2/2/2021

Sorting networks

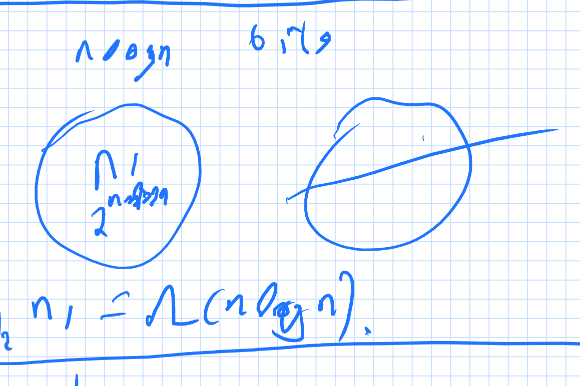
SORTING
n numbers $x_1, \dots, x_n$
$O(n \log n)$ time.
- Merge sort
- Quick sort ←
- Heap/heap sort / binary search tree

$O(n \log n)$ time

$\Omega(n \log n)$

$x_1, x_2, \dots, x_n$
sorting the numbers
computing a permutation
$1, 2, \dots, n$
$\pi(1), \pi(2), \dots, \pi(n)$
$n! \approx 2^{n \log n}$
$2^{c n \log n} \leq n! \leq 2^{c' n \log n}$
$2^{\frac{n}{2}\log\frac{n}{2}} \leq \left(\frac{n}{2}\right)^{n/2} \leq n! \leq n^n = 2^{n \log n}$

$n \log n$ bits



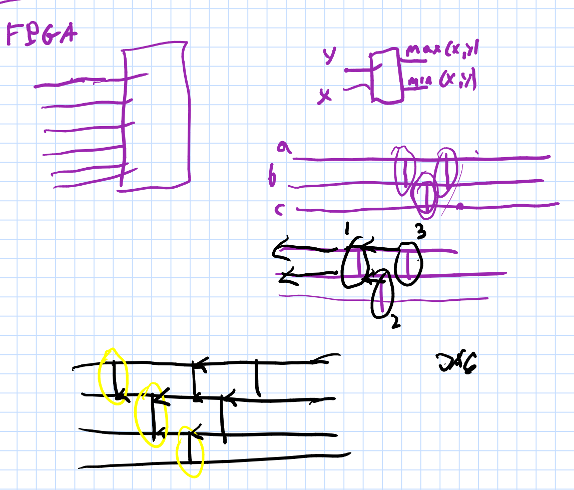
$\geq \log_2 n! = \Omega(n \log n)$.

Radix sort
$x_i = O(n^3)$

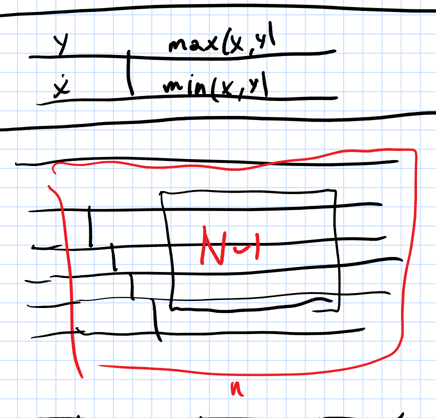
Spaghetti sort
$O(n)$

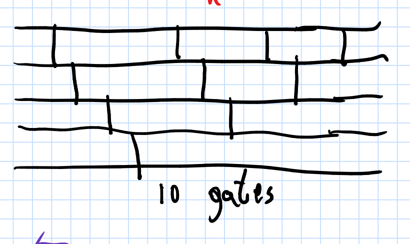
Circuit sorting (oblivious sorting)

FPGA

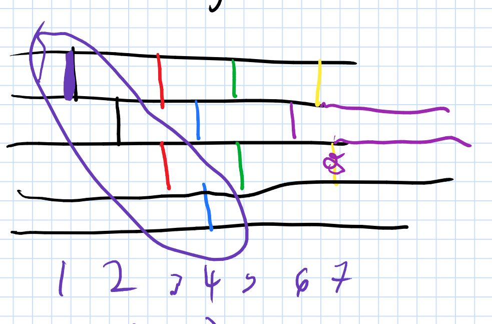


y max(x,y)
x min(x,y)





