

FLOYD-WARSHALL ALGORITAM

OBLIKOVANJE I ANALIZA ALGORITAMA

MARTIN VLAHOVIĆ

MATEMATIČKI ODSJEK
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
SVEUČILIŠTE U ZAGREBU

30. PROSINCA 2021.



- Zadan je težinski graf $G = (V, E)$ s vrhovima V . Enumerirajmo vrhove grafa: $V = \{v_1, v_2, \dots, v_{|V|}\}$.
- Zadane su težine $w(u, v) \geq 0$ za svaki par bridova $u, v \in V$. Ako vrhovi u i v nisu povezani u grafu, koristimo $w(u, v) = +\infty$.
- Želimo odrediti **najkraću duljinu puta između svih parova vrhova u grafu**, gdje duljinu puta od vrha u od v preko $v_{i_1}, \dots, v_{i_n} \in V$:

$$u \rightarrow v_{i_1} \rightarrow v_{i_2} \cdots \rightarrow v_{i_n} \rightarrow v,$$

definiramo kao sumu težina pripadnih bridova:

$$w(u, v_{i_1}) + w(v_{i_1}, v_{i_2}) + \cdots + w(v_{i_n}, v).$$

- Pretpostavljamo da graf G nema **negativnih ciklusa**, inače problem nema dobro definirano rješenje.

- Pretpostavimo da poznajemo najkraće duljine puteva za sve parove vrhova $u, v \in V$, ali uz ograničenje da se u putevima smiju koristiti samo vrhovi $v_1, v_2, \dots, v_k \in V$, gdje je $k < |V|$ neki fiksni broj. Označimo duljine tih puteva s $d_k(u, v)$.
- Pitamo se, možemo li poznavanjem $d_k(u, v)$, pronaći $d_{k+1}(u, v)$, za sve parove $u, v \in V$, tj. možemo li pronaći duljine najkraćih puteva za sve parove vrhova $u, v \in V$, gdje u putevima koristimo vrhove v_1, v_2, \dots, v_{k+1} ?
- Ako riješimo taj problem, onda jednostavnim iteriranjem postupka od $k = 1$ do $k = |V|$ dolazimo do rješenja problema: poznavanje $d_{|V|}(u, v)$, za sve parove $u, v \in V$.

- Pretpostavimo da poznajemo $d_k(u, v)$, za sve parove $u, v \in V$ i $k < |V|$. Uzmimo i fiksirajmo proizvoljni par vrhova $u, v \in V$. Želimo izračunati $d_{k+1}(u, v)$.
- Ako put najkraće duljine od u do v koristeći vrhove v_1, \dots, v_{k+1} ne prolazi kroz vrh v_{k+1} , tada je očito: $d_{k+1}(u, v) = d_k(u, v)$.
- U protivnom, najkraći put od u do v prolazi kroz međuvrh v_{k+1} . Najkraći put u svojim vrhovima **samo jednom** posjećuje vrh v_{k+1} , jer bismo u protivnom imali negativan ciklus!
- Tada je očito najkraći put od u do v koji koristi vrhove v_1, \dots, v_{k+1} , i samo jednom prolazi kroz vrh v_{k+1} konkatencija najkraćeg puta od u do v_{k+1} koji koristi vrhove v_1, \dots, v_k i najkraćeg puta od v_{k+1} do v , koji koristi vrhove v_1, \dots, v_k .
- Uzimajući oba slučaja u obzir, dobivamo:

$$d_{k+1}(u, v) = \min \left\{ d_k(u, v), d_k(u, v_{k+1}) + d_k(v_{k+1}, v) \right\}.$$

Algorithm 1 Floyd-Warshall algorithm

procedure FLOYDWARSHALL

Input: Graf s vrhovima $V = \{v_1, \dots, v_{|V|}\}$ i bridovima s težinama $w : V \times V \rightarrow \mathbb{R}$, bez negativnih ciklusa.

Output: Duljine najkraćih puteva $d(u, v)$ za sve parove vrhova $u, v \in V$.

- Inicijaliziraj $|V| \times |V|$ matricu $d(u, v) = w(u, v)$, za $u \neq v$, i $d(u, u) = 0$.

for $k = 1, \dots, |V|$ **do**

for $i = 1, \dots, |V|$ **do**

for $j = 1, \dots, |V|$ **do**

$$d(v_i, v_j) = \min \left\{ d(v_i, v_j), d(v_i, v_k) + d(v_k, v_j) \right\}$$

return d

end procedure

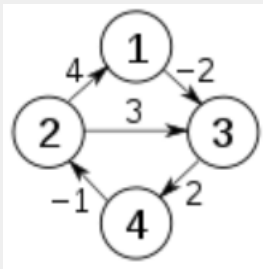
- Uočimo da u pseudokodu, iteraciju iz koraka k u $k + 1$ radimo na **jednoj matrici** d .
- S obzirom na to da tokom *update-anja* matrice vrijednostima $d_{k+1}(u, v)$ istovremeno i čitamo iz nje vrijednosti $d_k(u, v)$, postavlja se pitanje, je li opravdano koristiti samo jednu matricu d ?
- Ono što moramo pokazati jest da elementi u k -tom retku i k -tom stupcu: $d(:, v_k)$ i $d(v_k, :)$ ostaju nepromijenjeni tokom k -te iteracije vanjske petlje, jer jedino njih koristimo tokom *update-anja*.
- Ubacimo primjerice $u = v_{k+1}$ u osnovnu relaciju:

