アルゴリズム特論 I
ソーティングネットワーク
ーバイトニックソーティングー

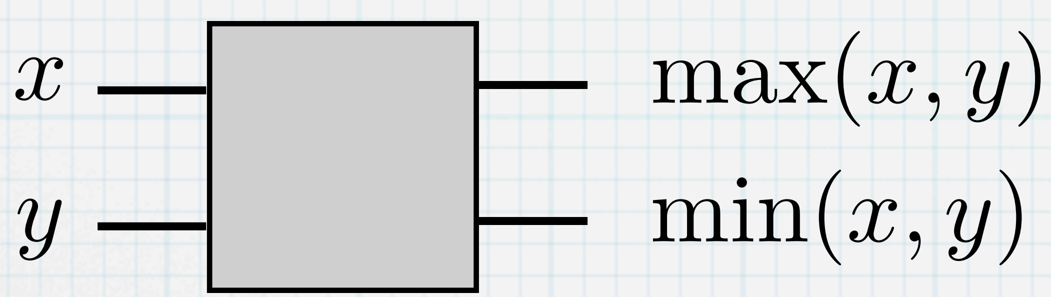
山本修身

ソーティングの概略

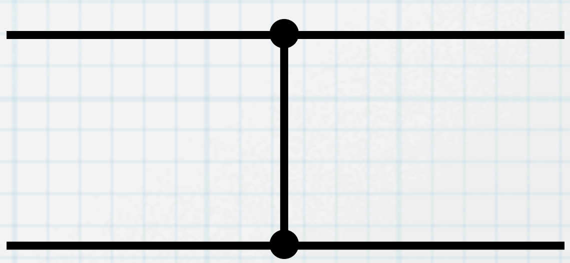
* O(n^2), バブルソート，挿入ソート（O(n)）
* O(n log n), マージソート，（クイックソート），ヒープソート
* 並列の世界では？ nプロセッサでソートすると O(log n)?

ソーティングネットワークとは

* ソーティングネットワークは下のような比較器 (comparator) を組み合わせて構成されるネットワークで，任意の入力に対して，ソーティングされた列を出力するものである．



comparator

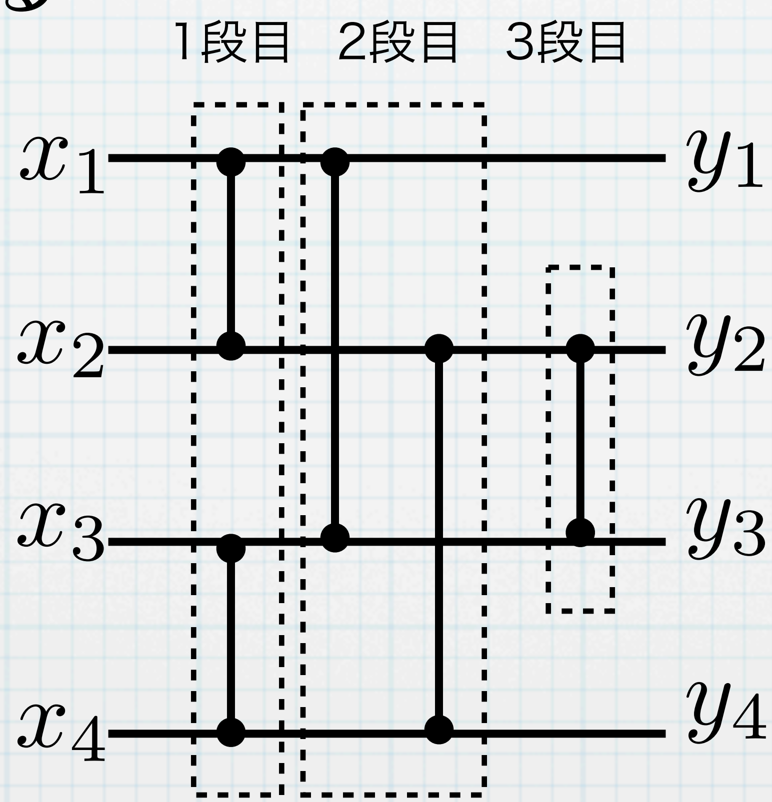


記法

※ 大きな値が上に小さな値が下に出力される.

