Základy teorie grafů

Základy vyšší matematiky (ZMTL)

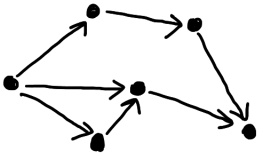
LDF MENDELU

Graf

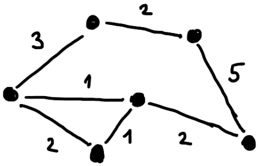
Definice (Graf)

- Grafem (neorientovaným grafem) rozumíme dvojici $G = (V, E)$, kde
  - V je neprázdná konečná množina, jejíž prvky nazýváme vrcholy;
  - E je podmnožina množiny všech nejvýše dvouprvkových neprázdných podmnožin množiny V. Prvky množiny E nazýváme hrany. Hrana spojuje dva vrcholy nebo vrchol sám se sebou (tzv. smyčka).
- Orientovaný graf je graf, kde ke každé hraně dodáme orientaci – jeden z vrcholů prohlásíme za počáteční a jeden z vrcholů za koncový.
- Graf nazýváme hranově ohodnocený, jestliže každé hraně je přiřazeno reálné číslo a vrcholově ohodnocený, jestliže každé každému vrcholu je přiřazeno reálné číslo.
- Graf, který nemá žádné hrany, se nazývá diskrétní.

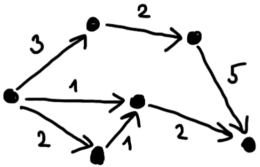
Ukázky grafů



ORIENTOVANÝ GRAF



HRANOVĚ OHODNOCENÝ
NEORIENTOVANÝ GRAF



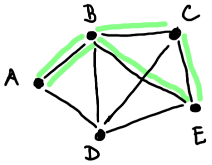
HRANOVĚ OHODNOCENÝ
ORIENTOVANÝ GRAF

Cestování po hranách

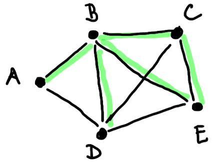
Definice (Sled, tah, cesta)

- Posloupnost hran, které na sebe navazují, se nazývá sled.
- Sled, ve kterém se neopakuje žádná hrana, se nazývá tah.
- Tah, ve kterém se naopakuje žádný vrchol, se nazývá cesta.
- Délkou sledu (tahu, cesty) rozumíme
    - buď počet hran (v grafech bez hranového ohodnocení),
    - nebo součet hranových ohodnocení všech hran v tomto sledu (tahu, cestě) v hranově ohodnoceném grafu.
- Graf, ve kterém mezi každými dvěma vrcholy existuje cesta, se nazývá souvislý.

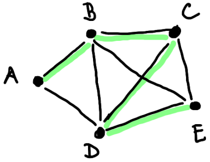
Sled, tah, cesta



SLED A → B → C → E → B → A
(NENÍ TAH ANI CESTA)



TAH A → B → C → E → B → D
(NENÍ CESTA)



CESTA A → B → C → D → E

Kreslení jedním tahem

Definice (Stupeň vrcholu)

Stupněm vrcholu rozumíme počet hran, které z/do tohoto vrcholu vedou.

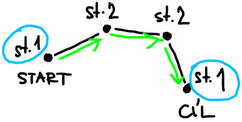
Věta

Graf je možno nakreslit jedním tahem právě tehdy, když je souvislý a všechny vrcholy jsou sudého stupně, nebo má právě dva vrcholy lichého stupně.

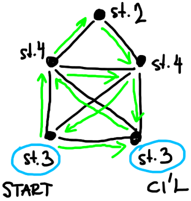
Poznámka

Jsou-li v grafu dva vrcholy lichého stupně, potom tah začíná v jednom z těchto vrcholů a končí ve druhém.

