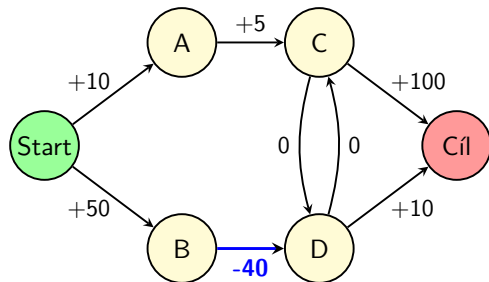


Krokování algoritmů: Bellman-Ford vs. Dijkstra

Orientovaný graf s obousměrnou hranou

Grafové algoritmy

Bellman-Fordův alg. - Inicializace

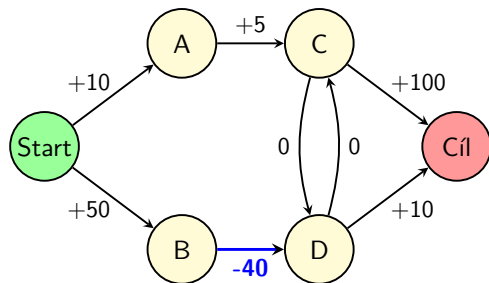


Stav tabulky:

Uzel	Hodnota
S	0
A	∞
B	∞
C	∞
D	∞
Cíl	∞

Začínáme ve Startu. Ostatní uzly jsou nastaveny na nekonečno.

Bellman-Fordův alg. - Krok 1

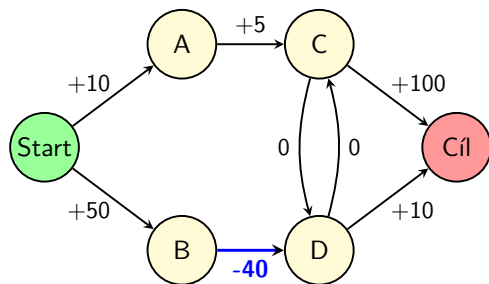


Po 1. iteraci všech hran:

Uzel	Hodnota
S	0
A	10 (přes S)
B	50 (přes S)
C	∞
D	∞
Cíl	∞

Zrelaxovali jsme cesty z uzlu S do A a do B.

Bellman-Fordův alg. - Krok 2

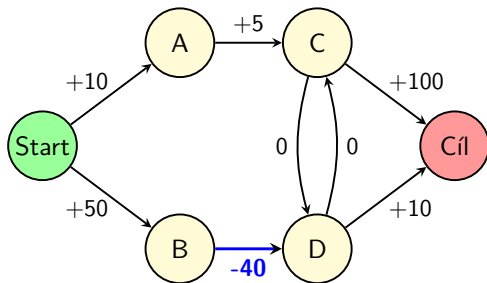


Po 2. iteraci všech hran:

Uzel	Hodnota
S	0
A	10
B	50
C	15 (přes A)
D	10 (přes B)
Cíl	∞

Z A jsme našli cestu do C
($10 + 5 = 15$). Z B jsme našli
zkratku do D ($50 - 40 = 10$).

Bellman-Fordův alg. - Krok 3

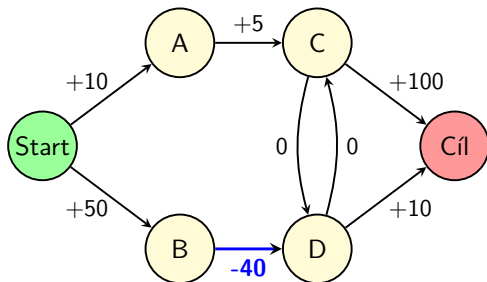


Po 3. iteraci všech hran:

Uzel	Hodnota
S	0
A	10
B	50
C	10 (přes D!)
D	10
Cíl	20 (přes D)

*Obousměrná hrana zafungovala!
Uzel D (10) zrelaxoval uzel C na $10 + 0 = 10$. Do Cíle se dostáváme z D za 20.*