ソーティングネットワークの例

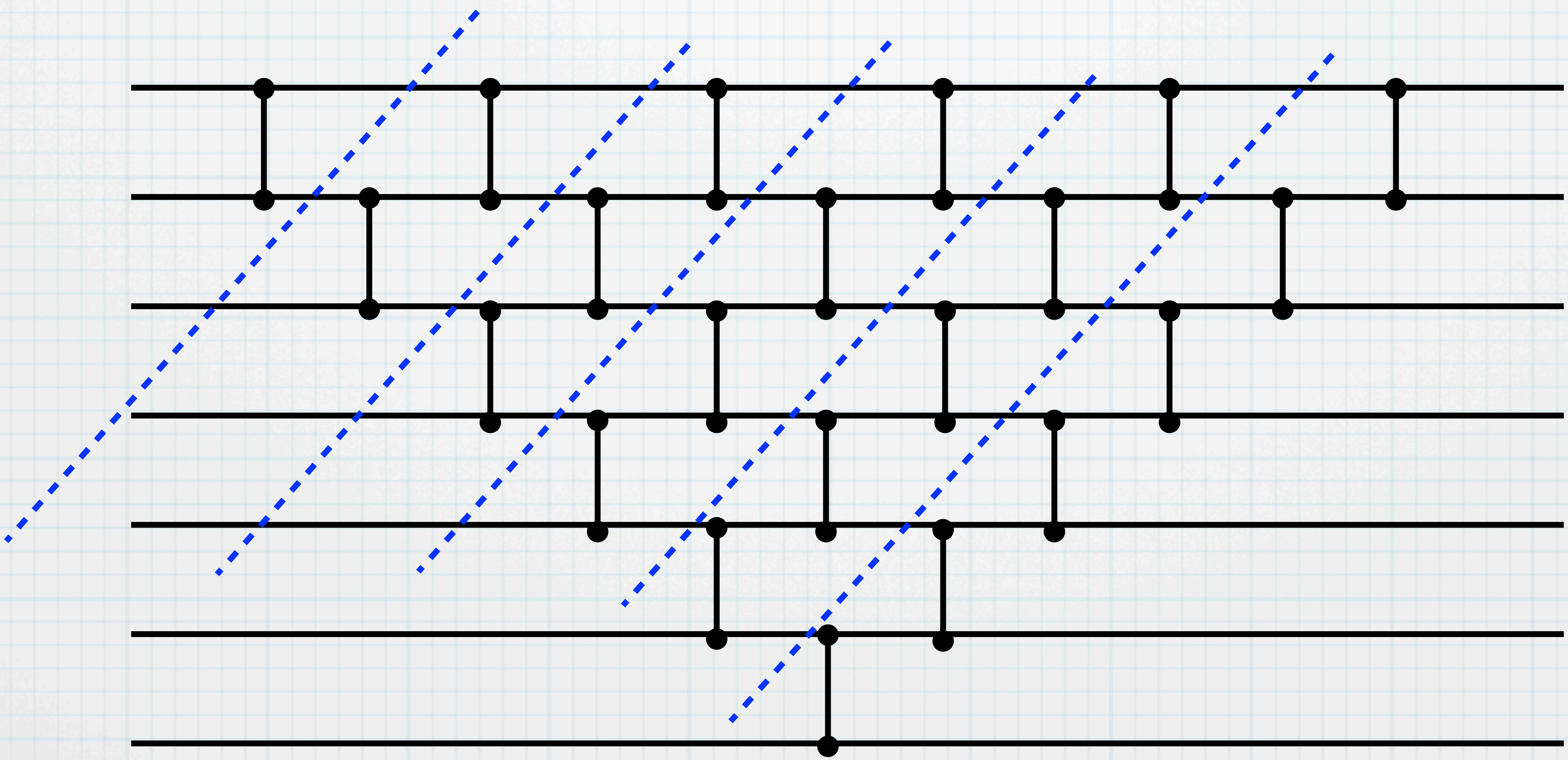
* ソーティングネットワークでは，直接に関係のない比較は並行して処理することができる．従って，ソーティングにかかる時間は並列化できないものが何段重なっているかということになる



