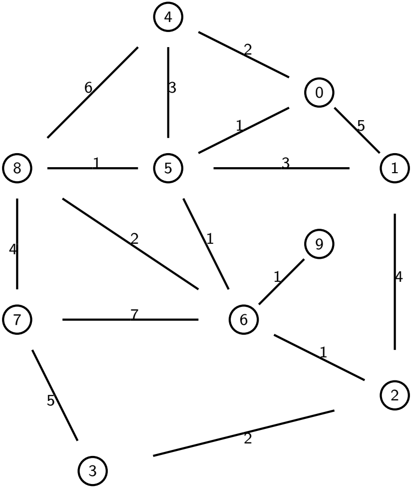
Pojmy v algoritmech hledání nejkratších cest

text pro studenty učitelství na FP TUL

Martina Šimůnková

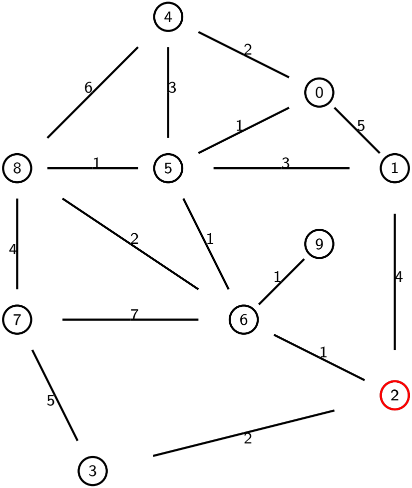
4. května 2023

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



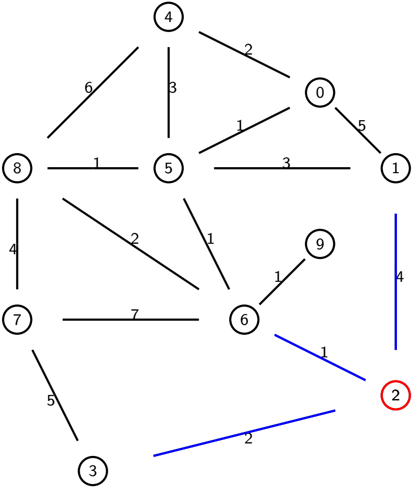
v	$h(v)$	$P(v)$
0	∞	
1	∞	
2	∞	
3	∞	
4	∞	
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	

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



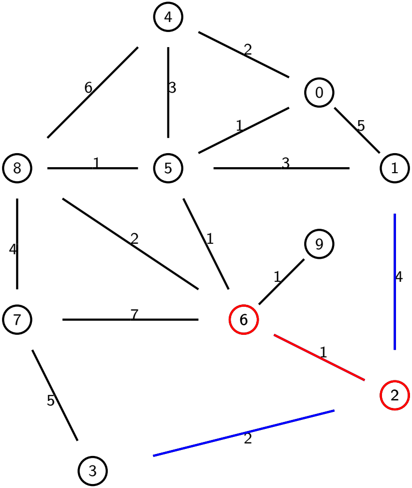
v	$h(v)$	$P(v)$
0	∞	
1	∞	
2	0	
3	∞	
4	∞	
5	∞	
6	∞	
7	∞	
8	∞	
9	∞	

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



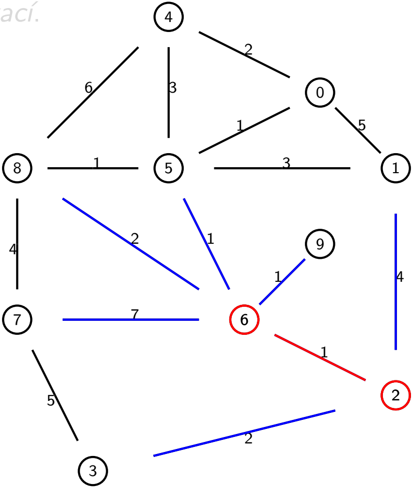
v	$h(v)$	$P(v)$
0	∞	
1	4	2
2	0	
3	2	2
4	∞	
5	∞	
6	1	2
7	∞	
8	∞	
9	∞	

