą krawędzie z A. Jako, że nowoznalezione drogi (AB i AC) są krótsze od dotychczasowych ($3 < \infty$ i $9 < \infty$), możemy zapisać ich długości w tabeli:

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
1	A	3A	9A	$+\infty$	$+\infty$	$+\infty$	$+\infty$
2							

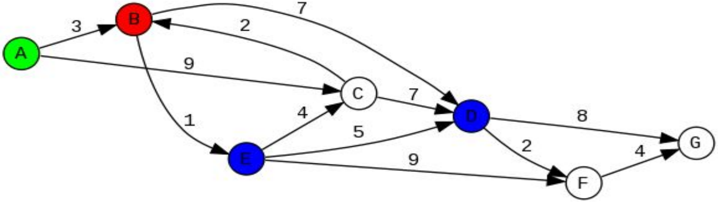
Algorytm Dijkstry - przykład - krok 1



Warto zauważyć, że poza nowymi odległościami, w tabeli notujemy jako wskaźnik - wierzchołek, z którego nowa, krótsza droga dotarła do danego. Teraz, do zbioru L dołączamy wierzchołek, który ma najmniejszą wartość funkcji d z jeszcze niedołączonych (czyli w tym wypadku B).

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
1	A	3A	9A	$+\infty$	$+\infty$	$+\infty$	$+\infty$
2	A,B						

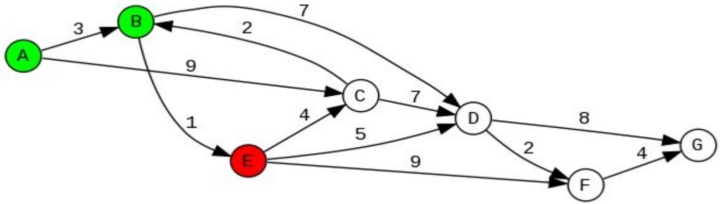
Algorytm Dijkstry - przykład - krok 2



Badamy teraz wierzchołki do których prowadzą krawędzie z B. Znajdujemy nowe wartości $d(D)$ i $d(E)$ dodając wagi krawędzi BD i BE do $d(B)$ i otrzymujemy propozycje $d(D) = 7 + 3 = 10$, $d(E) = 1 + 3 = 4$. Te wartości są mniejsze niż dotychczasowe $(+\infty)$, więc je zmieniamy, dopisując wskaźnik B.

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
1	A	3A	9A	$+\infty$	$+\infty$	$+\infty$	$+\infty$
2	A,B	3A	9A	10B	4B	$+\infty$	$+\infty$

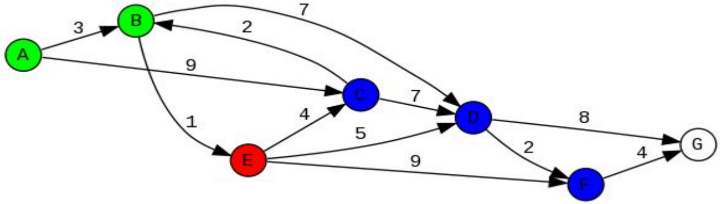
Algorytm Dijkstry - przykład - krok 2



Teraz, do zbioru L dołączamy wierzchołek, który ma najmniejszą wartość funkcji d z jeszcze niedołączonych (czyli w tym wypadku E). Zauważmy, że wśród potencjalnych kandydatów był wierzchołek C, choć nie zmienialiśmy jego odległości w tym kroku.

