

BELLMAN-FORD, FLOYD-WARSHALL, JOHNSON

Petr Ryšavý

13. listopadu 2025

Katedra počítačů, FEL, ČVUT

BELLMAN-FORDŮV ALGORITMUS

Vstup:

- Orientovaný graf $G = (V, E)$
- každá hrana má váhu
- počáteční vrchol s

Výstup:

- Pro každý vrchol $v \in V$ spočtěme

$$L(v) = \text{délka nejkratší cesty z } s \text{ do } v.$$

Předpoklady:

- (pro pohodlnost) Vše je dostupné z s .

- Podcesta nejkratší cesty je sama o sobě nejkratší cestou.
- Omezíme problém podle počtu hran v nejkratší cestě.

- Podcesta nejkratší cesty je sama o sobě nejkratší cestou.
- Omezíme problém podle počtu hran v nejkratší cestě.

Máme tedy jeden podproblém na

1. každý vrchol v
2. počet hran, které připouštíme na nejkratší cestě.

Lemma *Nechť $G = (V, E)$ je orientovaný graf s délkami hran l_e a počátečním vrcholem s . Pro všechny $v \in V$ a $i \in \mathbb{N}$ označme jako P nejkratší cestu z s do v s nejvýše i hranami (cykly jsou povoleny). Pak platí*

1. *pokud má P nejvýše $i - 1$ hran, pak je i nejkratší cestou z s do v s nejvýše $i - 1$ hranami;*
2. *pokud má P právě i hran s poslední hranou z w do v , pak cesta P' je nejkratší cestou z s do w s nejvýše $i - 1$ hranami.*

Kolik existuje kandidátů pro optimální řešení pro podproblém zahrnující vrchol v a i hran?

1. 2
2. $1 + \text{indegree}(v)$
3. $n - 1$
4. n

- Necht' $L_{i,v}$ je délka nejkratší cesty z s do v , která připouští nejvýše i hran.
- Pak $\forall v \in V, i \in \mathbb{N}$ platí

$$L_{i,v} = \min \begin{cases} L_{i-1,v}, \\ \min_{(w,v) \in E} \{L_{i-1,w} + l_{wv}\}. \end{cases}$$

Korektnost vychází z předchozího lemmatu.

Pokud graf G neobsahuje cyklus záporné délky

Pokud graf G neobsahuje cyklus záporné délky, pak

- nejkratší cesty neobsahují cykly,
- mají nejvýše $n - 1$ hran,
- stačí uvažovat pouze i do $n - 1$.

```
function BELLMAN-FORD(graph, s) returns shortest path to each v  
   $A \leftarrow$  prázdne pole indexované i a v  
   $A[0, s] = 0$   
   $\forall v \in V \setminus \{s\} : A[0, v] = \infty$   
  for i = 1 to n - 1 do  
    for each v ∈ V do  
       $A[i, v] = \min \{ A[i - 1, v], \min_{(w, v) \in E} \{ A[i - 1, w] + l_{wv} \} \}$   
    end for  
  end for  
end function
```


Jaký je čas běhu Bellman-Fordova algoritmu?

1. $\mathcal{O}(n^2)$
2. $\mathcal{O}(mn)$
3. $\mathcal{O}(n^3)$
4. $\mathcal{O}(m^2)$

- Pokud se nic nezmění pro nějaké $j < n - 1$, pak víme, že

$$\forall v \in V : A[j, v] = A[j - 1, v]$$

- Nic se nezmění ani pro $j + 1$, ani nikdy dále.
- Můžeme zastavit výpočet dříve.

DETEKCE ZÁPORNÝCH CYKLŮ

Tvrzení *G nemá cyklus záporné délky*



pokud přidáme jednu iteraci B-F algoritmu, pak

$\forall v \in V : A[n-1, v] = A[n, v]$.

- Máme převodní kurzy několika měn.
- Detekujeme cyklus, kde je součin kurzů ≥ 1 .
- UVA 104 - Arbitrage

- Máme převodní kurzy několika měn.
- Detekujeme cyklus, kde je součin kurzů ≥ 1 .
- UVA 104 - Arbitrage
- Formulujeme jako detekci negativního cyklu po nahrazení délek hran zápornou hodnotou jejich logaritmu.