10 gates



1 2 3 4 5 6 7

$2(n-1)-1 \approx 2n$

$O(n)$

gates
$n-1 + (n-2) + n-3 \dots$
$= \Theta(n^2)$
$\boxed{\Omega(n \log n)}$

$O(\log^2 n)$ time (depth of the circuit)
$O(n \log^2 n)$ Bitonic sort

AKS sorting network
$O(\log n)$ time
$O(n \log n)$ # of gates
Very complicated

$n \geq 10^{70}$ ←

NPC 3SAT
n $O(n)$
$2^{2^{\cdot^{\cdot^{2}}}}$ 64K

0-1 principle

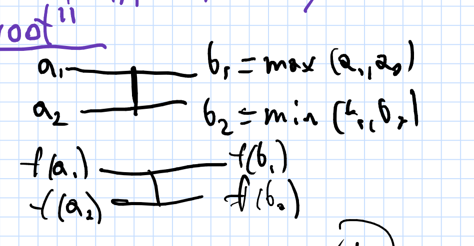
Sorting network that sorts correctly all binary inputs sorts correctly all inputs.

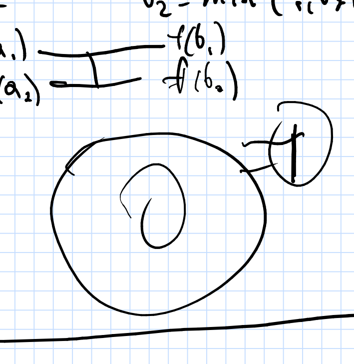
Claim

Let $a_1, \dots, a_n$ input sorting network
Let $b_1, \dots, b_n$ be the output.
f a monotone increasing function
The sorting network on the input
$f(a_1), f(a_2), \dots, f(a_n)$
then the output is
$f(b_1), f(b_2), \dots, f(b_n)$

"Proof"

a_1 — $b_1 = \max(a_1, a_2)$
a_2 — $b_2 = \min(a_1, a_2)$
$f(a_1)$ — $f(b_1)$
$f(a_2)$ — $f(b_2)$



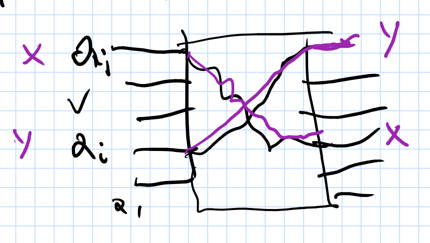


Claim

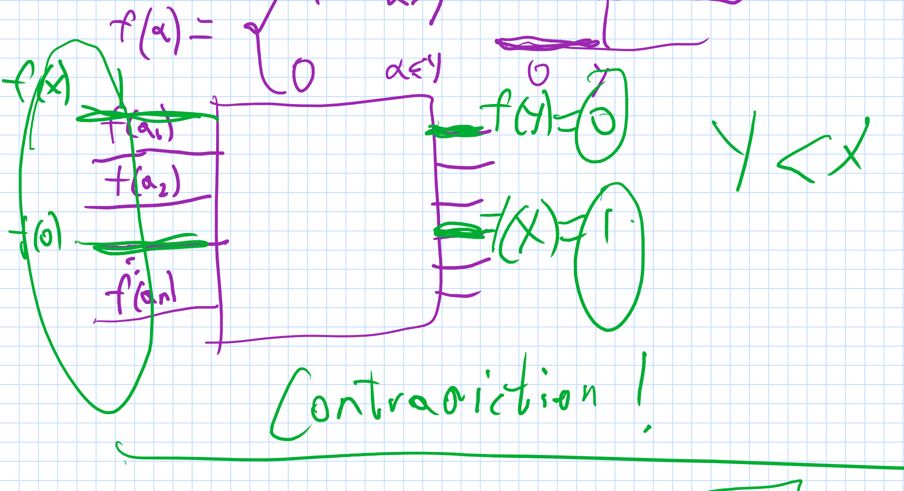
If a sorting network sort correctly all binary inputs then it sorts correctly all inputs.

Proof

Assume false.