4つの要素をソートするソーティングネットワーク

ソーティングネットワークによるインサーションソート

* インサーションソートに対応するソーティングネットワークは簡単に作ることができる.



段数は $O(n)$

補助定理１：単調性

* 比較器のネットワークの入力と出力の対応が

$$\langle a_1, a_2, \ldots, a_n \rangle \to \langle b_1, b_2, \ldots, b_n \rangle$$

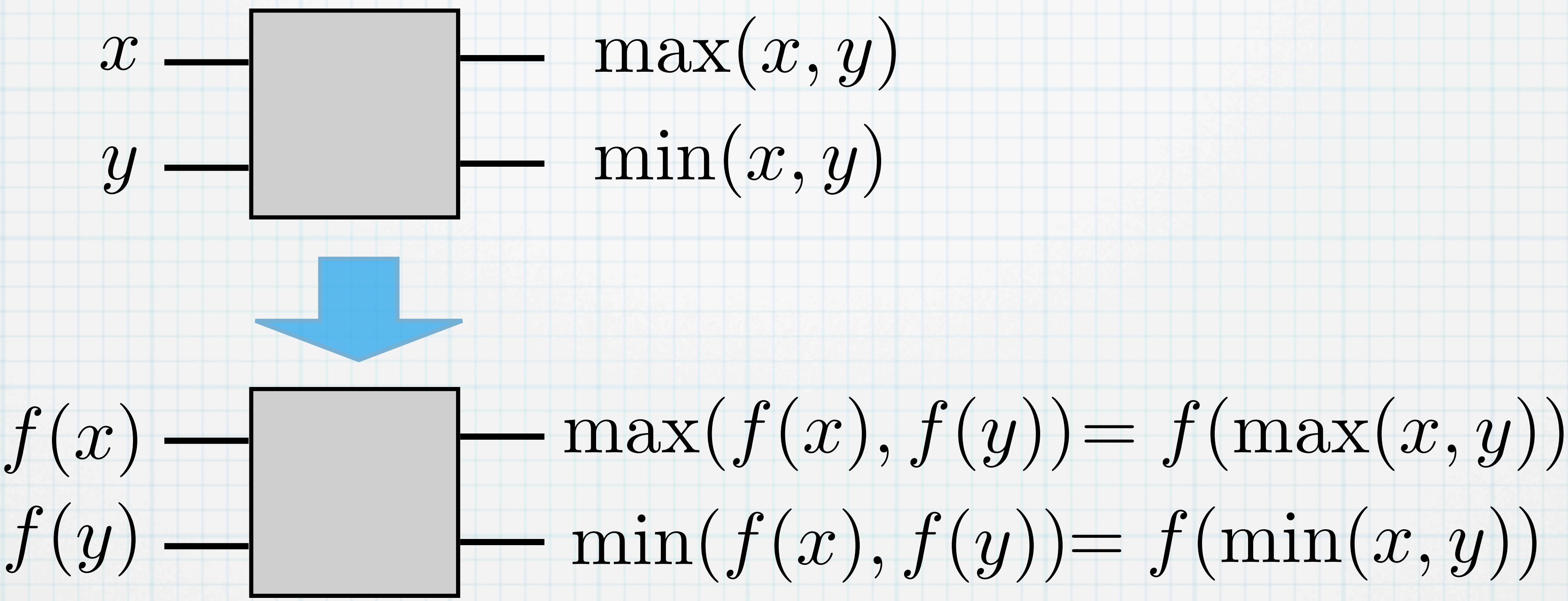
となっているとする．また，任意の単調増加関数を f とおけば，この関数を適用した入力は同じネットワークによって，

$$\langle f(a_1), f(a_2), \ldots, f(a_n) \rangle \to \langle f(b_1), f(b_2), \ldots, f(b_n) \rangle$$