FLOYD-WARSHALLŮV ALGORITMUS

Vstup:

- Orientovaný graf $G = (V, E)$
- každá hrana má váhu l_e

Výstup:

1. Pro každou dvojici vrcholů $u, v \in V$ spočtěme délku nejkratší cesty z u do v .

nebo

2. Identifikujeme cyklus se zápornou délkou.

APSP se redukuje na n volání SSSP.

Algoritmus	Čas běhu	Přípustná data
n volání Dijkstra	$\mathcal{O}(mn \log n)$	nezáporné délky hran
n volání Bellman-Forda	$\mathcal{O}(mn^2)$	obecný graf
Floyd-Warshall	$\mathcal{O}(n^3)$	obecný graf
Johnson ¹	$\mathcal{O}(mn \log n)$	obecný graf

¹1 volání B-F, n volání Dijkstra

Proč další algoritmus pro hledání nejkratších cest?

- Floyd-Warshall funguje i pro záporné délky hran v čase $\mathcal{O}(n^3)$
- Lepší než Bellman-Ford spuštěný n -krát
- V hustých grafech srovnatelný s n běhy Dijkstry
- Dodnes je otevřenou otázkou, zda lze dosáhnout v hustých grafech výrazně lepšího času než $\mathcal{O}(n^3)$

- Podobně jako u Bellman-Forda omezíme vrcholy, přes které cesta z s do t může procházet
- Omezíme se na konkrétní výběr vrcholů

Lemma *Nechť G neobsahuje cyklus záporné délky a vrcholy $V = \{1, 2, \dots, n\}$ jsou očíslované a nechť $V^{(k)} = \{1, 2, \dots, k\}$. Zafixujme počátek $i \in V$ a cíl $j \in V$. Nechť P je nejkratší cesta (bez cyklu) z i do j taková, že všechny vnitřní vrcholy jsou z $V^{(k)}$. Pak platí jedna z následujících možností.*

- *Pokud není k na cestě P , pak P je nejkratší cesta z i do j s vnitřními vrcholy z $V^{(k-1)}$.*
- *Pokud je k vnitřním vrcholem cesty P , pak je zde jednou a P lze rozdělit na dvě podcesty P_1 z i do k a P_2 z k do j . Přitom platí že obě dvě cesty mají vnitřní vrcholy z $V^{(k-1)}$ a jsou nejkratšími cestami bez cyklu mezi odpovídajícími vrcholy.*

Nechť A je trojrozměrné pole indexované $A[i, j, k]$ uchovávající délky nejkratších cest z i do j přes vrcholy z $\{1, 2, \dots, k\}$. Jak bude inicializované $A[i, j, 0]$ pro případy, kdy $i = j$, $(i, j) \in E$ a $(i, j) \notin E$ (v tomto pořadí):

1. $0, 0, \infty$
2. $0, c_{i,j}, c_{i,j}$
3. $0, c_{i,j}, \infty$
4. $\infty, c_{i,j}, \infty$
5. $\infty, c_{i,j}, 0$

function FLOYD-WARSHALL(*graph*) **returns** minimal path between all pairs of nodes

$A \leftarrow \text{EMPTY-3D-ARRAY}(n, n, n)$

$$A[i, j, 0] = \begin{cases} 0 & \text{if } i = j, \\ c_{ij} & \text{if } (i, j) \in \text{EDGES}(\textit{graph}), \\ \infty & \text{otherwise.} \end{cases}$$

for $k = 1$ **to** n **do**

for $i = 1$ **to** n **do**

for $j = 1$ **to** n **do**

$A[i, j, k] = \min \{A[i, j, k - 1], A[i, k, k - 1] + A[k, j, k - 1]\}$

end for

end for

end for

return $A[., ., n]$

end function

- Co se stane, máme-li v grafu cyklus záporné délky?

- Co se stane, máme-li v grafu cyklus záporné délky?
- Alespoň jedno $A[i, i, n]$ bude záporné.