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
1	A	3A	9A	$+\infty$	$+\infty$	$+\infty$	$+\infty$
2	A,B	3A	9A	10B	4B	$+\infty$	$+\infty$

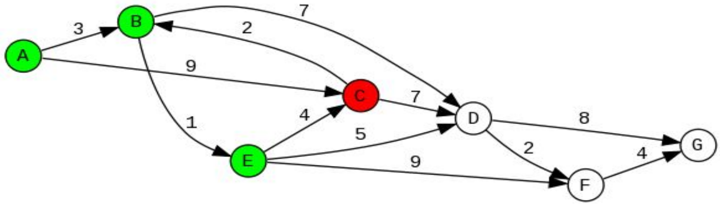
Algorytm Dijkstry - przykład - krok 3



Badamy teraz wierzchołki do których prowadzą krawędzie z E. Znajdujemy nowe wartości $d(C)$, $d(D)$ i $d(F)$ dodając wagi krawędzi EC, ED i EF do $d(E)$ i otrzymujemy propozycje $d(C) = 4 + 4 = 8$, $d(D) = 5 + 4 = 9$, $d(F) = 9 + 4 = 13$. Te wartości są mniejsze niż dotychczasowe (9,10 i $+\infty$), więc je zmieniamy, dopisując wskaźnik E.

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
2	A,B	3A	9A	10B	4B	$+\infty$	$+\infty$
3	A,B,E	3A	8E	9E	4B	13E	$+\infty$

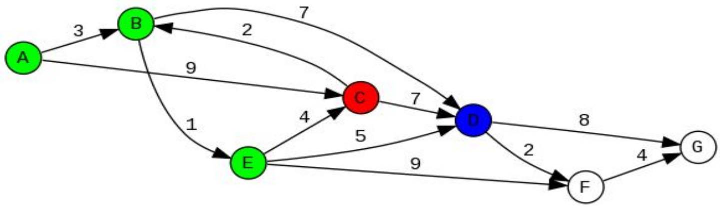
Algorytm Dijkstry - przykład - krok 3



Teraz, do zbioru *L* dołączamy wierzchołek, który ma najmniejszą wartość funkcji *d* z jeszcze niedołączonych (czyli w tym wypadku C).

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
3	A,B,E	3A	8E	9E	4B	13E	$+\infty$
4	A,B,E,C	3A	8E		4B		

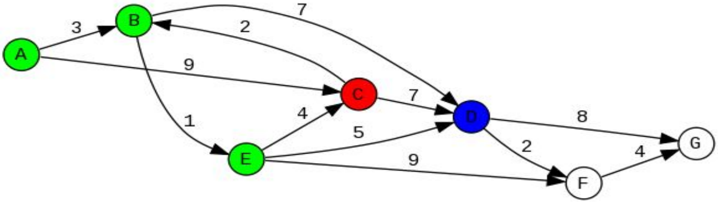
Algorytm Dijkstry - przykład - krok 4



Badamy teraz wierzchołki do których prowadzą krawędzie z C. Znajdujemy nową wartość $d(D)$ dodając wagę krawędzi CD do $d(C)$ i otrzymujemy propozycję $d(D) = 7 + 8 = 15$. Ta wartość jest WIĘKSZA niż dotychczasowa (9), więc jej nie zmieniamy, jak i wskaźnika.

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
3	A,B,E	3A	8E	9E	4B	13E	$+\infty$
4	A,B,E,C	3A	8E	9E	4B	13E	$+\infty$

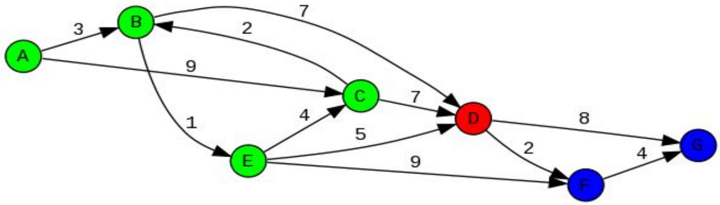
Algorytm Dijkstry - przykład - krok 4



Teraz, do zbioru L dołączamy wierzchołek, który ma najmniejszą wartość funkcji d z jeszcze niedołączonych (czyli w tym wypadku D).

