Pojmy v algoritmech hledání nejkratších cest

text pro studenty učitelství na FP TUL

Martina Šimůnková

24. dubna 2024

Cílem prezentace je vysvětlit, co je *otevření*, *uzavření* a *relaxace* vrcholu v algoritmech na hledání nejkratší cesty v grafu.

Během algoritmu prochází vrcholy stavy: *nenalezený*, *otevřený*, *uzavřený*.

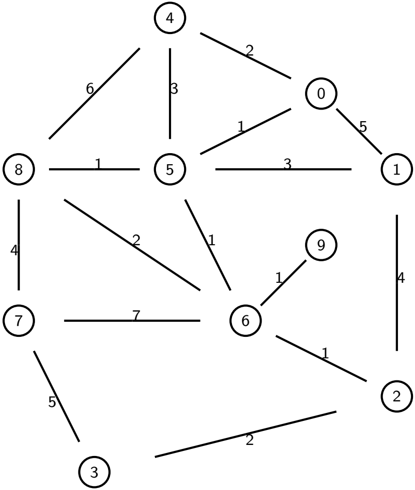
Na začátku jsou všechny vrcholy *nenalezené*.

Vrcholy postupně vkládáme do datové struktury, kterou může být fronta, list, nebo halda. Vrcholy v této struktuře nazýváme *otevřené*. Vrcholy ze struktury odebrané nazýváme *uzavřené*.

Vrchol *relaxujeme* před jeho uzavřením. *Relaxace* vrcholu v je:

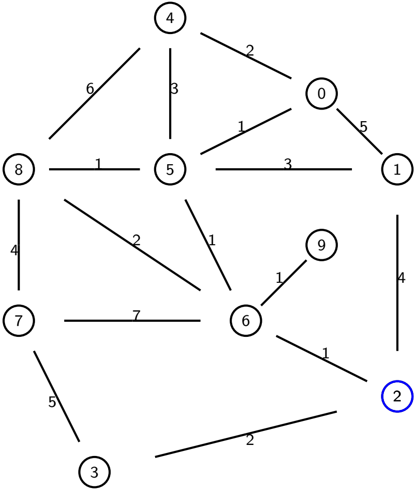
- pro všechny sousedy s vrcholu v
  - spočítáme délku cesty z v_0 přes v do s
  - pokud je kratší než stávající cesta z v_0 do s
    - zaktualizujeme délku cesty pro vrchol s
    - vrchol s otevřeme

Pojmy vysvětlíme nejdříve na Dijkstrově algoritmu.
Nenalezené vrcholy označíme černou barvou.
Otevřené vrcholy označíme modrou barvou.
Uzavřené vrcholy označíme červenou barvou.



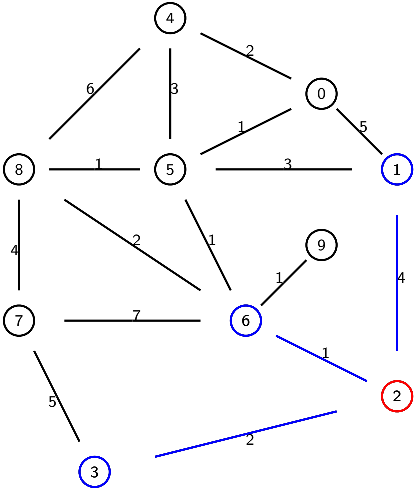
v	$h(v)$	$P(v)$
0	∞	
1	∞	
2	∞	
3	∞	
4	∞	
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	

Pojmy vysvětlíme nejdříve na Dijkstrově algoritmu.
Nenalezené vrcholy označíme černou barvou.
Otevřené vrcholy označíme modrou barvou.
Uzavřené vrcholy označíme červenou barvou.
Otevřeli jsme počáteční vrchol $v = 2$.



