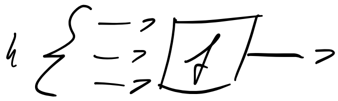
Hradla:

Σ ... konečná abeceda



$f: \Sigma^n \to \Sigma$

└ arita

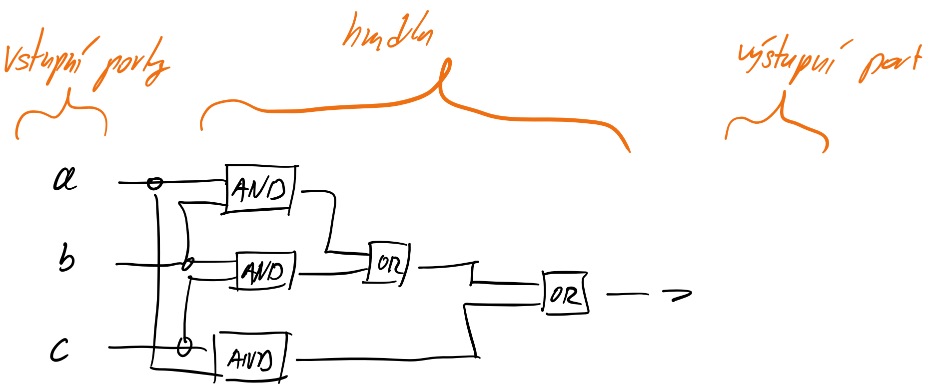
konstanty

	# hradel	
Nulární hradlo:	2	(0, 1)
Unární hradlo:	2	(id, ¬id)

Binární hradla: AND, OR...

Hradlová síť:

MAJORITA



vrcholy
- vstupy
- výstupy
- hradla → funkce, arita

hrany:
- vstup: $\deg^{in} = 0$
- výstup: $\deg^{in} = 1$
- hradlo: $\deg^{in} = \text{arita}$, $\deg^{out} > 0$

- každé hradlo má očíslované vstupy,
- graf musí být acyklický
- každé hradlo musí mít funkci

graf je DAG.

Výpočet: – běží po taktech

$t = 0$: výstup mají vstupní porty a unární hradla

$t > 0$: výstup vydají hradla, co mají všechny vstupy def.

rozdělení na vrstvy →

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

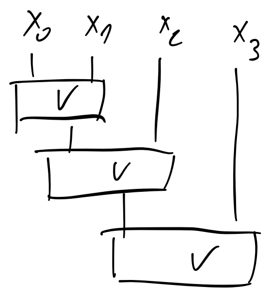
Arita hradel musí být omezena.

Existence 1 v n-bitovém vstupu.

Výstup: $x_0 \vee x_1 \ \dots \ x_{n-1}$

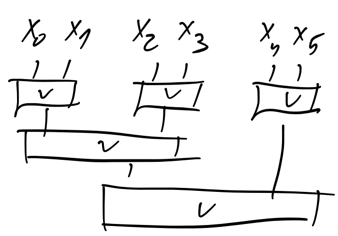
Obecně chceme časy $O(\log^k n)$

Triviální:



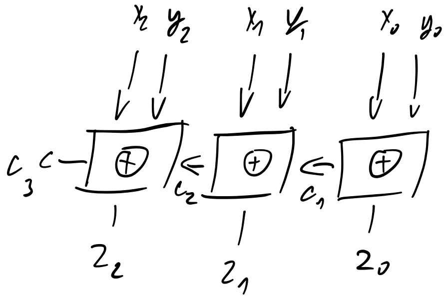
h: $O(n)$
v: $O(n)$

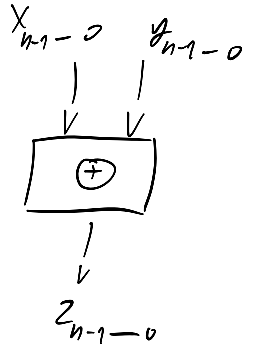
Párování:



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

Binární sčítačka:





$c_0 = 0$

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

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

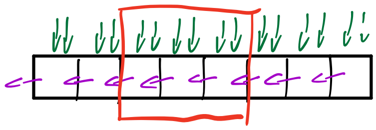
$h: \Theta(n)$

$v: \Theta(n)$

To je ale sekvenční

Pokud bych věděl přenos dopředu, může být výpočet v konstantním čase.

