

Porównanie algorytmów wyszukiwania najkrótszych ścieżek między wierzchołkami grafu.

Daniel Golubiewski

Instytut Informatyki

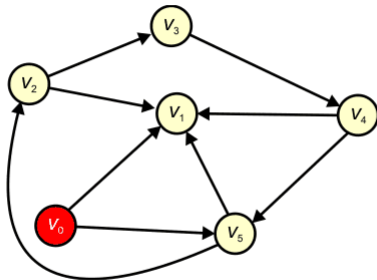
22 listopada 2015

Algorytm DFS "w głąb"

Algorytm przejścia/przeszukiwania w głąb (ang. Depth First Search - DFS). W przypadku grafu istnieje pewna trudność, która nie pojawiała się przy drzewach – w grafach krawędzie mogą tworzyć cykle lub pętle, czyli prowadzić do tego samego wierzchołka. Powoduje to konieczność modyfikacji podstawowego algorytmu w celu wyeliminowania zapętlenia się. Rozwiązaniem jest wprowadzenie dla każdego wierzchołka dodatkowego składnika, który będzie informował algorytm, czy wierzchołek ten został już odwiedzony.

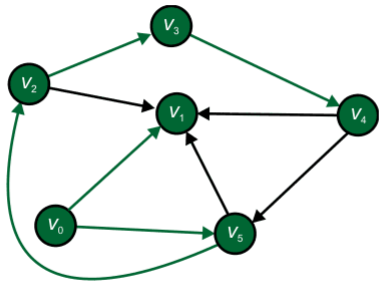
Algorytm DFS "w głąb"

Przykład:



Algorytm DFS "w głąb"

Wynik:



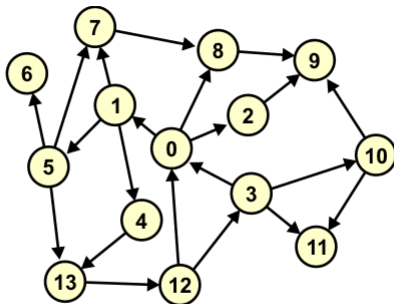
Algorytm BFS "wszerz"

Algorytm przechodzenia wszerz (ang. breadth-first search, BFS) opisaliśmy już przy przechodzeniu drzew binarnych. Dla grafu działa on następująco:

Zaczynamy odwiedzanie od wierzchołka startowego. Następnie odwiedzamy wszystkich jego sąsiadów. Dalej odwiedzamy wszystkich nieodwiedzonych jeszcze sąsiadów sąsiadów. Itd.

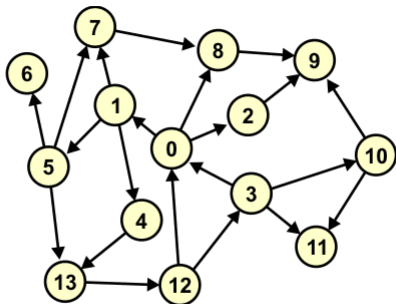
Algorytm BFS "wszerz"

Przykład:



Algorytm BFS "wszerz"

Wynik:



0,1,2,8,4,5,7,9,13,6,12,3,10,11

Algorytm Dijkstry

Algorytm Dijkstry, opracowany przez holenderskiego informatyka Edsgera Dijkstrę, służy do znajdowania najkrótszej ścieżki z pojedynczego źródła w grafie o nieujemnych wagach krawędzi.

Algorytm Dijkstry

Działanie

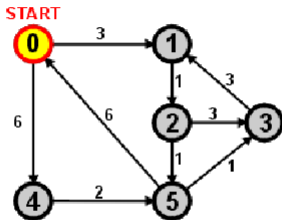
Algorytm Dijkstry znajduje w grafie wszystkie najkrótsze ścieżki pomiędzy wybranym wierzchołkiem a wszystkimi pozostałymi, przy okazji wyliczając również koszt przejścia każdej z tych ścieżek. Algorytm Dijkstry jest przykładem algorytmu zachłannego.

Zastosowanie

Z algorytmu Dijkstry można skorzystać przy obliczaniu najkrótszej drogi do danej miejscowości. Wystarczy przyjąć, że każdy z punktów skrzyżowań dróg to jeden z wierzchołków grafu, a odległości między punktami to wagi krawędzi.