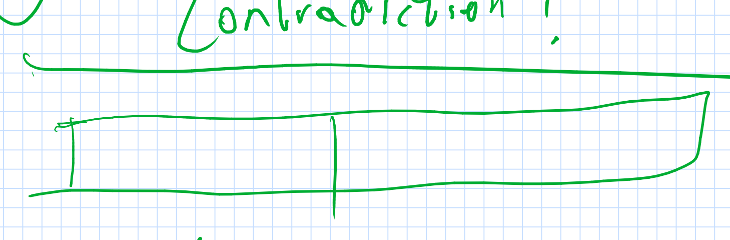


$x > y$
$f(\alpha) = \begin{cases} 1 & \alpha > y \\ 0 & \alpha \leq y \end{cases}$



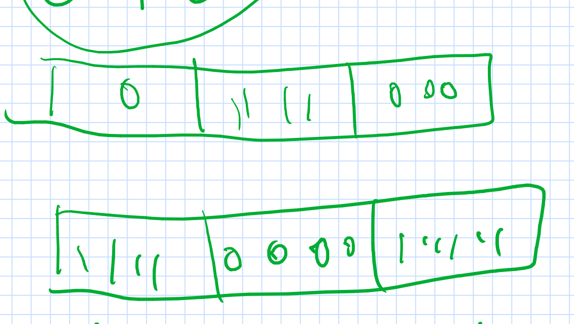
$y < x$

Contradiction!