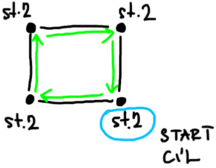
Kreslení jedním tahem



CESTA
2 VRCHOLY LICHÉHO STUPNĚ (START a CÍL)



2 VRCHOLY LICHÉHO STUPNĚ (START a CÍL)
=> LZE KRESLIT JEDNÍM TAHEM



VŠECHNY VRCHOLY SUDÉHO STUPNĚ
=> STAR = CÍL (UZAVŘENÝ TAH)

Nejkratší cesta

- Máme orientovaný nebo neorientovaný graf s nezáporně ohodnocenými hranami. Je zadaný výchozí a cílový bod (vrchol).
- Chceme najít délku nejkratší cesty mezi výchozím a cílovým bodem, tj. najít cestu s nejmenším součtem ohodnocení hran.
- Použití: nejkratší cesta v dopravě, nejkratší cesta ve výrobním procesu.

Dijkstrův algoritmus pro nalezení nejkratší cesty

Algoritmus najde délku nejkratší cesty z výchozího bodu do všech ostatních bodů grafu.

Nechť $G = (V, E)$ je graf, v němž hledáme nejkratší cestu.

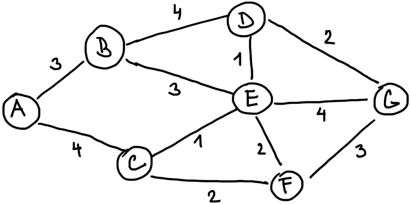
Pro každý vrchol $v \in V$ označme $d[v]$ jako délku dosud nalezené nejkratší cesty, která sem vede z výchozího bodu s. Na začátku má počáteční vrchol s hodnotu $d[s] = 0$, ostatní vrcholy v mají hodnotu $d[v] = \infty$.
Označme Z množinu zpracovaných vrcholů a N množinu nezpracovaných vrcholů.

Následující kroky opakujeme tak dlouho, dokud množina nezpracovaných vrcholů není prázdná.

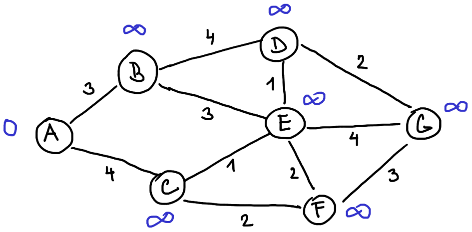
1. Vybereme z množiny N vrchol v s nejmenší vzdáleností od počátku $d[v]$ a tento vrchol přesuneme do množiny Z.
2. Podíváme se, jestli se dá vylepšit vzdálenost sousedních nezpracovaných vrcholů tohoto nově přesunutého vrcholu, pokud uvažujeme cestu přes tento vrchol. Vrcholům u, kterým se vzdálenost dá vylepšit, upravíme hodnotu $d[u]$.

Příklad: nejkratší cesta [1/8]

Najděte nejkratší cestu z A do G:

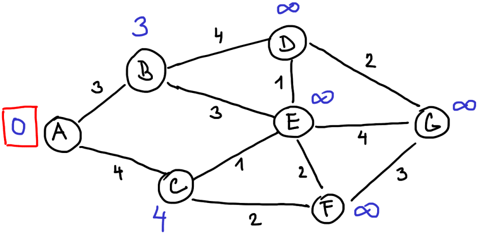


Najděte nejkratší cestu z A do G:



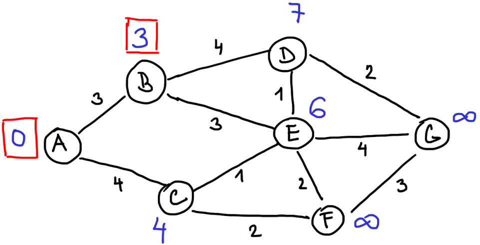
ZPRACOVANÉ	d[A]	d[B]	d[C]	d[D]	d[E]	d[F]	d[G]
$\emptyset$	0	∞	∞	∞	∞	∞	∞

Najděte nejkratší cestu z A do G:



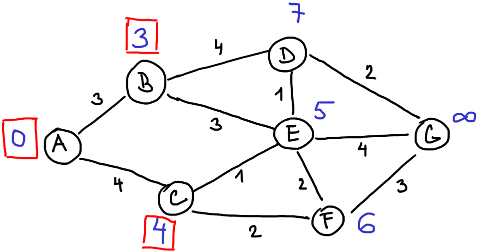
ZPRACOVANÉ	d[A]	d[B]	d[C]	d[D]	d[E]	d[F]	d[G]
∅	$\boxed{0}$	∞	∞	∞	∞	∞	∞
A		3	4	∞	∞	∞	∞