という入出力の対応となる．

補助定理１の証明

* 1個の比較器について考える．



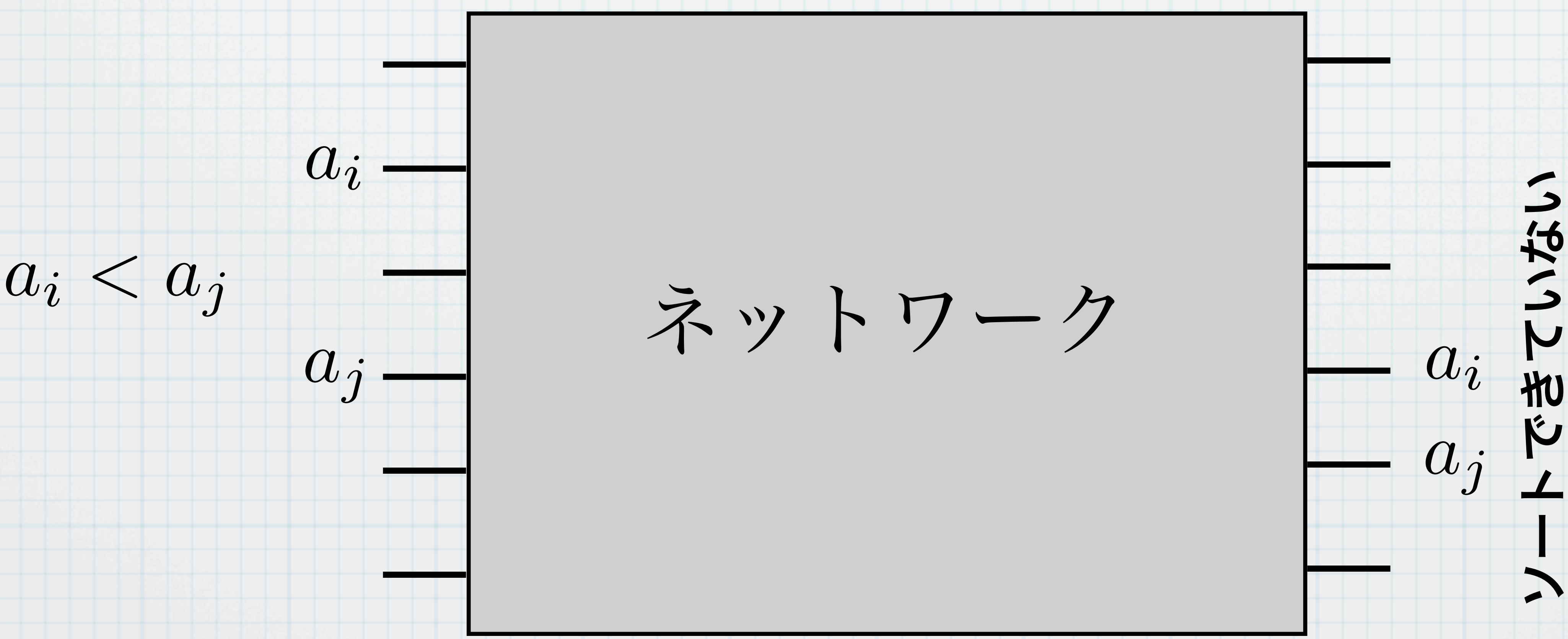
それぞれの比較器において，入力に f を適用すれば，出力に f が出てくることになるので，そのことにより，このネットワークの出力は f を適用したものとなる．

0-1原理

* 任意のネットワークが与えられたとき，このネットワークにあらゆる0-1のパターンの入力に対する出力はいずれもこのパターンをソートしたものになっているとき，このネットワークはソーティングネットワークである．

