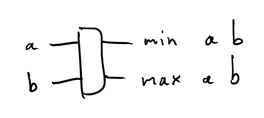
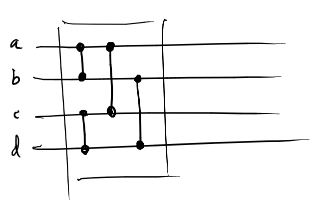
Sieci sortujące

Komparator

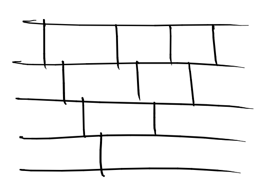




Zadanie

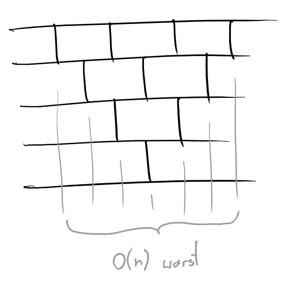
Dla danego n skonstruować sieć sortującą n wejść

Select sort:



Parametry: głębokość sieci, #przełączników

Ten sam select sort:



Minimalna gł.: $O(\log n)$

[tak jak przy przełącznikach - $n!$ permutacji $\mapsto O(\log(n!))$]

Lemat

Jeśli sieć komparatorów oblicza $\{a_i\}_n \mapsto \{b_i\}_n$,

to $\forall f$ - niemalejąca .

$\{f\ a_i\}_n \mapsto \{f\ b_i\}_n$

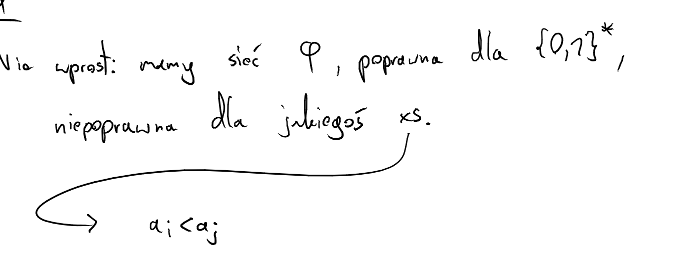
φ - sieć komparatorów $[a]_n \to [a]_n$

f - monotoniczna

$\Longrightarrow \forall xs.\ \varphi(\text{map } f\ xs) = \text{map } f\ (\varphi\ xs)$

D-d

Indukcja (po strukturze sieci ?)
głębokości?

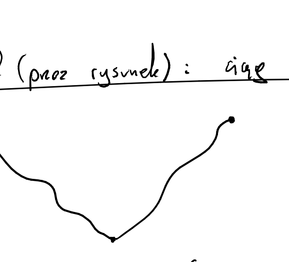


Wniosek (zasada 0-1)

Jeśli sieć poprawnie sortuje $\{0,1\}^*$, to wszystko sortuje poprawnie.

D-d

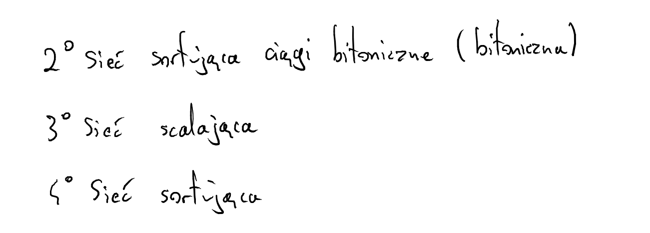
Nie wprost: mamy sieć φ, poprawną dla $\{0,1\}^*$, niepoprawną dla jakiegoś xs.



$$f(x) = \begin{cases} 0 & x \le a_i \\ 1 & \text{w.p.p.} \end{cases}$$

Konstrukcja sieci bitonicznej (Batcher)

Def (przez rysunek): ciąg bitoniczny



$a_1 \ge a_2 \ge \ldots \ge a_i \le a_{i+1} \le \ldots a_n$ $\qquad$ $a_1 \le a_2 \le \ldots \le a_i \ge a_{i+1} \ge \ldots a_n$

(jedno ekstremum)

Kroki konstrukcji:

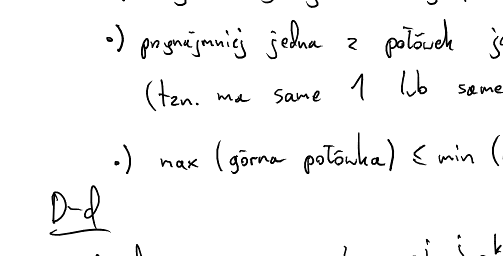
1° Sieć półczyszcząca

2° Sieć sortująca ciągi bitoniczne (bitoniczna)

