

Hradla:

$\mathcal{E} \dots$ konečná abeceda

$h: \mathcal{E}^k \rightarrow \mathcal{E}$

L arita

konstanty

hradel

Nulové hradlo: 2 (0, 1)

Umírní hradlo: 2 (id_1, id_2)

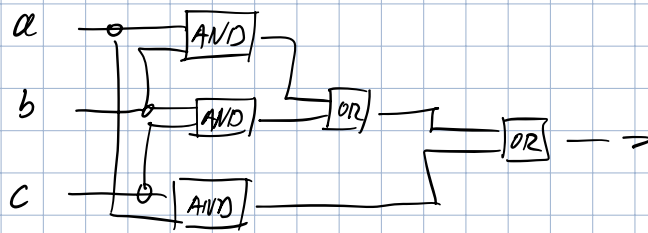
Binární hradla: AND, OR...

Hradlový graf:

MAJORITY

vstupy
výstupy
hradla $\rightarrow$ funkce, arita

vstupní porty
hradla
výstupní port



hruz:

vstup: $\deg^{in} = 0$

výstup: $\deg^{in} = 1$

hradlo: $\deg^{in} = \text{arita}$, $\deg^{out} > 0$

- každé hradlo má odeslané vstupy,

- graf musí být acyklický

- každé hradlo musí mít funkci

graf je DAG.

Výpočet:

- běží po tabulkách

$f = 0$: výstup mají vstupní porty a umírní hradla

$f > 0$: výstup vyjdají hradla, co mají řešené vstupy def.

rozdělení na vstupy $\rightarrow$

Arita hradel musí být omezena.

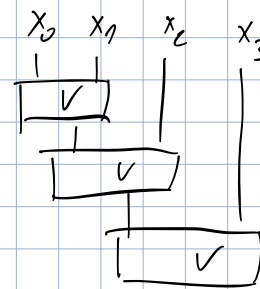
$V_i :=$ vstupy, co poprvé dají výstup v čase i .

Existence 1 v n -bitovém vstupu.

Výstup: $x_0 \vee x_1 \dots x_{n-1}$

Obecně chceme čas $O(\log^k n)$

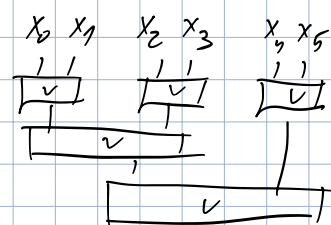
Triviální:



$h: O(n)$

$v: O(n)$

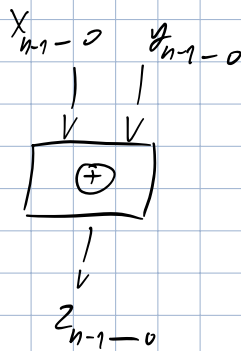
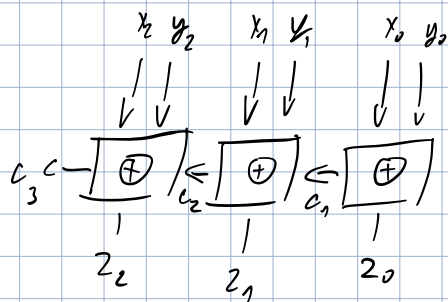
Přívání:



$h: O(\log n)$

$v: O(n)$

Binární sčítáček:



$$c_0 = 0$$

$$z_i = x_i \text{ XOR } y_i \text{ XOR } c_i$$

$$c_{i+1} = \text{maj}(x_i, y_i, c_i) = (x_i \wedge y_i) \vee (x_i \wedge c_i) \vee (y_i \wedge c_i)$$

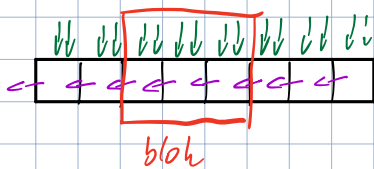
$$h: \Theta(n)$$

$$v: \Theta(n)$$

To je ale schvácení

Přímá bych větší přímou dopředu, může být výpočet v konstantním čase.

- Můžeme uvažovat obě možnosti pro každý bit zespodu.



chování bloku: $f: c_{in} \rightarrow c_{out}$

- identity = <

- negace = nikdy nemůže

- konst 0 = 0

- konst 1 = 1

↳ jednotkový blok nemůže,

složení takových bloků nepřetvoří negaci

1-bit blok

	x	
y	0	1
0	<	<
1	<	1

$$p = x \oplus y$$

$$q = x$$

	0	1	<
0	0	0	0
1	1	1	1
<	0	1	<

$$p_B = p_n \wedge p_e$$

$$q_B = (p_n \wedge q_e) \vee (\neg p_n \wedge q_n)$$

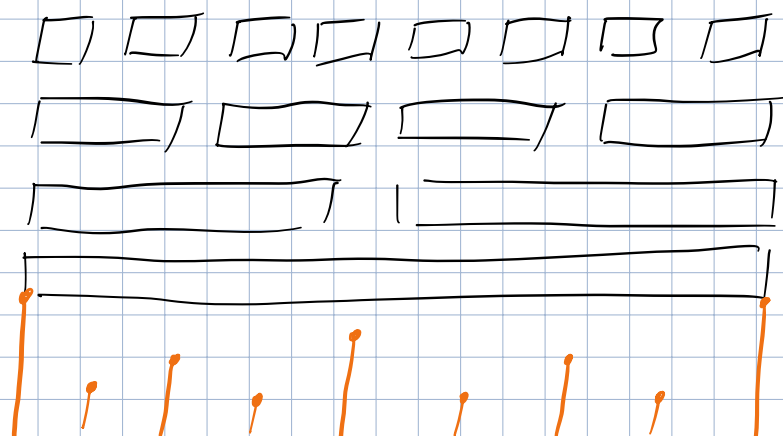
$$p \ q$$

$$< \ (1, *)$$

$$0 \ (0, 0)$$

$$1 \ (0, 1)$$

Jak udělat binární sčítáček? Pomocí kombinovaných bloků:



chování kombinovaných bloků

$$h: \Theta(\log n)$$

$$v: \Theta(n)$$

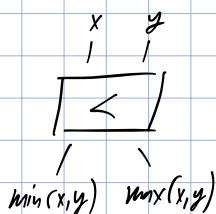
zjednodušené bloky

$$h: \Theta(\log n)$$

$$v: \Theta(n)$$

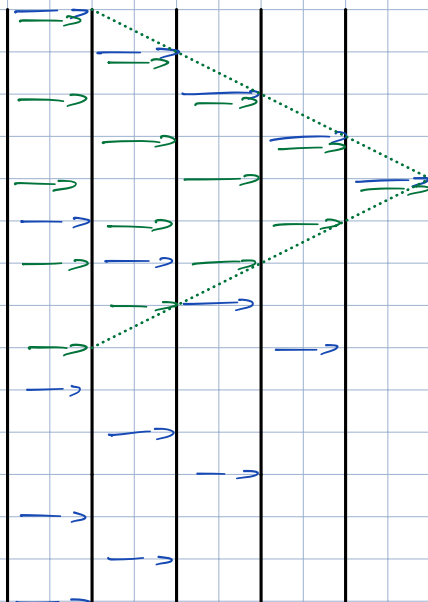
Trídící síť:

Comparator:



SWAP: Síť se mění.

Bubble Sort:



Tohle ale není paralelní způsob.

Tohle je paralelní způsob.

Hloubka $\Theta(n)$ $\rightarrow$ lineární paralelní čas
Velikost $\Theta(n^2)$ $\rightarrow$ logaritmus $\Theta(\log^2 n)$

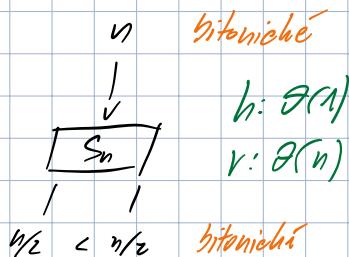
Merge Sort:

Df: Posl. $x_0 \dots x_{n-1}$ je bitonické $\equiv$

$\exists j: x_i < x_{i+1} < \dots < x_{i+j} > x_{i+j+1} > \dots > x_{i+n-1}$
(indexujeme mod n)



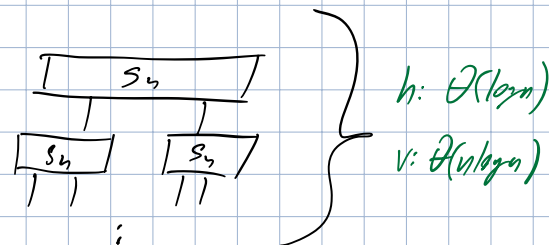
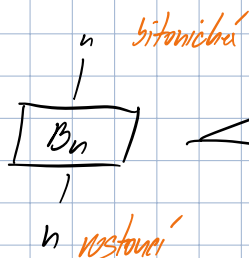
Separator:



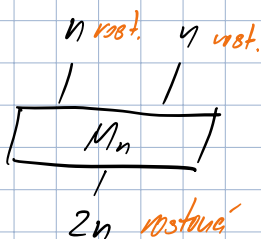
mačí epit dvě posloupnosti, které jsou bitonické, ale hlavní věci jsou seřazené.

Polovinu kudy opalujeme za sebou řadit, dostaneme seřazené jednopuhly $\rightarrow$ princip bitonické třídičky.

(Bitonická) třídička:



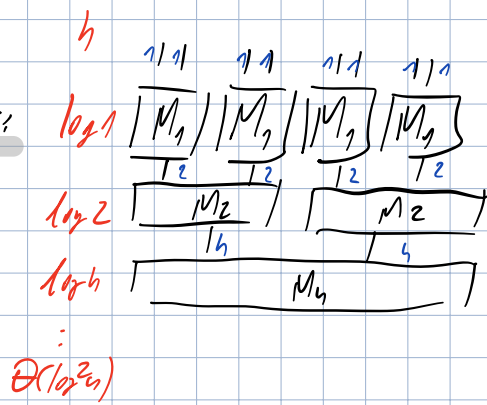
Sléváčka:



rovnou jednou spojíme dvěma otáčením a provedeme bitonickou třídičkou.

$h: \Theta(\log n)$
 $v: \Theta(n \log n)$

Merge Sort:



Velikost
vstupu / výstupu

$$h: \Theta(\log^2 n)$$

$$V: \Theta(n \cdot \log^2 n)$$