0-1原理：証明（1）

* 与えられた比較器によるネットワークについて，ある実数値の並びが与えられたとき，ソートしないとする．

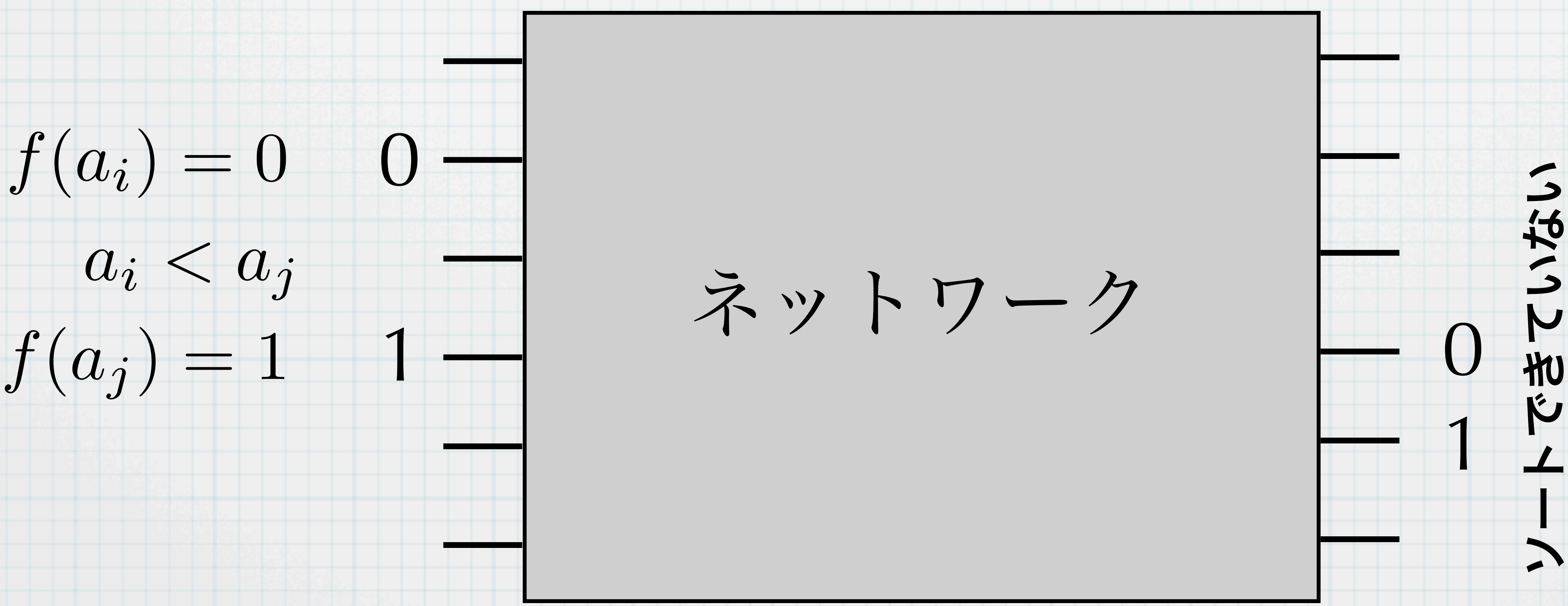


0-1原理：証明（2）

* 入力に以下の単調増加関数 f を適用すると，補助定理 1 より，以下のようになる．

$$f(x) = \begin{cases} 1 & (x \geq a_j) \\ 0 & (x < a_j) \end{cases}$$