Najděte nejkratší cestu z A do G:



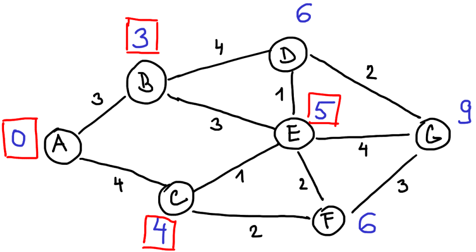
ZPRACOVANÉ	d[A]	d[B]	d[C]	d[D]	d[E]	d[F]	d[G]
∅	$\boxed{0}$	∞	∞	∞	∞	∞	∞
A		$\boxed{3}$	4	∞	∞	∞	∞
A, B			4	7	6	∞	∞

Najděte nejkratší cestu z A do G:



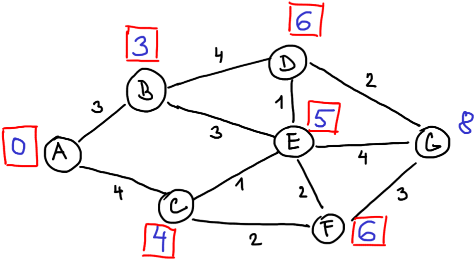
ZPRACOVANÉ	d[A]	d[B]	d[C]	d[D]	d[E]	d[F]	d[G]
$\emptyset$	$\boxed{0}$	∞	∞	∞	∞	∞	∞
A		$\boxed{3}$	4	∞	∞	∞	∞
A, B			$\boxed{4}$	7	6	∞	∞
A, B, C				7	5	6	∞

Najděte nejkratší cestu z A do G:



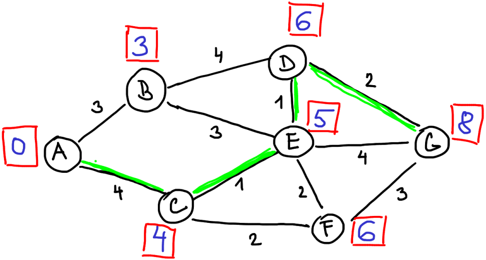
ZPRACOVANÉ	d[A]	d[B]	d[C]	d[D]	d[E]	d[F]	d[G]
∅	$\boxed{0}$	∞	∞	∞	∞	∞	∞
A		$\boxed{3}$	4	∞	∞	∞	∞
A, B			$\boxed{4}$	7	6	∞	∞
A, B, C				7	$\boxed{5}$	6	∞
A, B, C, E				6		6	9

Najděte nejkratší cestu z A do G:



ZPRACOVANÉ	d[A]	d[B]	d[C]	d[D]	d[E]	d[F]	d[G]
$\emptyset$	$\boxed{0}$	∞	∞	∞	∞	∞	∞
A		$\boxed{3}$	4	∞	∞	∞	∞
A, B			$\boxed{4}$	7	6	∞	∞
A, B, C				7	$\boxed{5}$	6	∞
A, B, C, E				$\boxed{6}$		$\boxed{6}$	9
A, B, C, D, E, F							8

Najděte nejkratší cestu z A do G:



ZPRACOVANÉ	d[A]	d[B]	d[C]	d[D]	d[E]	d[F]	d[G]
∅	$\boxed{0}$	∞	∞	∞	∞	∞	∞
A		$\boxed{3}$	4	∞	∞	∞	∞
A, B			$\boxed{4}$	7	6	∞	∞
A, B, C				7	$\boxed{5}$	6	∞
A, B, C, E				$\boxed{6}$		$\boxed{6}$	9
A, B, C, D, E, F							$\boxed{8}$

Les, strom, kostra

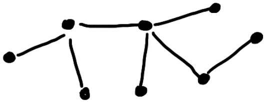
Definice

- Tah, ve kterém splývá počáteční a koncový bod se nazývá cyklus (v orientovaných grafech) nebo kružnice (v neorientovaných grafech).
- Graf, který neobsahuje kružnici, se nazývá les. Souvislý les (tj. souvislý graf, který neobsahuje kružnici) se nazývá strom.
- Je-li $G = (V, E)$ graf a jsou-li V' a E' podmnožiny množin V a E takové, že $G' = (V', E')$ je grafem, nazývá se G' podgrafem grafu G. Je-li $V' = V$ (podgraf obsahuje všechny vrcholy), nazývá se G' faktorem grafu G.
- Faktor, který je stromem, se nazývá kostra. Kostra grafu je tedy podgraf, který obsahuje všechny vrcholy a takové hrany, aby výsledný podgraf byl souvislý a neobsahoval kružnice.

