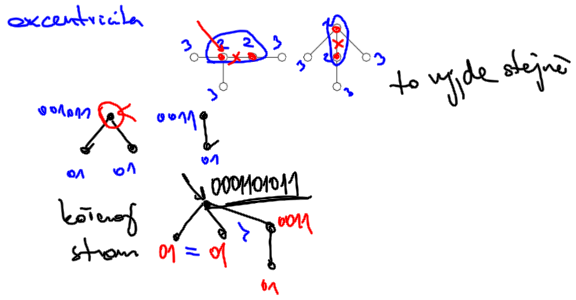
10. demonstrační cvičení

Příklad 66. *Pomocí kódu dokažte izomorfismus stromů:*



12 2-16:47

Příklad 67. ***Huffmanovo kódování*** *Pracujeme s pěstěnými binárními stromy, kde máme navíc každou hranu obarvenou některým symbolem z dané výstupní abecedy A (často $A = \{0,1\}$). Kódovými slovy C jsou slova nad abecedou A, na která převádíme symboly vstupní abecedy. Našim úkolem je reprezentovat daný text pomocí vhodných kódových slov nad výstupní abecedou.*

Je snadno vidět, že je užitečné chtít, aby seznam kódových slov byl bezprefixový (v opačném případě může nastat problém s dekódováním).

Ke konstrukci binárních prefixových kódů (tj. nad abecedou $A = \{0,1\}$) využijeme binárních stromů. Označíme-li hrany vycházející z každého uzlu 0, resp. 1, a označíme-li navíc listy stromu symboly vstupní abecedy, dostaneme prefixový kód nad A pro tyto symboly zřetězením označení hran na cestě z kořene do příslušného listu.

Takto vytvořený kód je zřejmě prefixový. Uděláme-li tuto konstrukci navíc tak, abychom odrazili četnost symbolů vstupní abecedy v kódovaném textu, dosáhneme tak dokonce bezztrátové komprese dat.

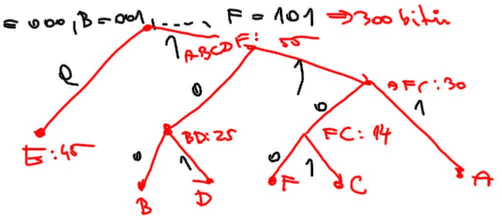
Nechť M je seznam četností symbolů vstupní abecedy v textu. Algoritmus postupně zkonstruuje optimální binární strom (tzv. minimum-weight binary tree) a přiřazení symbolů listům.

- *Vyber dvě nejmenší četnosti w_1, w_2 z M. Vyrob strom se dvěma listy označenými příslušnými symboly a kořenem označeným $w_1 + w_2$, odeber z M hodnoty w_1, w_2 a nahraď je hodnotou $w_1 + w_2$.*
- *Tento krok opakuj; pouze v případě, že vybraná hodnota z M je součtem, pak nevyráběj nový list, ale "připoj"příslušný již existující podstrom.*
- *Kód každého symbolu urči cestou od kořene (např. vlevo="0", vpravo="1").*

Nalezněte Huffmanův kód pro vstupní abecedu s frekvencemi `['A':16, 'B':13, 'C':9, 'D':12, 'E':45, 'F':5]`.

12 2-16:47

A = 000, B = 001, ..., F = 101 ⇒ 300 bitů



A: 16	111	3·16
B: 13	100	3·13
C: 9	1101	4·9
D: 12	101	3·12
E: 45	0	1·45
F: 5	1100	4·5

$3 \cdot 41 + 4 \cdot 14 + 1 \cdot 45 =$

$= \underline{\underline{224}}$

12 2-16:55

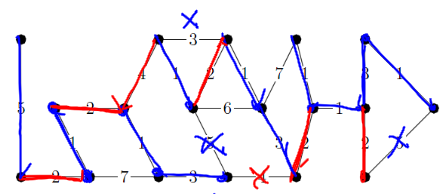
Příklad 68. *Najděte minimální kostru grafu na obrázku pomocí*

(a) Kruskalova

(b) Jarníkova (Primova) algoritmu.

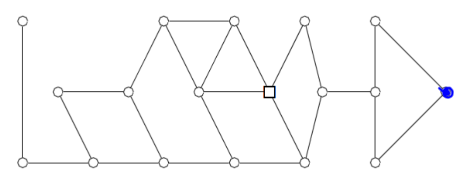
(c) Borůvkova algoritmu.

Jak se výpočet změní, pokud hledáme maximální kostru?