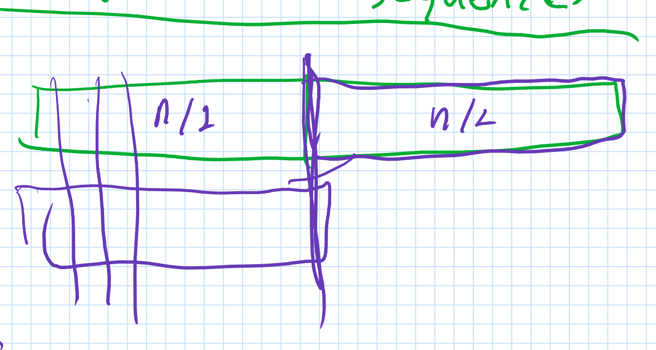
Bitonic sequences

$0^i 1^j 0^l$ shift

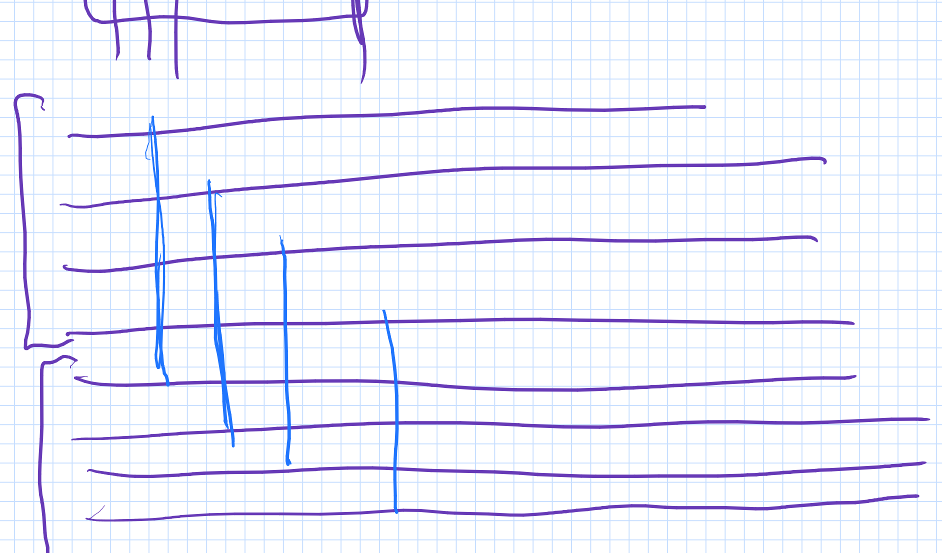


$1^i 0^j 1^k$ bitonic sequence

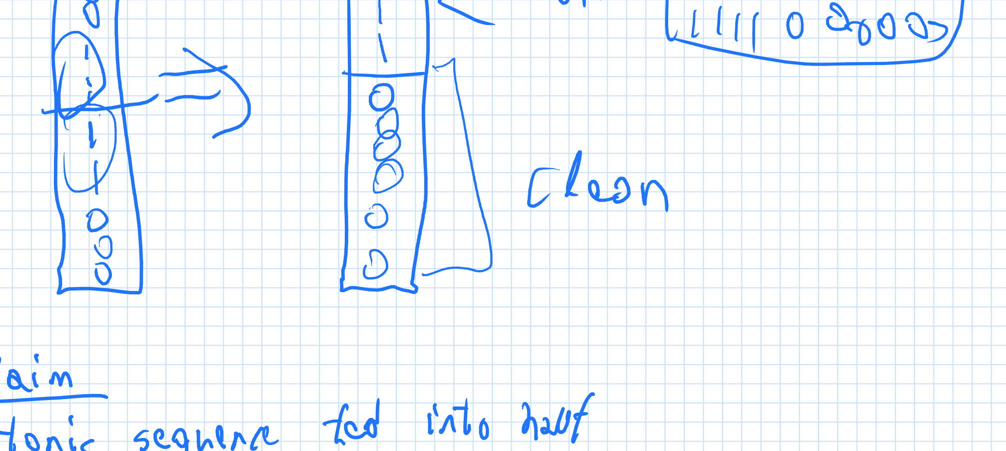
Sorting bitonic sequences



n



$\frac{n}{2}$ half cleaner[n]

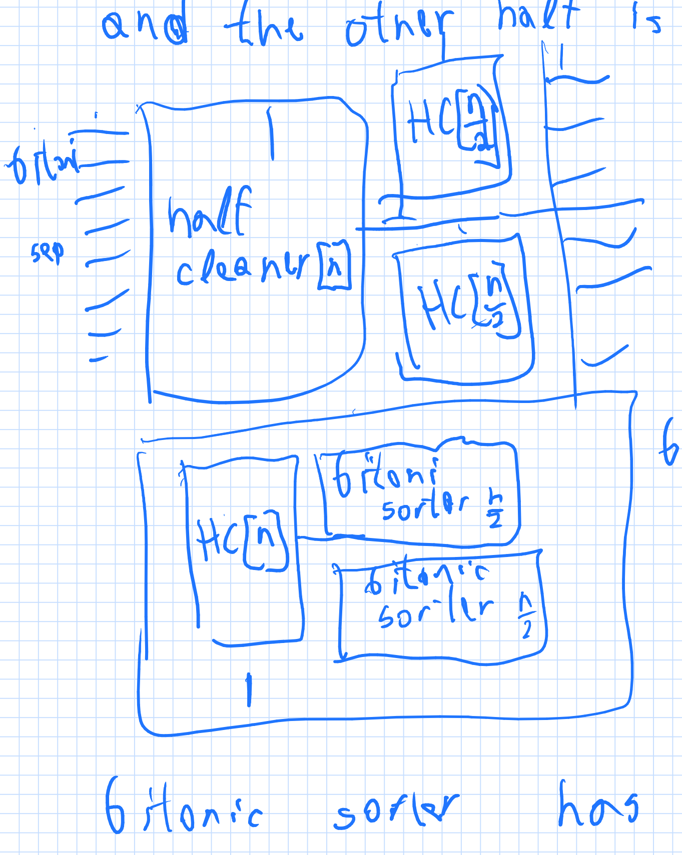


bitonic

clean

Claim

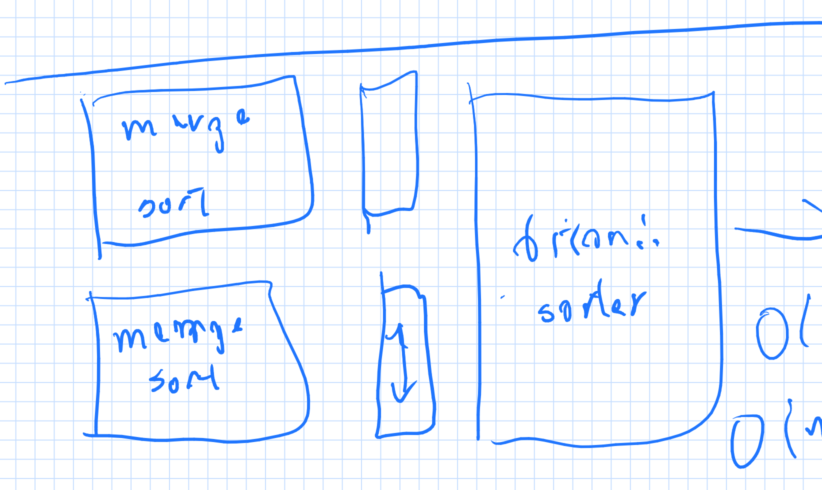
bitonic sequence fed into half cleaner result in one half being clean and the other half is bitonic.



$O(\log n)$ layers

bitonic sorter[n]

bitonic sorter has $O(\log n)$ depth
$O(n \log n)$ gates



$O(\log^2 n)$
$O(n \log^2 n)$