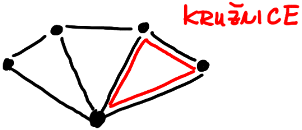
Věta

Počet hran stromu (a také kostry) v neorientovaném grafu je o jednu nižší než počet vrcholů.

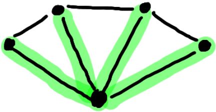
Strom a kostra



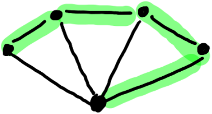
STROM



NENÍ STROM



KOSTRA



JINÁ KOSTRA
STEJNÉHO GRAFU

Minimální kostra

- Máme souvislý hranově ohodnocený graf.
- Chceme najít minimální kostru, tj. kostru s nejmenším součtem ohodnocení hran.
- Použití: minimalizace celkových nákladů na propojení míst (elektrickou sítí, počítačovou sítí, optimální postup při prohrnování sněhu mezi městy, ...)

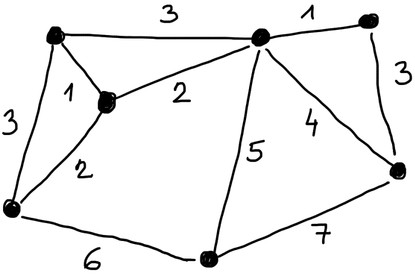
Borůvkův-Kruskalův algoritmus pro nalezení minimální kostry

1. Všechny hrany grafu uspořádáme podle velikosti do neklesající posloupnosti.
2. Začneme od podgrafu, který nemá žádnou hranu a má všechny vrcholy.
3. Postupně přidáváme do podgrafu hrany podle velikosti tak, aby se propojily všechny vrcholy, ale nevznikla kružnice.

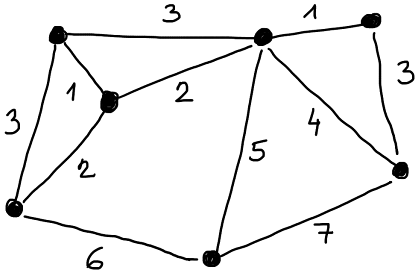
Algoritmus končí v konečném čase. Pro velké grafy je při zkušebním přidání každé hrany nutné použít algoritus, který zjistí, zda se v grafu objeví kružnice.

Příklad: minimální kostra [1/9]

Najděte minimální kostru:

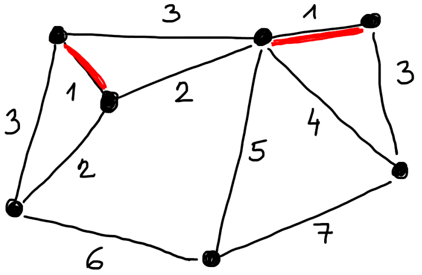


Najděte minimální kostru:



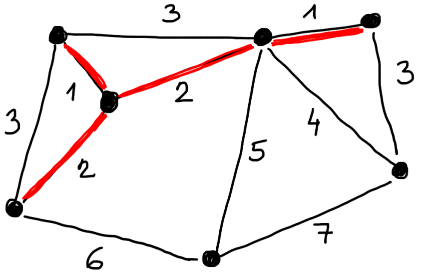
HRANY: 1, 1, 2, 2, 3, 3, 3, 4, 5, 6, 7

Najděte minimální kostru:



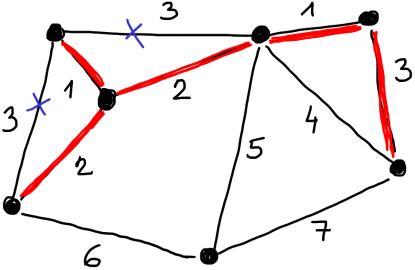
HRANY: 1, 1, 2, 2, 3, 3, 3, 4, 5, 6, 7
✓ ✓