- Stačí si zapamatovat pole $B[i, j]$ udávající nejvyšší vrchol na cestě z i do j
- Víme, že daný vrchol na cestě, je, můžeme ji rozdělit
- při druhém případě, kdy volíme $A[i, j, k] = A[i, k, k - 1] + A[k, j, k - 1]$ nastavujeme $B[i, j]$ na k

JOHNSONŮV ALGORITMUS

Vstup:

- Orientovaný graf $G = (V, E)$
- každá hrana má váhu l_e

Výstup:

1. Pro každou dvojici vrcholů $u, v \in V$ spočtěme délku nejkratší cesty z u do v .

nebo

2. Identifikujeme cyklus se zápornou délkou.

APSP se redukuje na n volání SSSP.

Algoritmus	Čas běhu	Přípustná data
n volání Dijkstra	$\mathcal{O}(mn \log n)$	nezáporné délky hran
n volání Bellman-Forda	$\mathcal{O}(mn^2)$	obecný graf
Floyd-Warshall	$\mathcal{O}(n^3)$	obecný graf
Johnson ²	$\mathcal{O}(mn \log n)$	obecný graf

²1 volání B-F, n volání Dijkstra

Nechť $G = (V, E)$ je orientovaný graf s obecnými délkami hran l_e .
Zafixujme číslo p_v pro každý vrchol $v \in V$. Nechť pro každou hranu $e = (u, v)$ grafu G je

$$l'_e = l_e + p_u - p_v.$$

Je-li P libovolná cesta z s do t délky L , jaká je délka P' , pokud převážíme hrany?

1. L
2. $L + p_s + p_t$
3. $L + p_s - p_t$
4. $L - p_s + p_t$

- Každá cesta z s do t se změní o stejnou hodnotu.
- Převážení zachovává uspořádání cest podle jejich délek.
- Nejkratší cesta zůstává nejkratší cestou.

Existují vždy takové váhy, aby se ceny hran staly nezáporné?

- Přidejme nový počáteční vrchol.
- Spustíme Bellman-Fordův algoritmus a spočtíme nejkratší cestu z s do libovolného vrcholu.
- hodnoty použijme jako váhy vrcholů

$$p_v = \text{délka nejkratší cesty z } s \text{ do } v.$$

Tvrzení *Pro všechny hrany $e = (u, v)$ platí*

$$l'_e = l_e + p_u - p_v \geq 0.$$

function JOHNSON-ALGORITHM(*graph*)

vytvoř G' z G přidáním vrcholu s a hran (s, v) o ceně 0 pro všechny $v \in V$

spuť Bellman-Fordův algoritmus na G' s počátečním vrcholem $s \triangleright$
zastav, pokud nalezneš cyklus záporné délky

$\forall v \in V : p_v \leftarrow$ délka nejkratší cesty z s do v v G'

$\forall e = (u, v) \in E : l'_e = l_e + p_u - p_v$

for each $u \in V$ **do**

spuť Dijkstrův algoritmus na grafu G s l'_e a s počátkem u
označ vypočtené délky jako $d'(u, v)$ pro všechny $v \in V$

end for

$\forall u, v \in V : d(u, v) \leftarrow d'(u, v) - p_u + p_v$

end function

$$\mathcal{O}(n) + \mathcal{O}(mn) + \mathcal{O}(m) + \mathcal{O}(mn \log n) + \mathcal{O}(n^2) = \mathcal{O}(mn \log n).$$

To je mnohem lepší než Floyd-Warshallův algoritmus pro řídké grafy.

- heavily inspired by Tim Roughgarden's online courses,
<http://theory.stanford.edu/~tim/videos.html>
- Robert Sedgewick and Kevin Wayne, Algorithms,
<http://algs4.cs.princeton.edu/home/>, namely
<http://algs4.cs.princeton.edu/44sp/>
- Halim, S., Halim, F., Skiena, S. S., & Revilla, M. A. (2013).
Competitive Programming 3. Lulu Independent Publish.

DĚKUJI ZA POZORNOST.
ČAS NA OTÁZKY!