– Můžu uvažovat obě možnosti pro carry bit zespodu.



blok

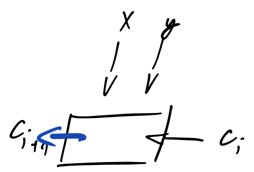
chování bloku: $f: c_{in} \to c_{out}$

- identita $= <$
- negace $=$ nikdy nenastane
- konst 0 $= 0$
- konst 1 $= 1$

↳ jednobitový blok neneguje, skládání takových bloků nevytvoří negaci

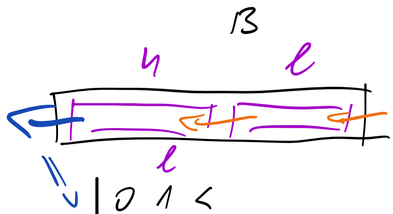
1-bit blok

$y \backslash x$	0	1
0	0	<
1	<	1



$p = x \oplus y$

$q = x$



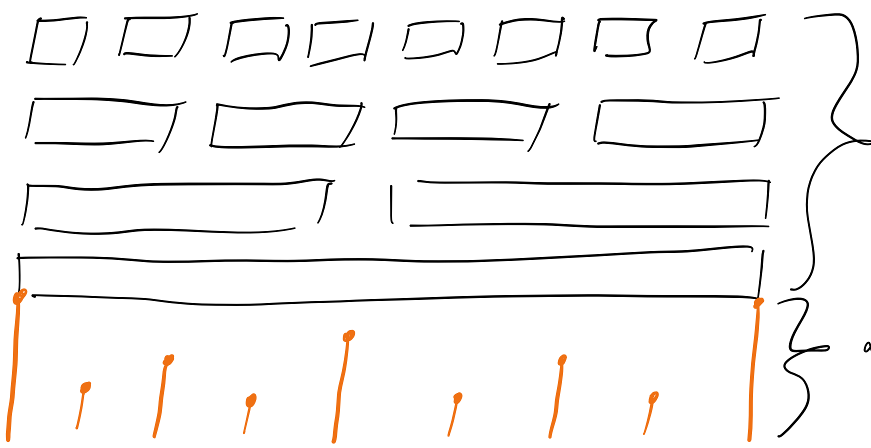
$h \backslash l$	0	1	<
0	0	0	0
1	1	1	1
<	0	1	<

	(p, q)
<	$(1, *)$
0	$(0, 0)$
1	$(0, 1)$

$P_B = p_h \wedge p_l$

$q_B = (p_h \wedge q_l) \vee (\neg p_h \wedge q_h)$

Jak udělat binární sčítačku? Pomocí kanonických bloků:



chování kanonických bloků

$h: \Theta(\log n)$

$v: \Theta(n)$

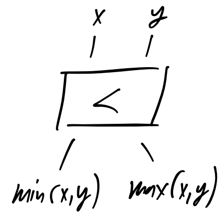
zahušťování bloků

$h: \Theta(\log n)$

$v: \Theta(n)$

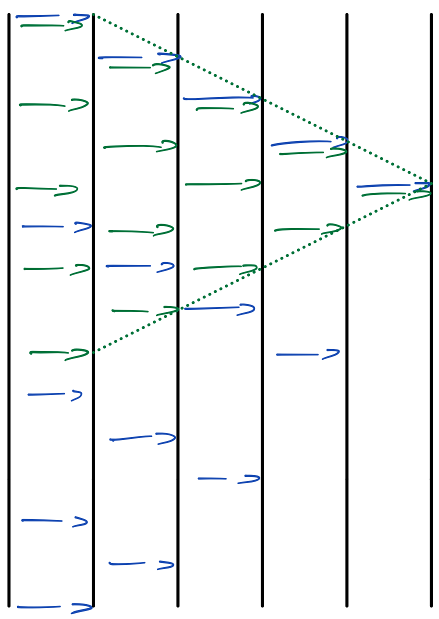
Třídicí síť:

Komparátor:



Smluva: Síť se nevětví.