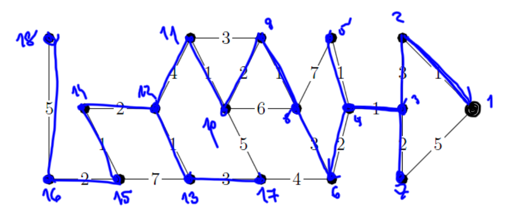


17 hran ✓ ⇒ min. kostra

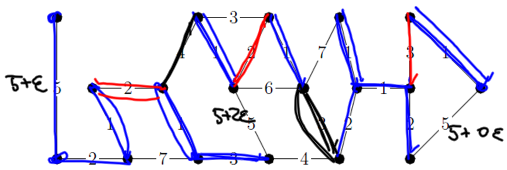
12 2-17:20



12 2-17:21

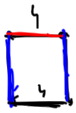


12 2-17:31



1. krok
2. krok

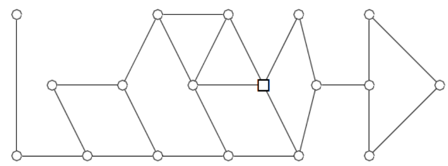
Pozor!



může vzniknout kružnice!

⇒ Borůvka pouze na prosté ohodnocení

12 2-17:35



12 2-17:21

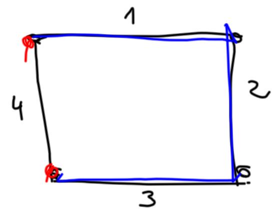
Příklad 69. *Najděte minimální kostru pomocí maticové verze Jarník-Primova algoritmu v grafu zadaném maticí ohodnocení hran*

$$\begin{pmatrix} - & 12 & - & 16 & - & - & - & 13 \\ 12 & - & 16 & - & - & - & 14 & - \\ - & 16 & - & 12 & - & 14 & - & - \\ 16 & - & 12 & - & 13 & - & - & - \\ - & - & - & 13 & - & 14 & - & 15 \\ - & - & 14 & - & 14 & - & 15 & - \\ - & 14 & - & - & - & 15 & - & 14 \\ 13 & - & - & - & 15 & - & 14 & - \end{pmatrix} \begin{matrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{matrix}$$

12 2-16:52

Příklad 70. *Udejte příklad, na němž ukážete nefunkčnost "naivního" algoritmu pro nalezení nejkratší cesty:*

1. *nalezněte minimální kostru,*
2. *za nejkratší cestu prohlásíme tu jedinou cestu spojující zadané 2 vrcholy, která leží v nalezení minimální kostře.*



$4 < 1+2+3$

12 2-16:55

Příklad 71. *Maximální párování v bipartitním grafu*

- *perfektní párování v obecném grafu (viz CPP - chinese postman problem)*
- *Tutteho věta – graf má perfektní párování, právě když pro každé $S \subset V$ počet lichých komponent $G \setminus S$ nepřevyšuje $|S|$.*

Maďarský algoritmus (König, Egerváry, Kuhn)
Označujme kvůli jednoduchosti popisu vrcholy jedné skupiny v bipartitním grafu jako **červené**, *vrcholy druhé skupiny jako* **modré**. *Budeme předpokládat, že vrcholy jsou obarveny tak, že počet červených vrcholů nepřevyšuje počet modrých.*

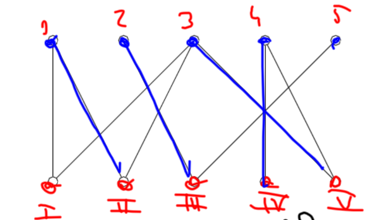
Nechť je dán bipartitní graf $G = (V, E)$ a párování $M \subseteq E$ (množina hran). *M-alternující cestou v G nazveme takovou cestu v G, jejíž hrany tvoří střídavě hrany z M a z $E \setminus M$. M-rozšiřující cestou v G nazveme M-alternující cestu, která spojuje dosud nepřiřazený červený vrchol u s dosud nepřiřazeným modrým vrcholem v.*

12 2-17:20

1. *Nalezneme libovolné párování M. Označíme všechny červené vrcholy jako přípustné.*
2. *Vyberme některý dosud nepřiřazený přípustný červený vrchol v (pokud neexistuje, jsme hotovi) a nalezněme pro něj (např. prostřednictvím prohledávání do hloubky) M-alternující strom s kořenem ve v (pokud neexistuje, označme v jako nepřípustný a proces opakujme s jiným vrcholem). Pokud strom obsahuje nějakou M-rozšiřující cestu, odstraníme M-hrany v této cestě z M a ostatní hrany této cesty do M přidáme. Označmě všechny červené vrcholy jako přípustné a proces opakujme.*

V bipartitním grafu na obrázku najděte takovou maximální podmnožinu hran, že žádné dvě z nich nesdílejí společný vrchol.