これはある0-1のパターンについてソートできないことになる．対偶により，0-1原理が成立．

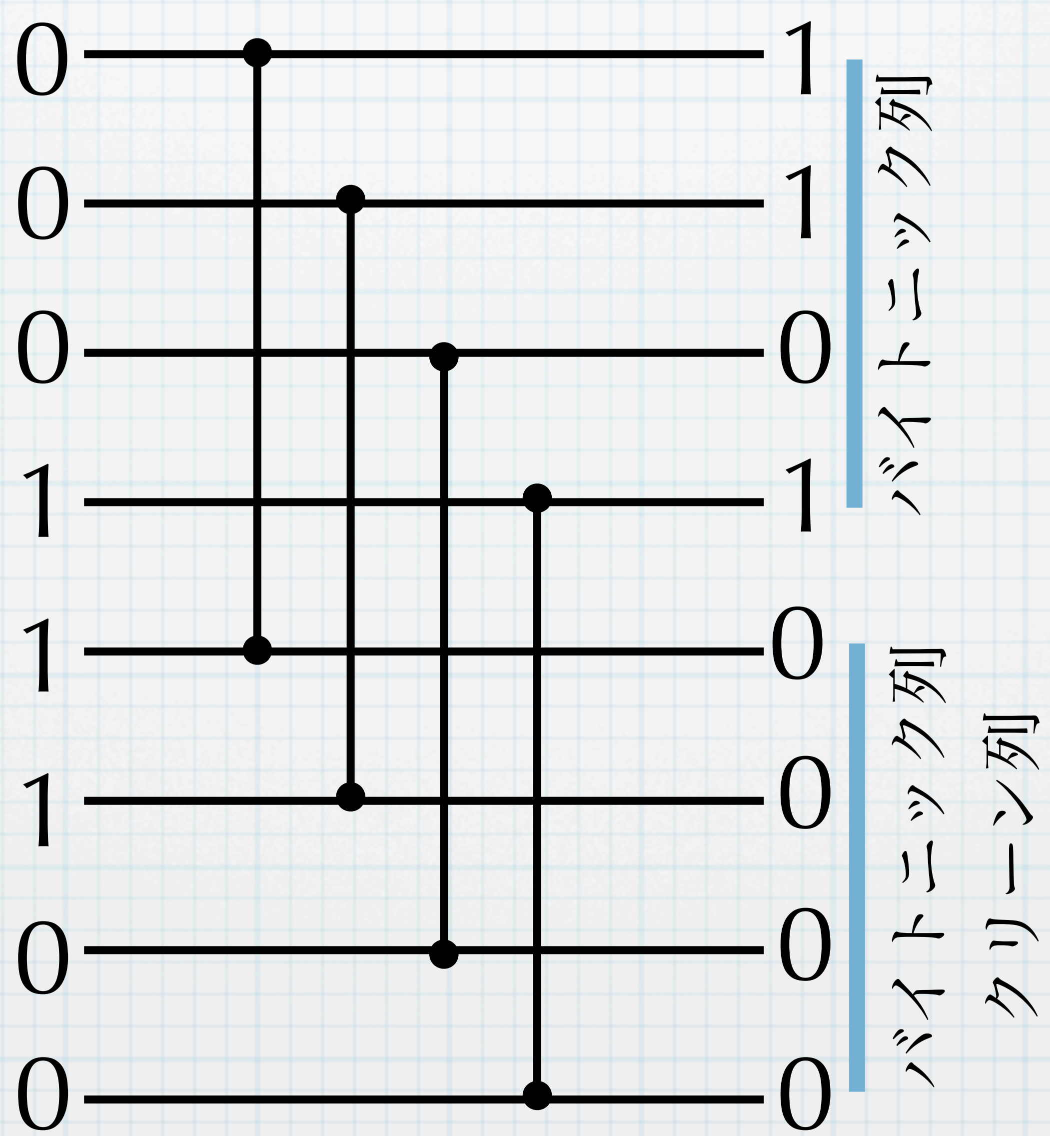


バイトニックソーター

* すべてのバイトニック0-1列をソートするソーティングネットワークのことをバイトニックソータと呼ぶ.
* バイトニック列とは, 「上がって下がる」または「下がって上がる」0-1列のことである.
例：00000111100000, 11100000001111