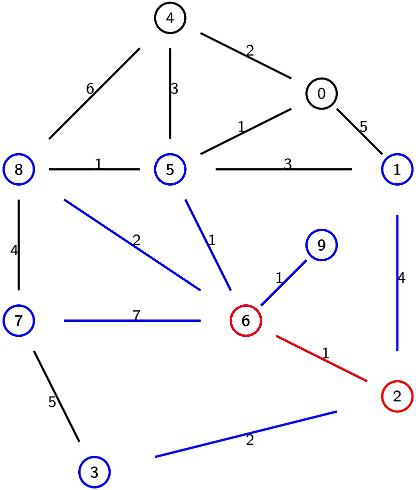
v	$h(v)$	$P(v)$
0	∞	
1	∞	
2	0	
3	∞	
4	∞	
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	

Pojmy vysvětlíme nejdříve na Dijkstrově algoritmu.
Nenalezené vrcholy označíme černou barvou.
Otevřené vrcholy označíme modrou barvou.
Uzavřené vrcholy označíme červenou barvou.
Zrelaxovali a *uzavřeli* jsme vrchol $v = 2$.



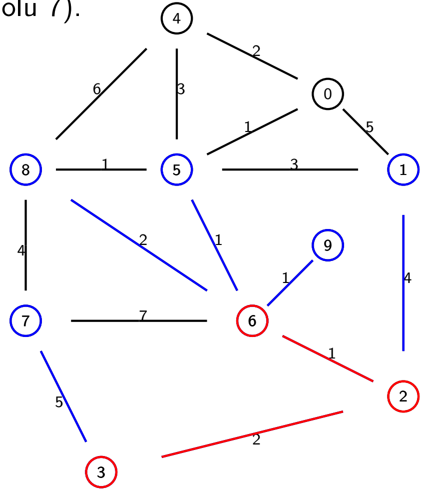
v	$h(v)$	$P(v)$
0	∞	
1	4	2
2	0	
3	2	2
4	∞	
5	∞	
6	1	2
7	∞	
8	∞	
9	∞	

Pojmy vysvětlíme nejdříve na Dijkstrově algoritmu.
Nenalezené vrcholy označíme černou barvou.
Otevřené vrcholy označíme modrou barvou.
Uzavřené vrcholy označíme červenou barvou.
Zrelaxovali a *uzavřeli* jsme vrchol $v = 6$.



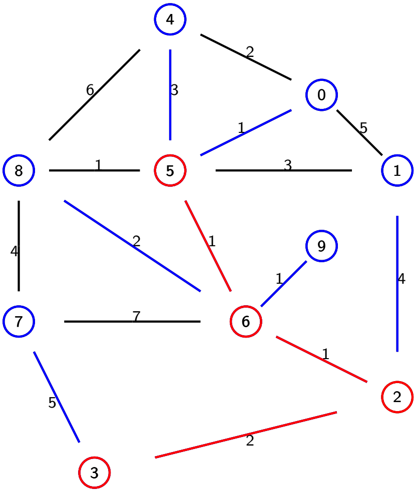
v	$h(v)$	$P(v)$
0	∞	
1	4	2
2	0	
3	2	2
4	∞	
5	2	6
6	1	2
7	8	6
8	3	6
9	2	6

Pojmy vysvětlíme nejdříve na Dijkstrově algoritmu.
Nenalezené vrcholy označíme černou barvou.
Otevřené vrcholy označíme modrou barvou.
Uzavřené vrcholy označíme červenou barvou.
Zrelaxovali a *uzavřeli* jsme vrchol $v = 3$ (všimněte si nové cesty do vrcholu 7).



v	$h(v)$	$P(v)$
0	∞	
1	4	2
2	0	
3	2	2
4	∞	
5	2	6
6	1	2
7	7	3
8	3	6
9	2	6

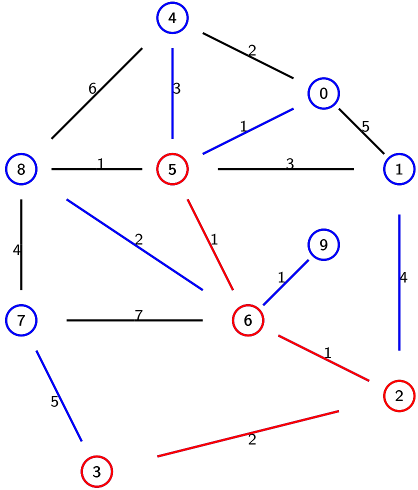
Pojmy vysvětlíme nejdříve na Dijkstrově algoritmu.
Nenalezené vrcholy označíme černou barvou.
Otevřené vrcholy označíme modrou barvou.
Uzavřené vrcholy označíme červenou barvou.
Zrelaxovali a *uzavřeli* jsme vrchol $v = 5$.



v	$h(v)$	$P(v)$
0	3	5
1	4	2
2	0	
3	2	2
4	5	5
5	2	6
6	1	2
7	7	3
8	3	6
9	2	6

Algoritmus pokračuje zrelaxováním a uzavřením dalších vrcholů.