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



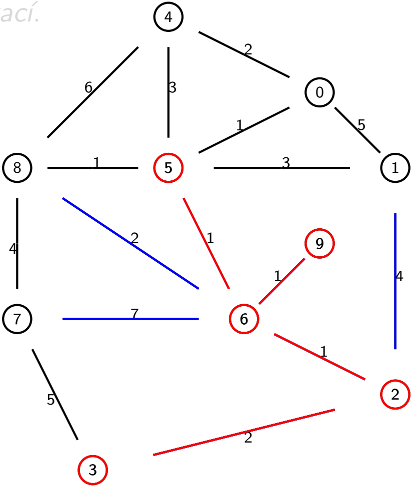
v	$h(v)$	$P(v)$
0	∞	
1	4	2
2	0	
3	2	2
4	∞	
5	∞	
6	1	2
7	∞	
8	∞	
9	∞	

Vrcholy 5, 7, 8, 9 *otevřeme*, když jsme našli cestu přes jejich souseda, vrchol 6. V tabulce vyznačíme délku cesty $h(v)$ a předchůdce na cestě $P(v)$. Vrchol $v = 3$ *uzavřeme* při nalezení cesty z $v_0 = 2$. Při uzavření vrcholu (pře)počítáme délky cest přes tento vrchol do jeho sousedů. Do vrcholu $v = 7$ vede kratší cesta přes $P(v) = 3$ než přes $P(v) = 6$. Toto přepočítání nazýváme *relaxací*.



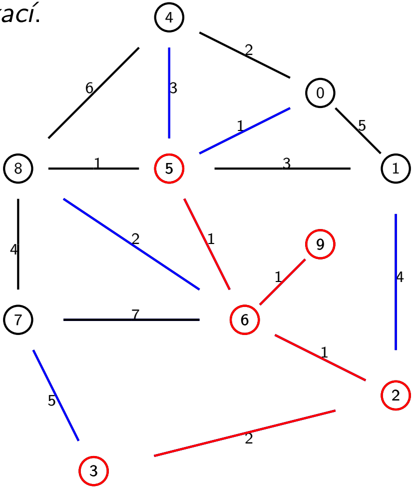
v	$h(v)$	$P(v)$
0	∞	
1	4	2
2	0	
3	2	2
4	∞	
5	2	6
6	1	2
7	8	6
8	3	6
9	2	6

Vrcholy 5, 7, 8, 9 *otevřeme*, když jsme našli cestu přes jejich souseda, vrchol 6. V tabulce vyznačíme délku cesty $h(v)$ a předchůdce na cestě $P(v)$. Vrchol $v = 3$ *uzavřeme* při nalezení cesty z $v_0 = 2$. Při uzavření vrcholu (pře)počítáme délky cest přes tento vrchol do jeho sousedů. Do vrcholu $v = 7$ vede kratší cesta přes $P(v) = 3$ než přes $P(v) = 6$. Toto přepočítání nazýváme *relaxací*.



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

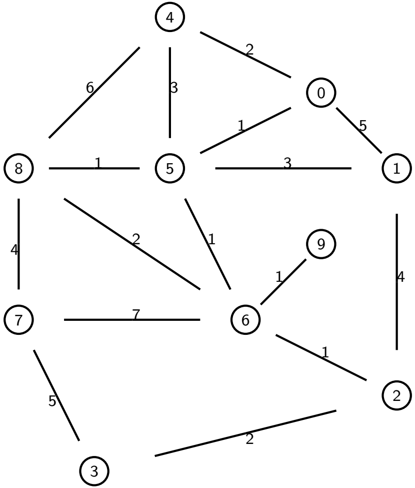
Vrcholy 5, 7, 8, 9 *otevřeme*, když jsme našli cestu přes jejich souseda, vrchol 6. V tabulce vyznačíme délku cesty $h(v)$ a předchůdce na cestě $P(v)$. Vrchol $v = 3$ *uzavřeme* při nalezení cesty z $v_0 = 2$. Při uzavření vrcholu (pře)počítáme délky cest přes tento vrchol do jeho sousedů. Do vrcholu $v = 7$ vede kratší cesta přes $P(v) = 3$ než přes $P(v) = 6$. Toto přepočítání nazýváme *relaxací*.