3° Sieć scalająca

4° Sieć sortująca

Ad 1°: półczyszcząca



gł.: 1 #: n

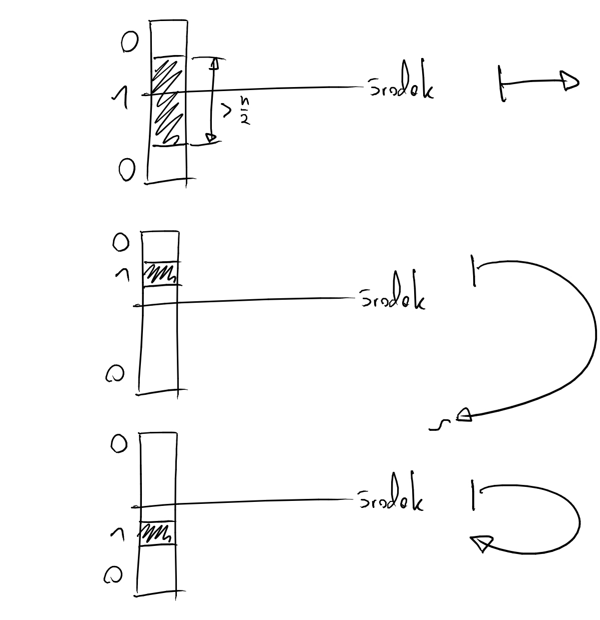
Lemat:

Jeśli na wejściu sieci półczyszczącej jest ciąg bitoniczny $\{0,1\}^*$, to ciąg na wyjściu ma własności:

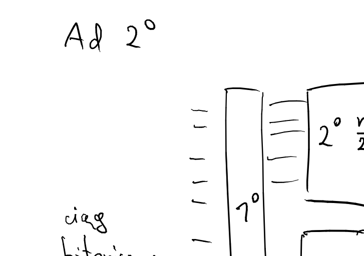
- ciągi na górnej i dolnej połowie są bitoniczne
- przynajmniej jedna z połówek jest czysta (tzn. ma same 1 lub same 0)
- max (górna połówka) ≤ min (dolna połówka)

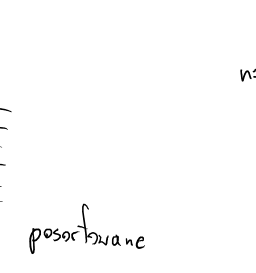
D-d

Niech ciąg ma postać $0^i 1^j 0^k$.
(analogicznie $1^i 0^j 1^k$)



Ad 2°

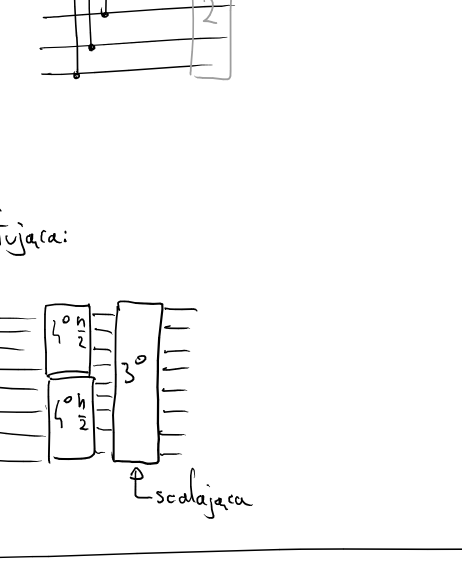




Ad 3°
(Lemat: zasada 0-1 dla sieci scalających)

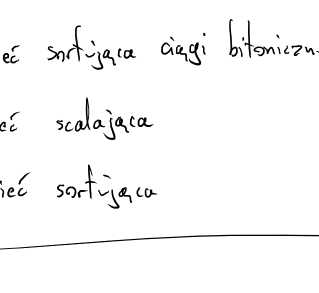
Sieć scalająca

$0^i 1^j 0^k 1^l \mapsto 0^i 1^{j+l} 0^k$ - ciąg bitoniczny!



Ad 4°

Sieć sortująca:



	głębokość:
1° Sieć półczyszcząca	$O(1)$
2° Sieć sortująca ciągi bitoniczne (bitoniczna)	$O(\log n)$
3° Sieć scalająca	$O(\log n)$
4° Sieć sortująca	$O(\log^2 n)$

oblivious - algorytm wykonuje porównania zawsze na tych samych elementach, niezależnie od wykonania

(Ajtai, Komlos, Szemeredi)

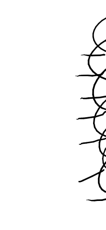
Konstrukcja sieci $O(\log n)$ (sieć AKS)
$\uparrow$ 6000

$6000 \log n < \frac{1}{2} \log^2 n$

$12000 < \log n$

Inne kryteria: rozmiar obszaru, na którym można upakować

Inne sieci: ~cykliczne



głębokość: $O(\log\log n)$

$3 \in O(1)$

czas działania: $O(\log^2 n)$