Po skončení algoritmus vydá délky $h(v)$ nejkratších cest z vrcholu 2 do vrcholu v a předposlední vrchol $P(v)$ na této cestě.



v	$h(v)$	$P(v)$
0	3	5
1	4	2
2	0	
3	2	2
4	5	5
5	2	6
6	1	2
7	7	3
8	3	6
9	2	6

Ukážeme ještě odlišnosti Dijkstrova a Bellman-Fordova algoritmu.

Zatímco v Dijkstrově algoritmu každý vrchol *otevíráme* a *zavíráme* právě jednou, v Bellman-Fordově algoritmu jsou *otevřené* ty vrcholy, které čekají ve frontě, a to může nastat opakovaně.

Bellman-Fordův algoritmus projdeme v prezentaci až do konce.

Ukážeme ještě odlišnosti Dijkstrova a Bellman-Fordova algoritmu.

Zatímco v Dijkstrově algoritmu každý vrchol *otevíráme* a *zavíráme* právě jednou, v Bellman-Fordově algoritmu jsou *otevřené* ty vrcholy, které čekají ve frontě, a to může nastat opakovaně.

Bellman-Fordův algoritmus projdeme v prezentaci až do konce.

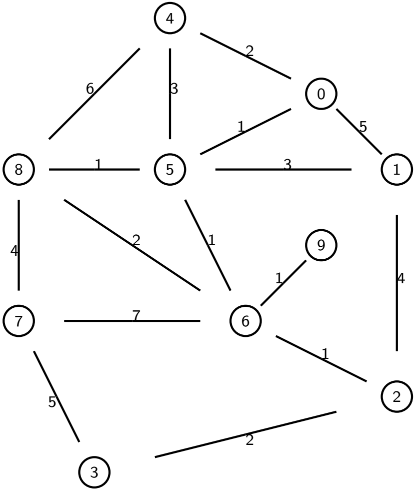
Ukážeme ještě odlišnosti Dijkstrova a Bellman-Fordova algoritmu.

Zatímco v Dijkstrově algoritmu každý vrchol *otevíráme* a *zavíráme* právě jednou, v Bellman-Fordově algoritmu jsou *otevřené* ty vrcholy, které čekají ve frontě, a to může nastat opakovaně.

Bellman-Fordův algoritmus projdeme v prezentaci až do konce.

Na začátku jsou všechny vrcholy *nenalezené*.

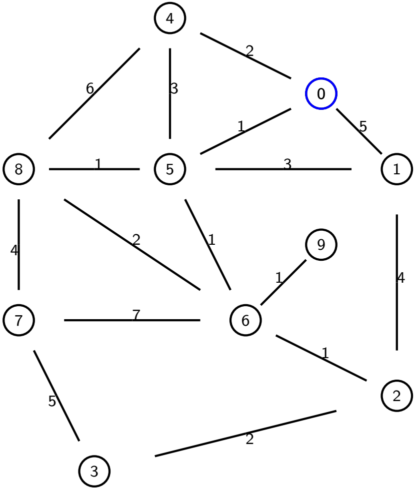
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	∞	
1	∞	
2	∞	
3	∞	
4	∞	
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	

Otevřeli jsme vrchol $v = 0$.

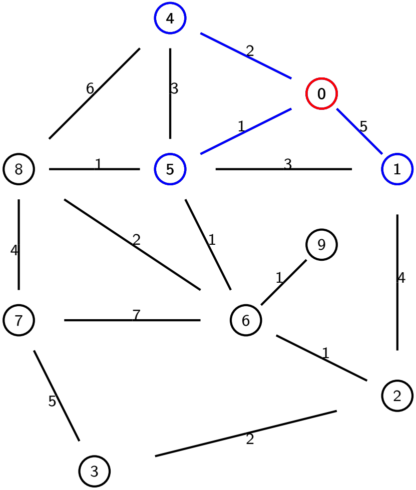
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	∞	
2	∞	
3	∞	
4	∞	
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	

Zrelaxovali jsme a uzavřeli vrchol $v = 0$.