$$d_{k+1}(u, v) = \min \left\{ d_k(u, v) , d_k(u, v_{k+1}) + d_k(v_{k+1}, v) \right\},$$

tada dobivamo:

$$d_{k+1}(v_{k+1}, v) = \min \{ d_k(v_{k+1}, v), \underbrace{d_k(v_{k+1}, v_{k+1})}_0 + d_k(v_{k+1}, v) \} = d_k(v_{k+1}, v).$$

- Analogno za $v = v_{k+1}$. Dakle, tokom k -te iteracije, se k -ti redak i k -ti stupac matrice d ne mijenjanju, te je stoga dovoljno koristiti samo jednu matricu u pseudokodu.



		j			
		1	2	3	4
$k = 0$	1	0	∞	-2	∞
	2	4	0	3	∞
	3	∞	∞	0	2
	4	∞	-1	∞	0

$k = 0$		j			
		1	2	3	4
i	1	0	∞	-2	∞
	2	4	0	3	∞
	3	∞	∞	0	2
	4	∞	-1	∞	0

$k = 1$		j			
		1	2	3	4
i	1	0	∞	-2	∞
	2	4	0	2	∞
	3	∞	∞	0	2
	4	∞	-1	∞	0

$k = 1$		j			
		1	2	3	4
i	1	0	∞	-2	∞
	2	4	0	2	∞
	3	∞	∞	0	2
	4	∞	-1	∞	0

$k = 2$		j			
		1	2	3	4
i	1	0	∞	-2	∞
	2	4	0	2	∞
	3	∞	∞	0	2
	4	3	-1	1	0

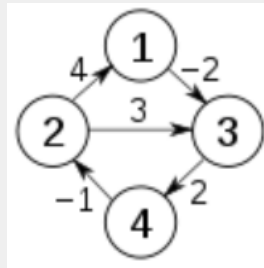
$k = 2$		j			
		1	2	3	4
i	1	0	∞	-2	∞
	2	4	0	2	∞
	3	∞	∞	0	2
	4	3	-1	1	0

$k = 3$		j			
		1	2	3	4
i	1	0	∞	-2	0
	2	4	0	2	4
	3	∞	∞	0	2
	4	3	-1	1	0

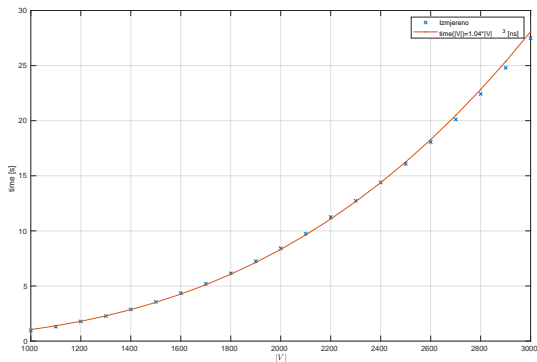
$k = 3$		j			
		1	2	3	4
i	1	0	∞	-2	0
	2	4	0	2	4
	3	∞	∞	0	2
	4	3	-1	1	0

$k = 4$		j			
		1	2	3	4
i	1	0	-1	-2	0
	2	4	0	2	4
	3	5	1	0	2
	4	3	-1	1	0

$k = 4$		j			
		1	2	3	4
i	1	0	-1	-2	0
	2	4	0	2	4
	3	5	1	0	2
	4	3	-1	1	0



- Algoritam očito ima $\Theta(|V|^3)$ vremensku i $\Theta(|V|^2)$ prostornu složenost, neovisno o broju netrivialnih bridova (onih za koje je $w(u, v) < +\infty$). To sugerira da je algoritam primjeren za *guste* grafove s velikim brojem netrivialnih bridova.
- Implementacijom algoritma u C programskom jeziku i mjerenjem vremena izvršavanja na slučajno generiranim grafovima to potvrđuje i u praksi.





T. H. CORMEN, C. E. LEISERSON, R. L. RIVEST, C. STEIN,
INTRODUCTION TO ALGORITHMS (3RD ED.),
MIT PRESS AND MCGRAW-HILL (2009)



FLOYD-WARSHALL ALGORITHM, WIKIPEDIA ČLANAK, [LINK](#)



GITHUB REPOZITORIJ S IMPLEMENTACIJOM I TESTOVIMA FLOYD-WARSHALL ALGORITMA,
[HTTPS://GITHUB.COM/MAVLAHO/OAA](https://github.com/MAVLAHO/OAA)