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

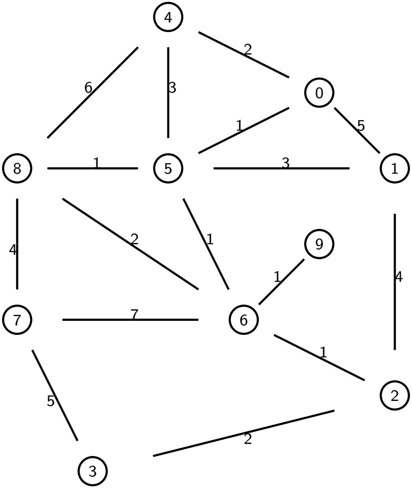
Ukážeme ještě odlišnosti Dijkstrova a Bellman-Fordova algoritmu.

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, jako v tomto případě s vrcholem $v = 1$, nebo později $v = 3$.



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

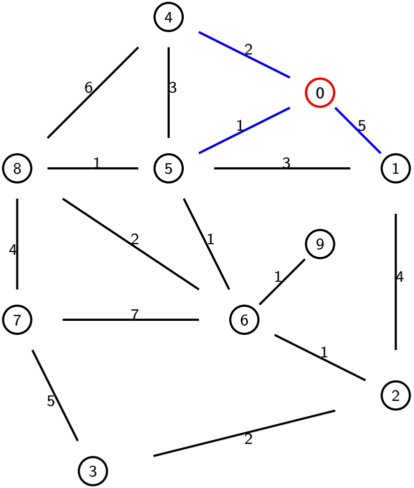
Ukážeme ještě odlišnosti Dijkstrova a Bellman-Fordova algoritmu. Zatímco v Dijkstrově algoritmu každý vrchol *otevíráme* a *zavíráme* právě jednou, v Bellman-Fordově algoritmu jsou *otevřené* ty vrcholy, které čekají ve frontě, a to může nastat opakovaně, jako v tomto případě s vrcholem $v = 1$, nebo později $v = 3$.



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

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, jako v tomto případě s vrcholem $v = 1$, nebo později $v = 3$.

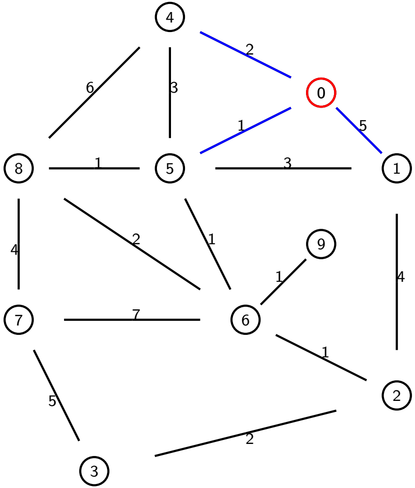
Fronta: 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



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

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, jako v tomto případě s vrcholem $v = 1$, nebo později $v = 3$.

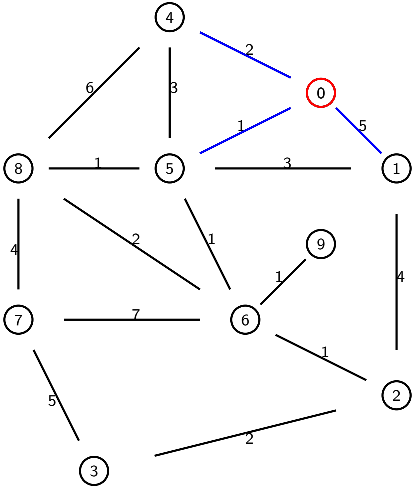
Fronta: 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



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

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, jako v tomto případě s vrcholem $v = 1$, nebo později $v = 3$.