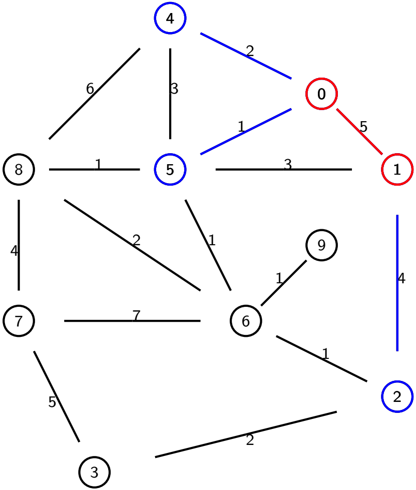
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	5	0
2	∞	
3	∞	
4	2	0
5	1	0
6	∞	
7	∞	
8	∞	
9	∞	

Zrelaxovali jsme a uzavřeli vrchol $v = 1$.

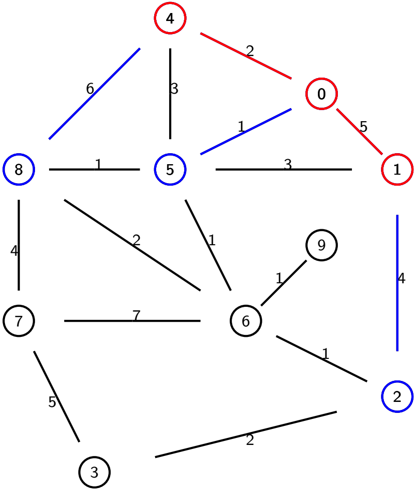
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	5	0
2	9	1
3	∞	
4	2	0
5	1	0
6	∞	
7	∞	
8	∞	
9	∞	

Zrelaxovali jsme a uzavřeli vrchol $v = 4$.

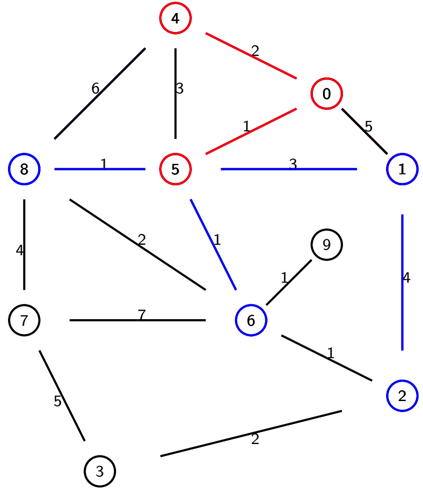
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	5	0
2	9	1
3	∞	
4	2	0
5	1	0
6	∞	
7	∞	
8	8	4
9	∞	

Zrelaxovali jsme a uzavřeli vrchol $v = 5$. Všimněte si *otevření* vrcholu $v = 1$ a přepočtu vzdálenosti do vrcholu $v = 8$.

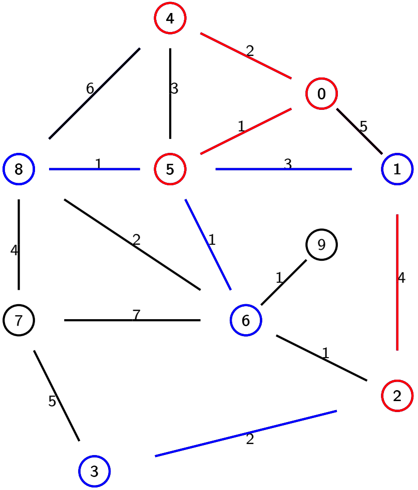
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	4	5
2	9	1
3	∞	
4	2	0
5	1	0
6	2	5
7	∞	
8	2	5
9	∞	

Zrelaxovali jsme a uzavřeli vrchol $v = 2$. Všimněte si otevřeného vrcholu 1. Před uzavřením vrchol 1 zrelaxujeme, přepočítáme vzdálenost do vrcholu 2 a tím vrchol 2 znovu otevřeme.