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
4	A,B,E,C	3A	8E	9E	4B	13E	$+\infty$
5	A,B,E,C,D	3A	8E	9E	4B		

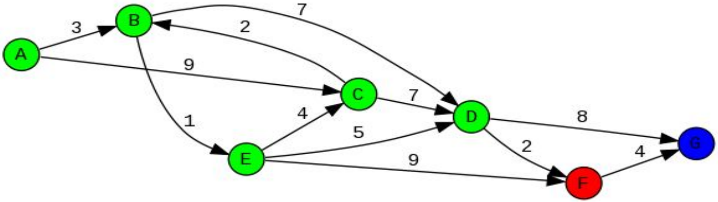
Algorytm Dijkstry - przykład - krok 5



Badamy teraz wierzchołki do których prowadzą krawędzie z D. Znajdujemy nowe wartości $d(F)$ i $d(G)$ dodając wagi krawędzi DF i DG do $d(D)$ i otrzymujemy propozycje $d(F) = 2 + 9 = 11$, $d(G) = 8 + 9 = 17$. Te wartości są mniejsze niż dotychczasowe (13 i $+\infty$), więc je zmieniamy, dopisując wskaźnik D.

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
4	A,B,E,C	3A	8E	9E	4B	13E	$+\infty$
5	A,B,E,C,D	3A	8E	9E	4B	11D	17D

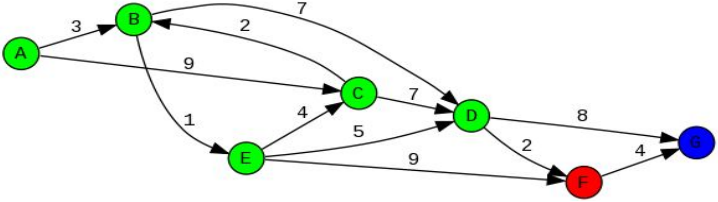
Algorytm Dijkstry - przykład - krok 5



Teraz, do zbioru *L* dołączamy wierzchołek, który ma najmniejszą wartość funkcji *d* z jeszcze niedołączonych (czyli w tym wypadku F).

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G))
5	A,B,E,C,D	3A	8E	9E	4B	11D	17D
6	A,B,E,C,D,F	3A	8E	9E	4B	11D	

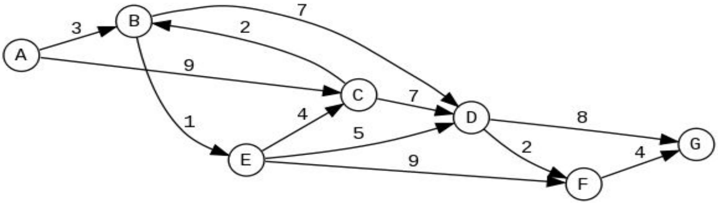
Algorytm Dijkstry - przykład - krok 6



W ostatnim kroku zostało tylko sprawdzenie, czy droga przez wierzchołek F zmniejsza odległość wierzchołka G od A. Znajdujemy nową wartość $d(G)$ dodając wagę krawędzi FG do $d(F)$ i otrzymujemy propozycję $d(G) = 11 + 4 = 15$. Te wartość jest mniejsza niż dotychczasowa (17), więc ją zmieniamy, dopisując wskaźnik F, co kończy działanie algorytmu.

Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G))
5	A,B,E,C,D	3A	8E	9E	4B	11D	17D
6	A,B,E,C,D,F	3A	8E	9E	4B	11D	15F

Algorytm Dijkstry - wynik



Wynikiem algorytmu jest tabela:

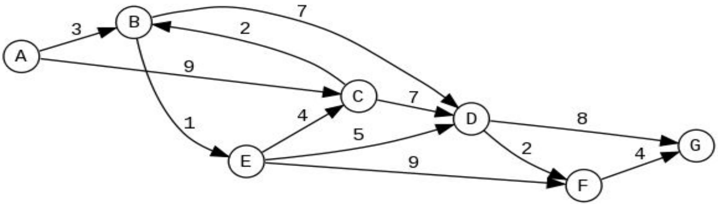
Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
1	A	3A	9A	$+\infty$	$+\infty$	$+\infty$	$+\infty$
2	A,B	3A	9A	10B	4B	$+\infty$	$+\infty$
3	A,B,E	3A	8E	9E	4B	13E	$+\infty$
4	A,B,E,C	3A	8E	9E	4B	13E	$+\infty$
5	A,B,E,C,D	3A	8E	9E	4B	11D	17D
6	A,B,E,C,D,F	3A	8E	9E	4B	11D	15F

Algorytm Dijkstry - wynik

Ze względu na jasny kontekst algorytmu, możemy zapisać tabelę krócej. Nie ma konieczności zapisywania, że obliczamy wagę drogi do wierzchołka i wskaźnik wierzchołka, wystarczy odpowiednią kolumnę oznaczyć wierzchołkiem. Warto też zauważyć, że od momentu, gdy wierzchołek trafi do zbioru L, jego parametry w algorytmie się nie zmieniają, więc nie trzeba ich przepisywać:

Etap	zbiór L	B	C	D	E	F	G
1	A	3A	9A	$+\infty$	$+\infty$	$+\infty$	$+\infty$
2	A,B	-	9A	10B	4B	$+\infty$	$+\infty$
3	A,B,E	-	8E	9E	-	13E	$+\infty$
4	A,B,E,C	-	-	9E	-	13E	$+\infty$
5	A,B,E,C,D	-	-	-	-	11D	17D
6	A,B,E,C,D,F	-	-	-	-	-	15F

Algorytm Dijkstry - wynik



Nr etapu	zbiór L	d,p(B)	d,p(C)	d,p(D)	d,p(E)	d,p(F)	d,p(G)
6	A,B,E,C,D,F	3A	8E	9E	4B	11D	15F

Z tabeli, dzięki wskaźnikom, możemy odczytać nie tylko wagę minimalnej drogi do danego punktu, ale i jej przebieg. Na przykład, by odczytać drogę od A do G odczytujemy: $p(G) = F$, $p(F) = D$, $p(D) = E$, $p(E) = B$, $p(B) = A$ i piszemy w odwrotnej kolejności: $ABEDFG$.

Uwagi dotyczące algorytmu Dijkstry i jego zapisu

- Od momentu, gdy jakiś wierzchołek zostanie włączony do zbioru L, jego wartości d i p się nie zmieniają.
- W każdym kroku ustalamy odległość jednego wierzchołka, więc algorytm (i jego tabelka) powinien mieć tyle kroków, ile jest wierzchołków (poza startowym) w tej samej składowej spójnej grafu. Jeśli jest więcej niż jeden wierzchołek poza L w tej samej, najmniejszej odległości od punktu startowego, wybieramy dowolny.
- Dla grafu $G = (V, E)$, czas działania algorytmu Dijkstry jest $O(|V|^2)$.

Uwagi dotyczące algorytmu Dijkstry i jego zapisu

- Jeżeli na koniec algorytmu Dijkstry dla jakiegoś wierzchołka v zachodziłoby $d(v) = \infty$, oznaczałoby to, że nie istnieje droga z początkowego wierzchołka do wierzchołka
- Jeśli chcielibyśmy znaleźć drogi minimalne od każdego wierzchołka do każdego innego można po prostu użyć algorytmu Dijkstry dla każdego z wierzchołków z osobna. Lepszy sposób wykonania takiego zadania proponuje algorytm Warshalla (ze wskaźnikami).
- W kolejnej prezentacji zmodyfikujemy algorytm Dijkstry tak, by wyznaczał drogi maksymalne, czyli takie o możliwie największej wadze.
- Jeśli w grafie dopuszczamy krawędzie o wagach ujemnych, w celu znalezienia drogi minimalnej algorytm Dijkstry trzeba zastąpić (znacznie wolniejszym) algorytmem Bellmana-Forda.