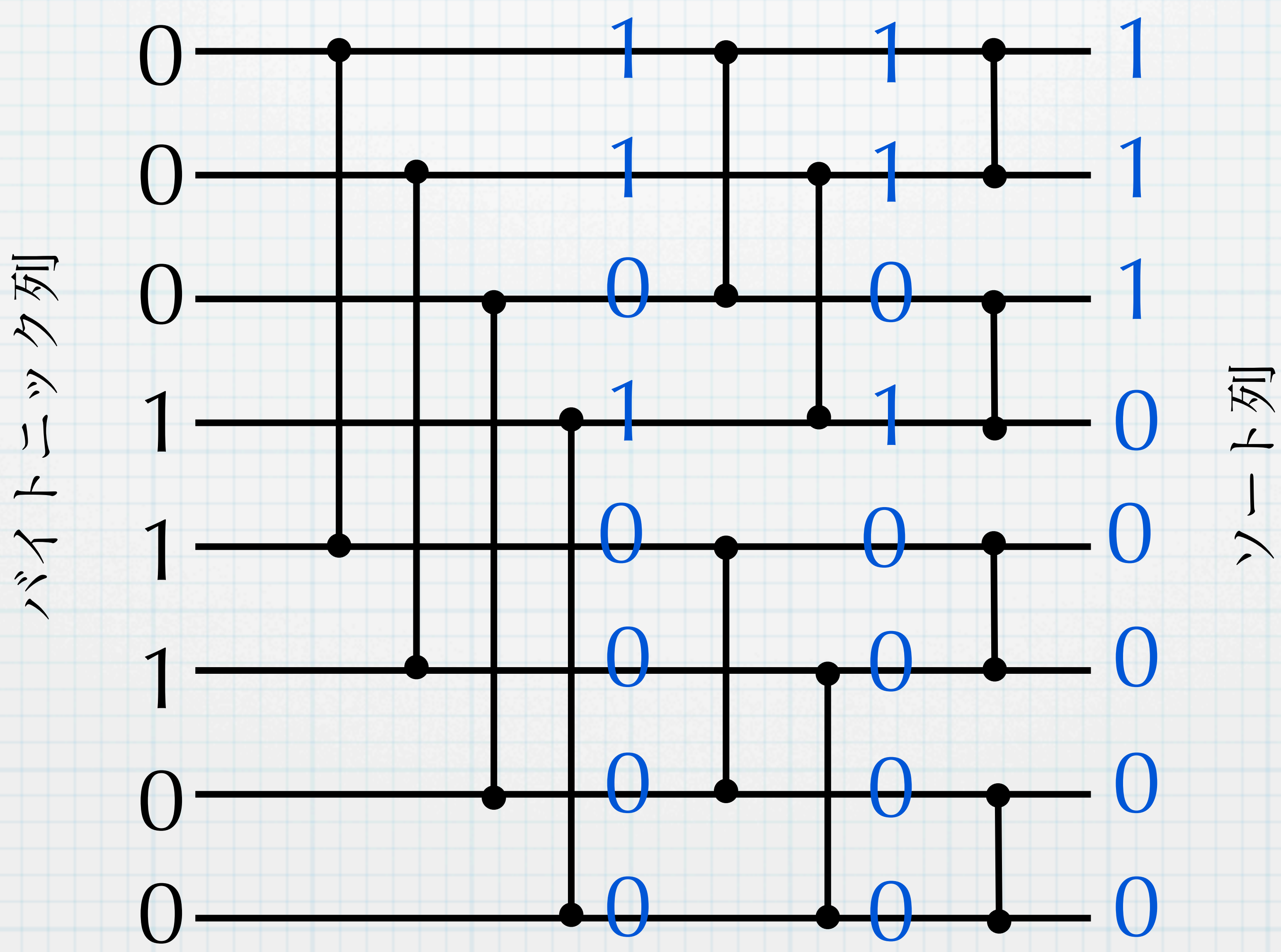
ハーフクリーナ

* バイトニックソータを実現するために用いられるのがハーフクリーナである.