Dijkstrův algoritmus - Krok 1 (Zpracování S)

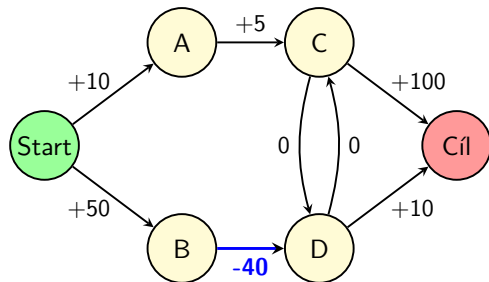


Fronta priorit:

Uzel	Hodnota
A	10 (přes S)
B	50 (přes S)
C	∞
D	∞
Cíl	∞

Vybrali jsme S. Zrelaxujeme sousedy A a B. **Vyřízeno: S.**

Dijkstrův algoritmus - Krok 2 (Zpracování A)

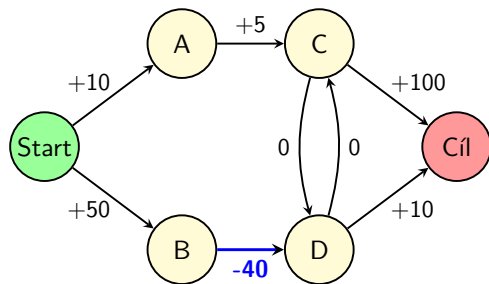


Fronta priorit:

Uzel	Hodnota
B	50 (přes S)
C	15 (přes A)
D	∞
Cíl	∞

*Hladový výběr: Uzel A má nejmenší hodnotu (10). Zrelaxujeme souseda C. **Vyřízeno: S, A.***

Dijkstrův algoritmus - Krok 3 (Zpracování C)

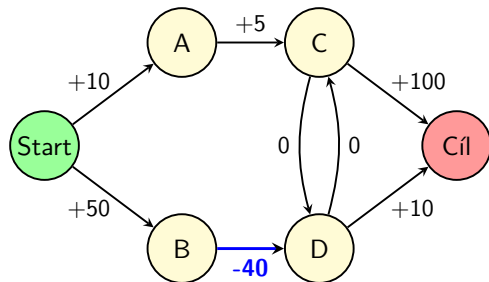


Fronta priorit:

Uzel	Hodnota
D	15 (přes C)
B	50 (přes S)
Cíl	115 (přes C)

Hladový výběr: Uzel C (15).
Zrelaxujeme D ($15 + 0 = 15$) a Cíl ($15 + 100 = 115$). **Vyřízeno: S, A, C.**

Dijkstrův algoritmus - Krok 4 (Zpracování D)

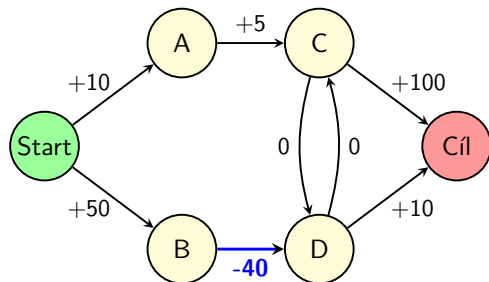


Fronta priorit:

Uzel	Hodnota
Cíl	25 (přes D)
B	50 (přes S)

Bereme D (15). Zrelaxujeme Cíl na 25. Cesta D → C se neprovede, protože uzel C je už vyřízený.

Vyřízeno: S, A, C, D.



Dijkstrův výsledek:

- Cesta: $S \rightarrow A \rightarrow C \rightarrow D \rightarrow \text{Cíl}$
- Celková cena: **25**

Fatální chyba: Algoritmus nyní sice zpracuje uzel B (50) a uvidí hranu do D za -40, ale uzel D už je uzavřen. Dijkstra tak mine správnou cestu (za 20) i případné zlevnění uzlu C zespoda!