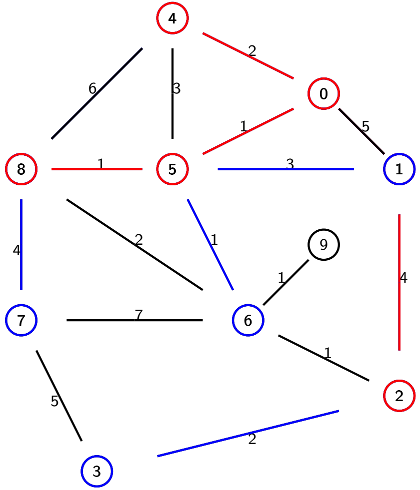
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	4	5
2	9	1
3	11	2
4	2	0
5	1	0
6	2	5
7	∞	
8	2	5
9	∞	

Zrelaxovali jsme a uzavřeli vrchol $v = 8$.

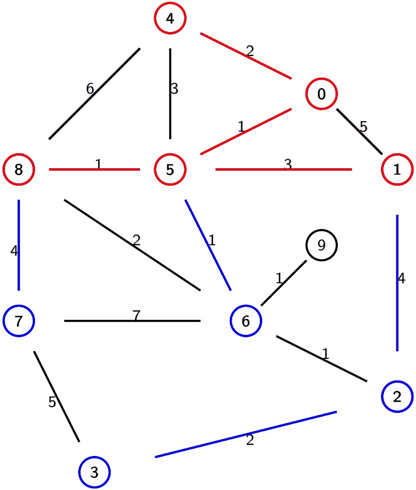
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	4	5
2	9	1
3	11	2
4	2	0
5	1	0
6	2	5
7	6	8
8	2	5
9	∞	

Zrelaxovali jsme a uzavřeli vrchol $v = 1$. Všimněte si *otevření* vrcholu $v = 2$.

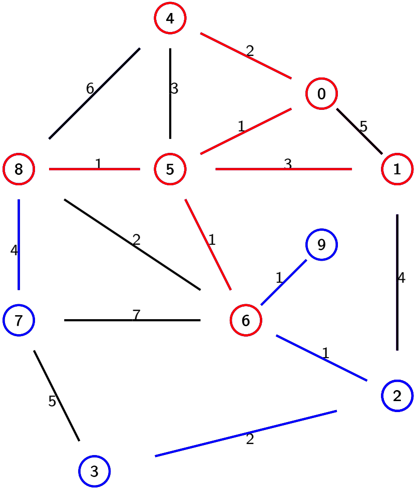
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	4	5
2	8	1
3	11	2
4	2	0
5	1	0
6	2	5
7	6	8
8	2	5
9	∞	

Zrelaxovali jsme a uzavřeli vrchol $v = 6$. Všimněte si přepočtu vzdálenosti do vrcholu $v = 2$.

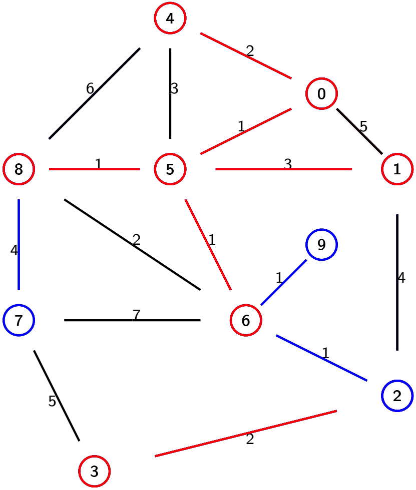
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	4	5
2	3	6
3	11	2
4	2	0
5	1	0
6	2	5
7	6	8
8	2	5
9	3	6

Zrelaxovali jsme a uzavřeli vrchol $v = 3$.

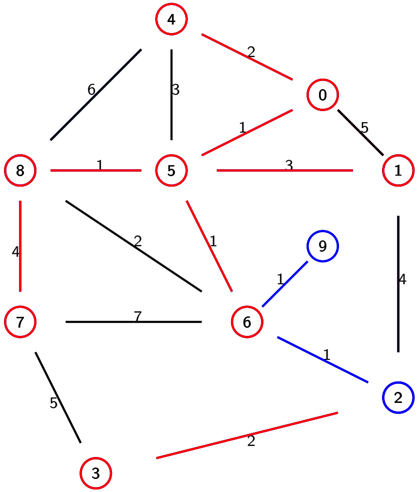
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	4	5
2	3	6
3	11	2
4	2	0
5	1	0
6	2	5
7	6	8
8	2	5
9	3	6

Zrelaxovali jsme a uzavřeli vrchol $v = 7$.