12 2-16:53



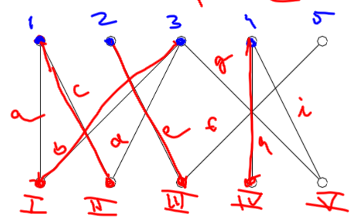
prohl. do hloubky

I − 1 − ... nerozšiřuje M - rozšiř. cesta

⇒ konec

12 2-16:55

zleva doprava



M = ∅
M = {a}
M = {b, c}
M = {b, c, e}
M = {b, c, e, h}

I − 1 ; II − 1 − I − 3 ; III − 2 ; IV − 4

V − 3 − I − 1 − II − 3 ↯

V − 4 − IV ↯ } ⇒ konec

max. |párování| = 4

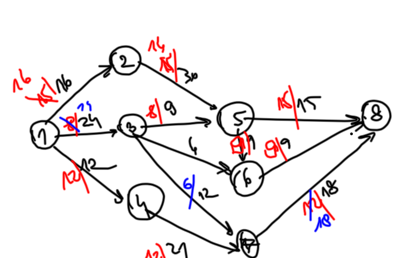
12 2-18:09

12 2-18:08

Příklad 72. *Určete hodnotu maximálního toku a najděte minimální řez v síti dané maticí kapacit A, kde vrchol 1 je zdroj a vrchol 8 stok.*

$$A = \begin{pmatrix} - & 16 & 24 & 12 & - & - & - & - \\ - & - & - & - & 30 & - & - & - \\ - & - & - & - & 9 & 6 & 12 & - \\ - & - & - & - & - & - & 21 & - \\ - & - & - & - & - & 9 & - & 15 \\ - & - & - & - & - & - & - & 9 \\ - & - & - & - & - & - & - & 18 \\ - & - & - & - & - & - & - & - \end{pmatrix}$$

12 2-16:53



zlepšující (polo)cesta

1 − 2 − 5 − 8 r. 15
1 − 2 − 5 − 6 − 8 r. 1
1 − 3 − 5 − 6 − 8 r. 8
1 − 4 − 7 − 8 r. 12

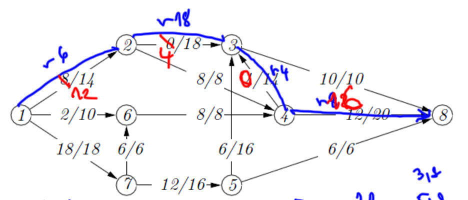
$1 \xrightarrow{16} 3 \xrightarrow{12} 7 \xrightarrow{6} 8$ r. 6

tok 42.

min. řez: [5,8] 15, [6,8] 9, [7,8] 18 Σ 42

12 2-18:21

Příklad 73. *Na obrázku je uveden tok v dané síti (čísla f/c udávají současný tok a kapacitu dané hrany). Zjistěte, je-li uvedený tok maximální, pokud ano, své tvrzení zdůvodněte. Pokud maximálním tokem není, maximální tok najděte a svůj postup podrobně popište. Uveďte některý minimální řez v dané síti.*



tok: 28 + 4 = 32

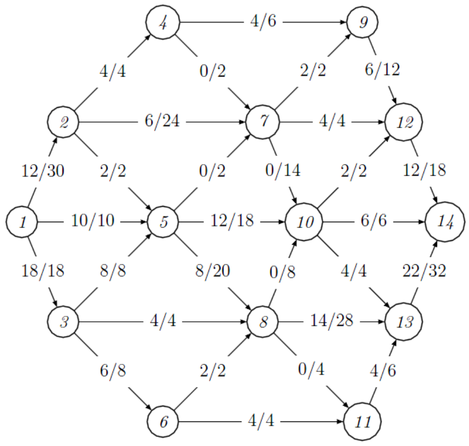
řez 32

zlepšující polocesta $1 \xrightarrow{6} 2 \xrightarrow{18} 3 \xleftarrow{4} 4 \xrightarrow{8} 8$

r. 4

12 2-16:53

Příklad 74. *Nalezněte maximální tok a minimální řez v síti na obrázku (zdroj=1,stok=14).*



12 2-16:54

Příklad 75. *Pomocí Ford-Fulkersonova algoritmu (prohledávání do hloubky, vrcholy volte vzestupně podle čísel) nalezněte maximální tok v síti na množině vrcholů $\{1,2,\dots,9\}$ se zdrojem 1 a stokem 9. Nalezněte minimální řez v této síti. Jednotlivé kroky svého postupu podrobně popište. Hrany $e \in E$,* ***dolní*** *omezení, resp.* ***horní*** *omezení na tok danou hranou ($d(e)$, resp. $h(e)$) a současný tok na dané hraně $f(e)$ jsou uvedeny v tabulce:*

e	$d(e)$	$h(e)$	$f(e)$
(1,2)	0	6	0
(1,3)	0	6	0
(1,6)	0	4	0
(2,3)	0	2	0
(2,4)	0	3	0
(3,4)	0	4	0
(3,5)	0	4	0
(4,5)	3	5	4
(4,8)	0	3	0

e	$d(e)$	$h(e)$	$f(e)$
(5,1)	0	3	0
(5,6)	0	6	0
(5,7)	0	5	4
(5,8)	0	5	0
(6,9)	0	5	0
(7,4)	1	6	4
(7,9)	0	3	0
(8,9)	0	9	0

12 2-16:54

12 2-17:52