Algorytm Dijkstry

Przykład:



Algorytm Dijkstry

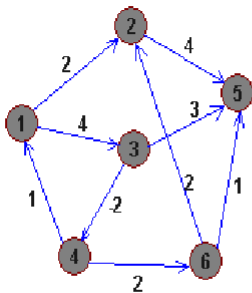
D	1	2	3	4	5
1,2,3,4,5	3	-	-	6	-
2,3,4,5	3	4	-	6	-
3,4,5	3	4	7	6	5
3,4	3	4	6	6	5
4	3	4	6	6	5

Algorytm Bellmana-Forda

Algorytm Bellmana-Forda rozwiązuje problem najkrótszej ścieżki, tj. pozwala znaleźć ścieżkę o najmniejszej wadze pomiędzy dwoma wierzchołkami w grafie ważonym. W odróżnieniu od algorytmu Dijkstry, poprawność algorytmu Bellmana-Forda nie opiera się na założeniu, że wagi w grafie są nieujemne (nie może jednak występować cykl o łącznej ujemnej wadze osiągalny ze źródła). Za tę ogólność płaci się jednak wyższą złożonością czasową.

Algorytm Bellmana-Forda

Przykład:



	1	2	3	4	5	6
1	0	2	4	*	*	*
2	*	0	*	*	4	*
3	*	*	0	-2	3	*
4	1	*	*	0	*	2
5	*	*	*	*	0	*
6	*	2	*	*	1	0

Algorytm Bellmana-Forda

Przebieg:

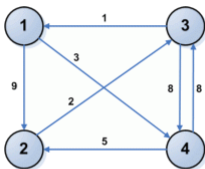
k	D(1)	D(2)	D(3)	D(4)	D(5)	D(6)
0	0	2	4	*	*	*
1	0	2	4	2_a	6_b	4_c
2	0	2	4	2	5_d	4
3	0	2	4	2	5	4

Algorytm Floyda-Warshalla

Algorytm Floyda-Warshalla jest przykładem algorytmu wyznaczającego koszt najkrótszej ścieżki między każdą parą wierzchołków w grafie ważonym. Do wyznaczania najkrótszych ścieżek stosuje się najczęściej reprezentację grafu w postaci macierzy sąsiedztwa. Rozwiązanie problemu jest również zapisywane w postaci tablicy dwuwymiarowej D o wymiarach $N \times N$, gdzie N jest liczbą wierzchołków grafu. W rezultacie wykonania algorytmu w każdej komórce $D[i,j]$ znajduje się wartość odpowiadająca kosztowi najkrótszej ścieżki prowadzącej z wierzchołka i do wierzchołka j .

Algorytm Floyda-Warshalla

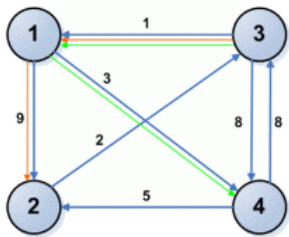
Przykład:



		Wzłez koñcowy			
		1	2	3	4
Wzłez poczãtkowy	1	0	9	999	3
	2	999	0	2	999
	3	1	999	0	8
	4	999	5	8	0

Algorytm Floyd-Warshalla

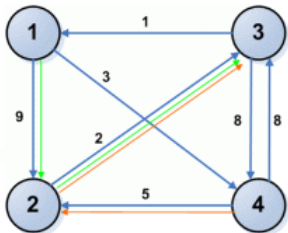
D(1)



		Węzeł końcowy			
		1	2	3	4
Węzeł początkowy	1	0	9	999	3
	2	999	0	2	999
	3	1	10	0	4
	4	999	5	8	0

Algorytm Floyda-Warshalla

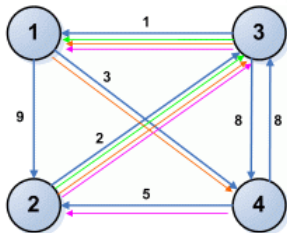
D(2)



		Węzeł końcowy			
		1	2	3	4
Węzeł początkowy	1	0	9	11	3
	2	999	0	2	999
	3	1	10	0	4
	4	999	5	7	0

Algorytm Floyda-Warshalla

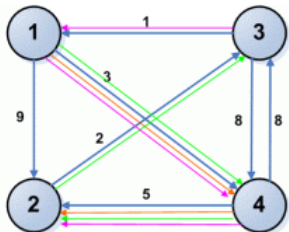
D(3)



		Węzeł końcowy			
		1	2	3	4
Węzeł początkowy	1	0	9	11	3
	2	3	0	2	6
	3	1	10	0	4
	4	8	5	7	0

Algorytm Floyda-Warshalla

D(4)



		Węzeł końcowy			
		1	2	3	4
Węzeł początkowy	1	0	8	10	3
	2	3	0	2	6
	3	1	9	0	4
	4	8	5	7	0

Dziękuję za uwagę :)