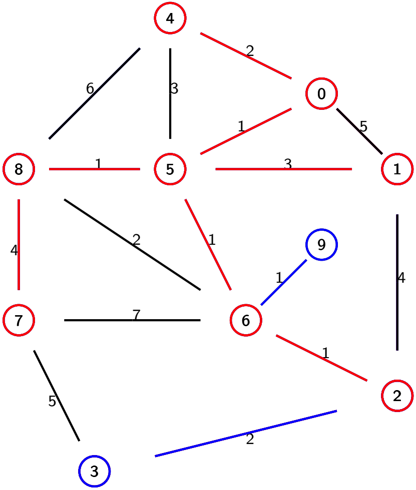
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	4	5
2	3	6
3	11	2
4	2	0
5	1	0
6	2	5
7	6	8
8	2	5
9	3	6

Zrelaxovali jsme a uzavřeli vrchol $v = 2$. Všimněte si otevření vrcholu $v = 3$.

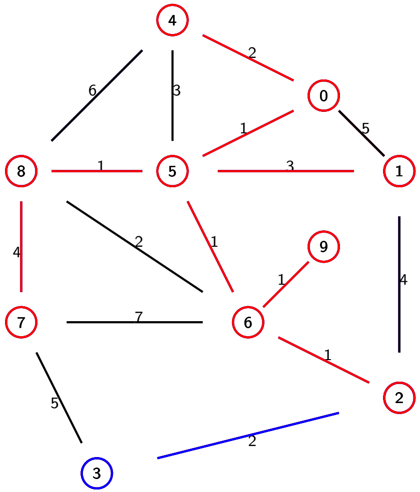
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	4	5
2	3	6
3	5	2
4	2	0
5	1	0
6	2	5
7	6	8
8	2	5
9	3	6

Zrelaxovali jsme a uzavřeli vrchol $v = 9$.

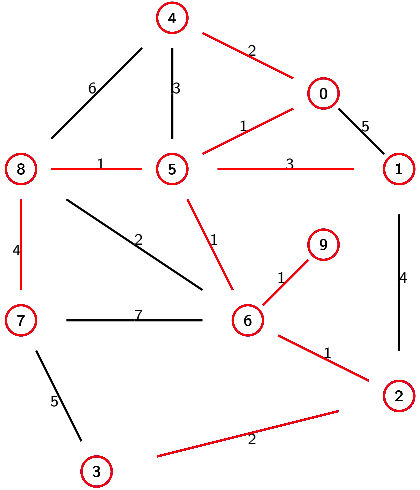
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	4	5
2	3	6
3	5	2
4	2	0
5	1	0
6	2	5
7	6	8
8	2	5
9	3	6

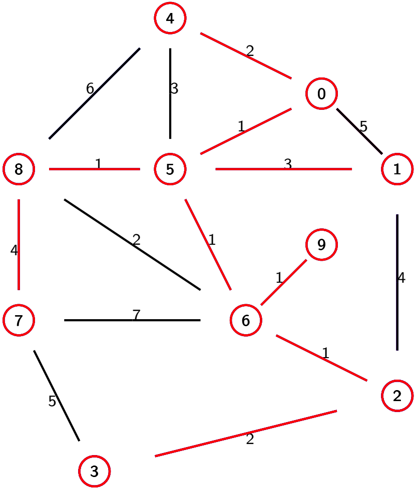
Zrelaxovali jsme a uzavřeli vrchol $v = 3$.

Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	4	5
2	3	6
3	5	2
4	2	0
5	1	0
6	2	5
7	6	8
8	2	5
9	3	6

Fronta je prázdná, algoritmus skončil.
Výsledkem jsou délky $h(v)$ nejkratších cest z vrcholu 0 do vrcholu v a předposlední vrchol $P(v)$ na této cestě.
Fronta: 0 | 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



v	$h(v)$	$P(v)$
0	0	
1	4	5
2	3	6
3	5	2
4	2	0
5	1	0
6	2	5
7	6	8
8	2	5
9	3	6