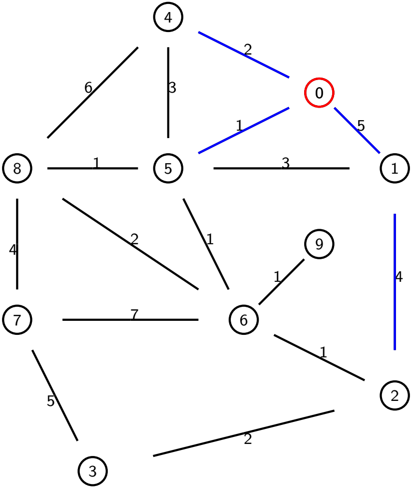
Fronta: 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



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

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, jako v tomto případě s vrcholem $v = 1$, nebo později $v = 3$.

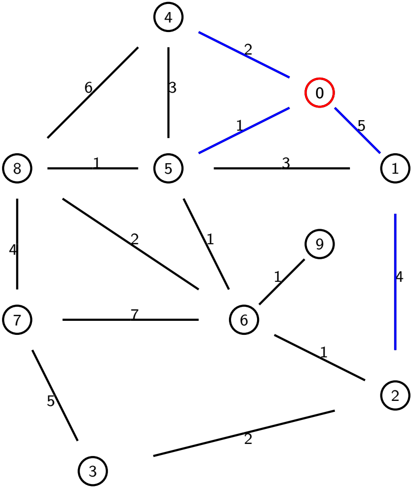
Fronta: 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



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

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, jako v tomto případě s vrcholem $v = 1$, nebo později $v = 3$.

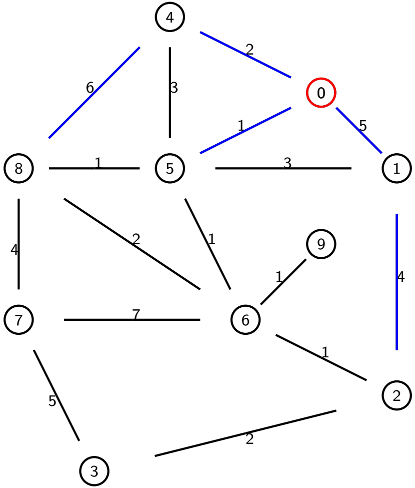
Fronta: 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



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

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, jako v tomto případě s vrcholem $v = 1$, nebo později $v = 3$.

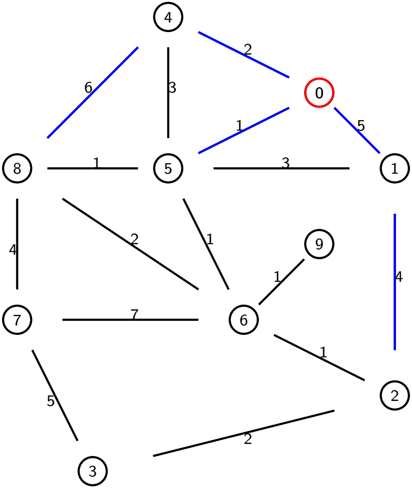
Fronta: 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



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	∞	

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, jako v tomto případě s vrcholem $v = 1$, nebo později $v = 3$.

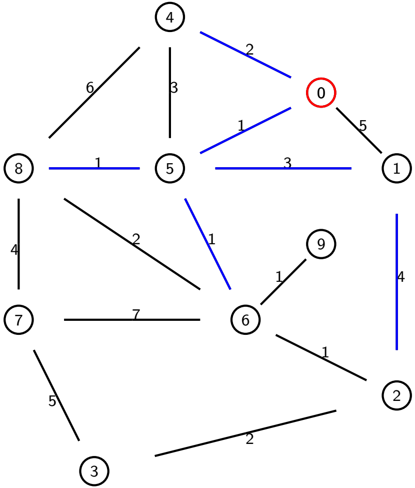
Fronta: 1 4 5 | 2 8 1 6 | 3 7 2 9 | 3



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	∞	

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, jako v tomto případě s vrcholem $v = 1$, nebo později $v = 3$.

Fronta: 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	∞	