Bubble Sort:



Tohle ale není paralelní způsob.

Tohle je paralelní způsob.

Hloubka $\Theta(n)$ --> lineární paralelní čas

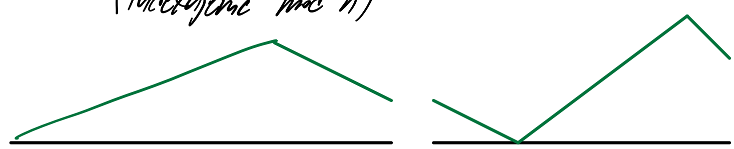
Velikost $\Theta(n^2)$ lze udělat $\Theta(\log^2 n)$

MergeSort:

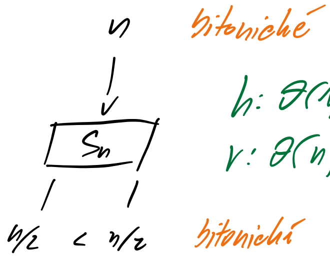
Df: Posl. $x_0 \dots x_{n-1}$ je bitonická $\equiv$

$\exists i,j: x_i < x_{i+1} < \dots < x_{i+j} > x_{j+1} > \dots > x_{i+n-1}$

(indexujeme mod n)



Separátor:



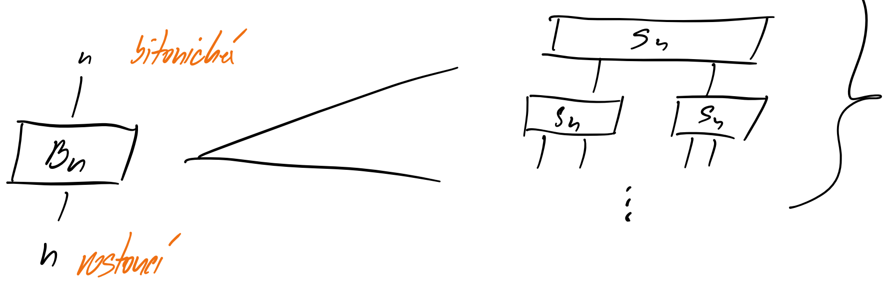
h: $\Theta(1)$

v: $\Theta(n)$

vrací zpět dvě posloupnosti, které jsou bitonické, ale hlavně vůči sobě seřazené.

Pokud budu opakovaně za sebou větvit, dostanu seřazené jednoprvky --> princip bitonické třídičky.

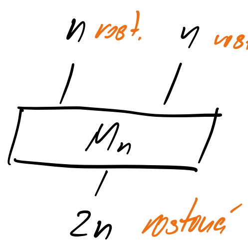
(Bitonická) třídička:



h: $\Theta(\log n)$

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

Slévačka:



--> vezmu jednu, spojím druhou otočenou a pořeším bitonickou třídičkou.

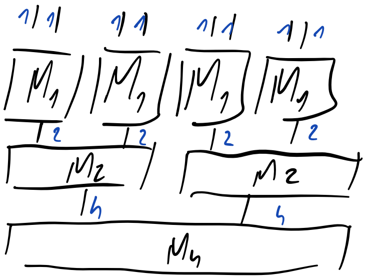
h: $\Theta(\log n)$

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

MergeSort:

n

log 1



log 2

log 4

⋮

$\Theta(\log^2 n)$

Velikost vstupu/výstupu

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

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