Najděte minimální kostru:



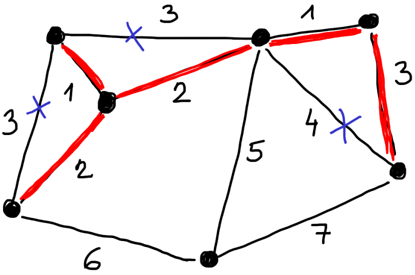
HRANY: 1, 1, 2, 2, 3, 3, 3, 4, 5, 6, 7

Najděte minimální kostru :



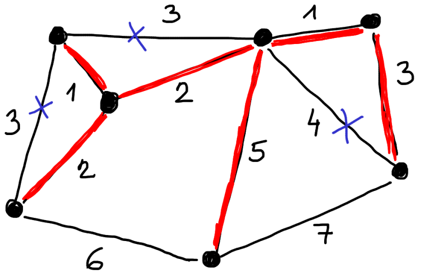
HRANY : 1, 1, 2, 2, 3, 3, 3, 4, 5, 6, 7

Najděte minimální kostru:



HRANY: 1, 1, 2, 2, 3, 3, 3, 4, 5, 6, 7

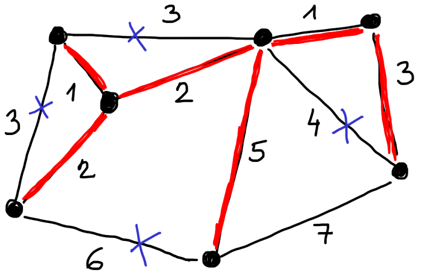
Najděte minimální kostru:



HRANY:

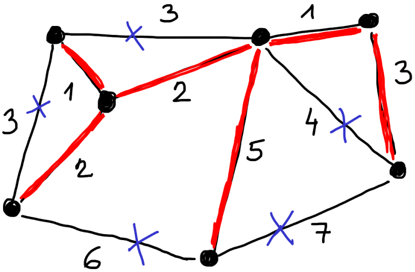
1, 1, 2, 2, 3, 3, 3, 4, 5, 6, 7

Najděte minimální kostru:



HRANY: 1, 1, 2, 2, 3, 3, 3, 4, 5, 6, 7

Najděte minimální kostru:



HRANY: 1, 1, 2, 2, 3, 3, 3, 4, 5, 6, 7
✓ ✓ ✓ ✓ ✓ × × × ✓ × ×

Kritická cesta (CPM)

- Máme orientovaný hranově ohodnocený graf bez cyklů odpovídající sledu činností na projektu. Hrany představují jednotlivé činnosti a dobu jejich trvání. Vrcholy jsou fáze projektu (vyjadřují začátek nebo konec určité činnosti).

- Chceme najít minimální dobu trvání projektu, časové rezervy jednotlivých činností (nejdříve a nejpozději možný termín), harmonogram činností. Pro každý vrchol chceme najít časy $t_{\min}$ a $t_{\max}$, kde
  - $t_{\min}$ je nejdříve možný termín, tj. čas, kdy se nejdříve může projekt dostat do tohoto stavu,
  - $t_{\max}$ je nejpozdější přípustný termín, tj. čas, kdy nejpozději musí projekt dospět do tohoto stavu, pokud nemá dojít k prodloužení doby trvání celého projektu.

- Použití: plánování výrobního procesu, detekce kritických činností ve výrobním procesu.

Algoritmus pro nalezení kritické cesty

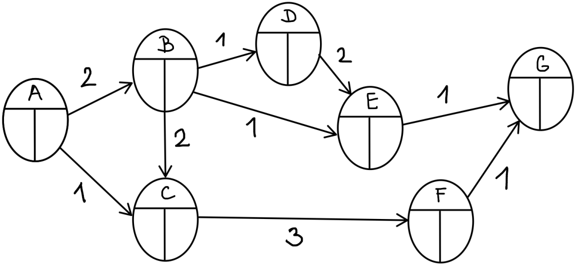
Algoritmus má dvě části:

1. Hledáme $t_{\min}$ všech vrcholů. Prvnímu vrcholu (tj. vrcholu, do kterého nevedou žádné hrany) přiřadíme $t_{\min} = 0$. Pokud do vrcholu v vedou pouze hrany z vrcholů se známým $t_{\min}$, určíme $t_{\min}$ vrcholu v jako maximum z čísel, která získáme jako součet hranového ohodnocení hrany mířící do v a $t_{\min}$ počátečního bodu této hrany. Pokud je projekt smysluplně navržen, jsme takto schopni určit $t_{\min}$ všech vrcholů.
2. Hledáme $t_{\max}$ všech vrcholů. Poslednímu vrcholu (tj. vrcholu, z kterého nevedou žádné hrany) nastavíme $t_{\max} = t_{\min}$. Pokud z vrcholu v vedou pouze hrany do vrcholů se známým $t_{\max}$, určíme $t_{\max}$ vrcholu v jako minimum z čísel, která získáme jako rozdíl $t_{\max}$ koncového bodu hrany a hranového ohodnocení hrany, přičemž uvažujeme všechny hrany vycházející z v.

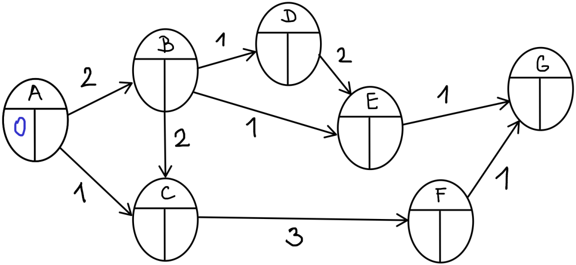
Vrcholy, pro které je $t_{\min} = t_{\max}$, jsou vrcholy, v nichž nesmí nastat zpoždění. Hrany spojující tyto vrcholy tvoří kritickou cestu. Činnosti odpovídající těmto hranám se nesmí zpozdit.

Příklad: kritická cesta [1/14]

Najděte kritickou cestu:



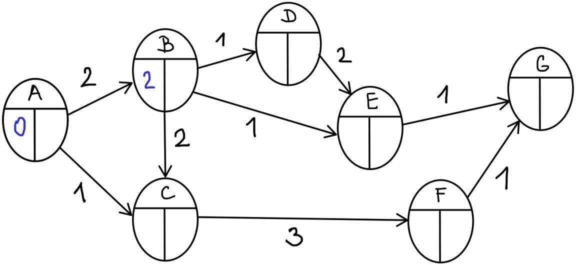
Najděte kritickou cestu:



$\underline{t_{min}}$: A: 0

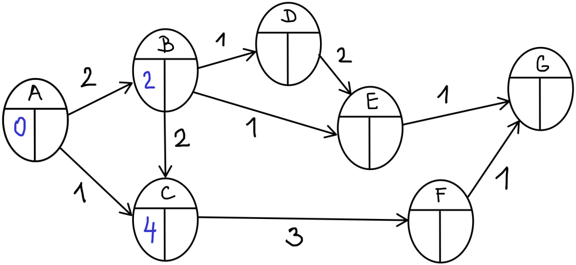
Příklad: kritická cesta [3/14]

Najděte kritickou cestu:



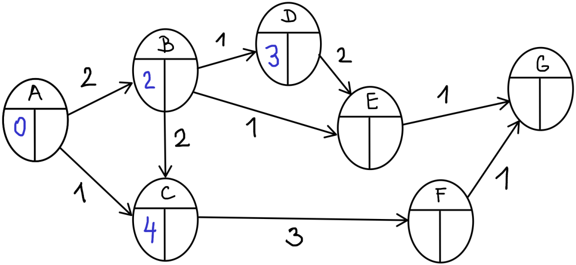
$\underline{t_{min}}$: A: 0, B : 0 + 2 = 2

Najděte kritickou cestu:



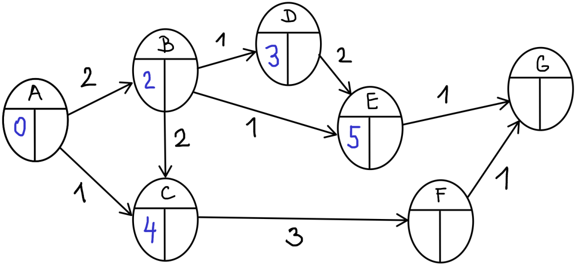
$\underline{t_{min}}$: $A: 0$, $B: 0+2=2$, $C: \max\{0+1, 2+2\} = 4$

Najděte kritickou cestu:



$\underline{t_{min}}$: A: 0, B: 0+2=2, C: max{0+1, 2+2} = 4, D: 2+1 = 3

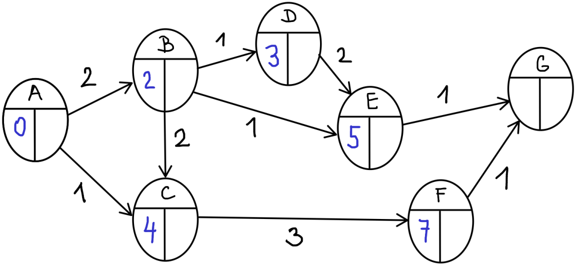
Najděte kritickou cestu:



$\underline{t_{min}}$: A: 0, B: $0+2=2$, C: $\max\{0+1, 2+2\}=4$, D: $2+1=3$,

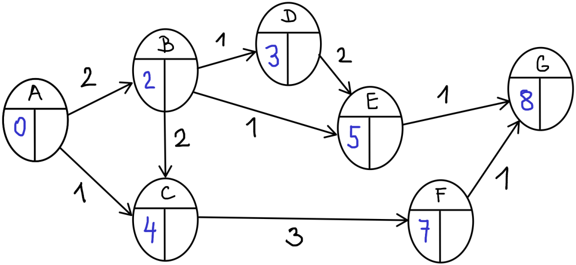
E: $\max\{2+1, 3+2\}=5$

Najděte kritickou cestu:



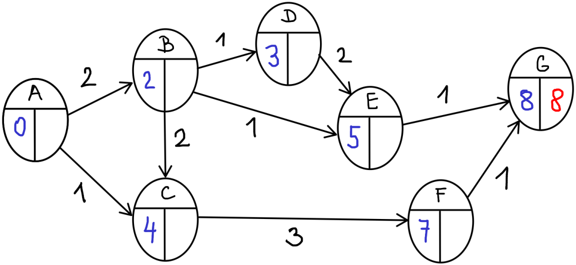
t_{min}: A: 0, B: 0 + 2 = 2, C: max{0 + 1, 2 + 2} = 4, D: 2 + 1 = 3,
E: max{2 + 1, 3 + 2} = 5, F: 4 + 3 = 7

Najděte kritickou cestu:



$\underline{t_{min}}$: A: 0, B: 0+2=2, C: max{0+1, 2+2} = 4, D: 2+1 = 3,
E: max{2+1, 3+2} = 5, F: 4+3 = 7, G: max{5+1, 7+1} = 8

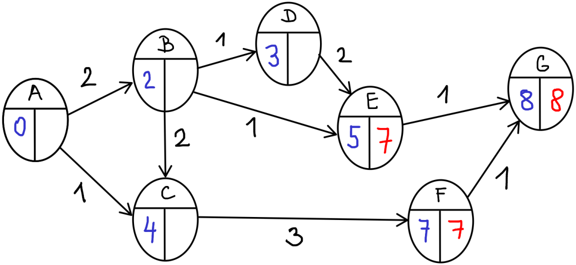
Najděte kritickou cestu:



t_{min}: A: 0, B: $0+2=2$, C: $\max\{0+1, 2+2\}=4$, D: $2+1=3$,
E: $\max\{2+1, 3+2\}=5$, F: $4+3=7$, G: $\max\{5+1, 7+1\}=8$

t_{max}: G: 8

Najděte kritickou cestu:

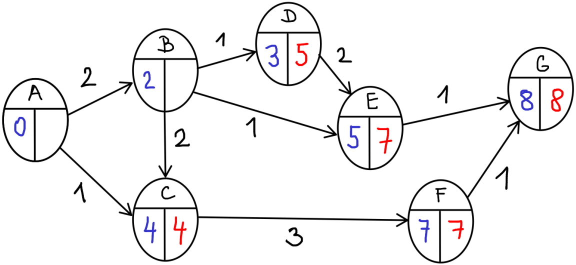


t_{min}: A: 0, B: 0+2=2, C: max{0+1, 2+2} = 4, D: 2+1 = 3,
E: max{2+1, 3+2} = 5, F: 4+3 = 7, G: max{5+1, 7+1} = 8

t_{max}: G: 8, F: 8−1 = 7, E: 8−1 = 7

Příklad: kritická cesta [11/14]

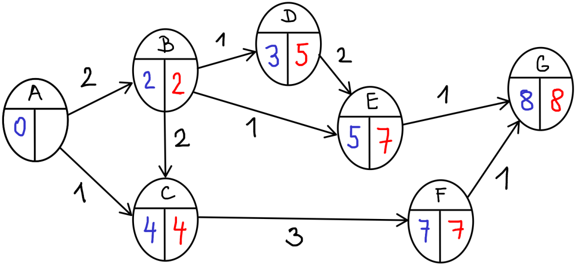
Najděte kritickou cestu:



t_{min}: A: 0, B: 0+2=2, C: max{0+1, 2+2} = 4, D: 2+1 = 3,
E: max{2+1, 3+2} = 5, F: 4+3 = 7, G: max{5+1, 7+1} = 8

t_{max}: G: 8, F: 8−1 = 7, E: 8−1 = 7, D: 7−2 = 5, C: 7−3 = 4

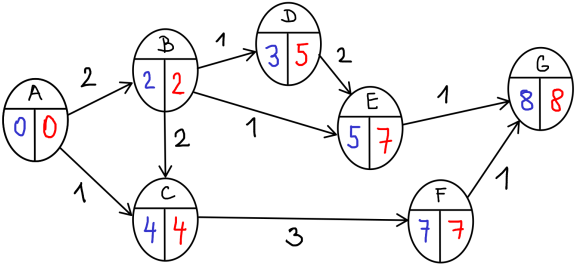
Najděte kritickou cestu:



t_{min}: A: 0, B: $0+2=2$, C: $\max\{0+1, 2+2\}=4$, D: $2+1=3$,
E: $\max\{2+1, 3+2\}=5$, F: $4+3=7$, G: $\max\{5+1, 7+1\}=8$

t_{max}: G: 8, F: $8-1=7$, E: $8-1=7$, D: $7-2=5$, C: $7-3=4$,
B: $\min\{5-1, 7-1, 4-2\}=2$

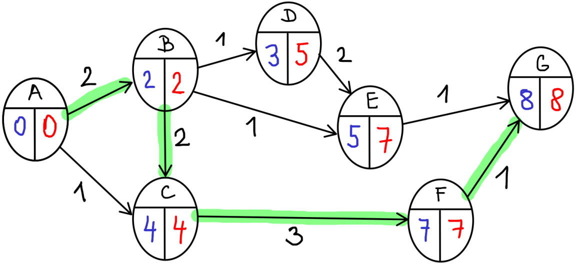
Najděte kritickou cestu:



t_{min}: A: 0, B: $0+2=2$, C: $\max\{0+1, 2+2\}=4$, D: $2+1=3$,
E: $\max\{2+1, 3+2\}=5$, F: $4+3=7$, G: $\max\{5+1, 7+1\}=8$

t_{max}: G: 8, F: $8-1=7$, E: $8-1=7$, D: $7-2=5$, C: $7-3=4$,
B: $\min\{5-1, 7-1, 4-2\}=2$, A: $\min\{2-2, 4-1\}=0$

Najděte kritickou cestu:



tmin: A: 0, B: 0+2 = 2, C: max{0+1, 2+2} = 4, D: 2+1 = 3,
E: max{2+1, 3+2} = 5, F: 4+3 = 7, G: max{5+1, 7+1} = 8

tmax: G: 8, F: 8−1 = 7, E: 8−1 = 7, D: 7−2 = 5, C: 7−3 = 4,
B: min{5−1, 7−1, 4−2} = 2, A: min{2−2, 4−1} = 0