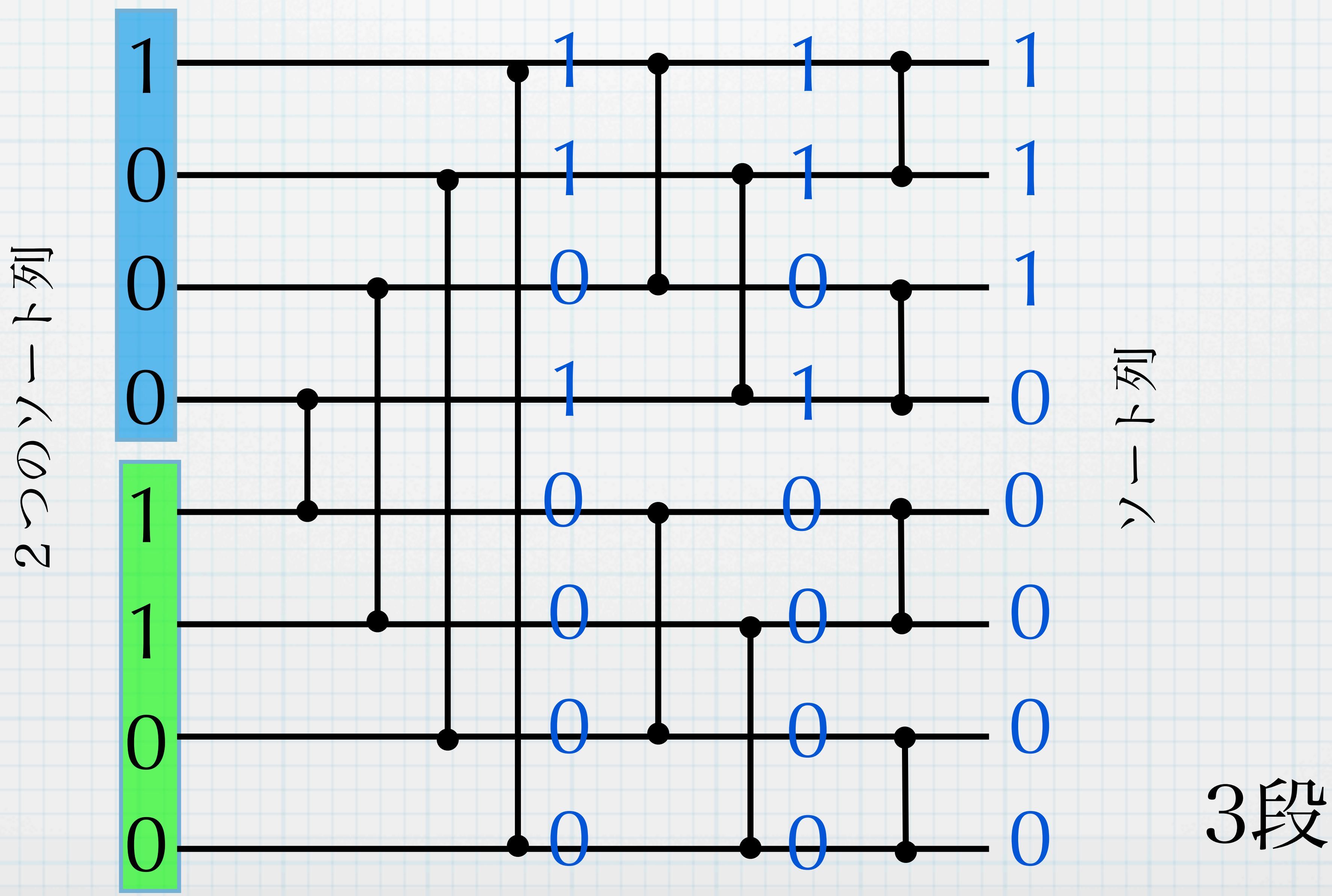
ハーフクリーナの重ね

* ハーフクリーナを重ねていくことにより，バイトニックソータが実現できる．



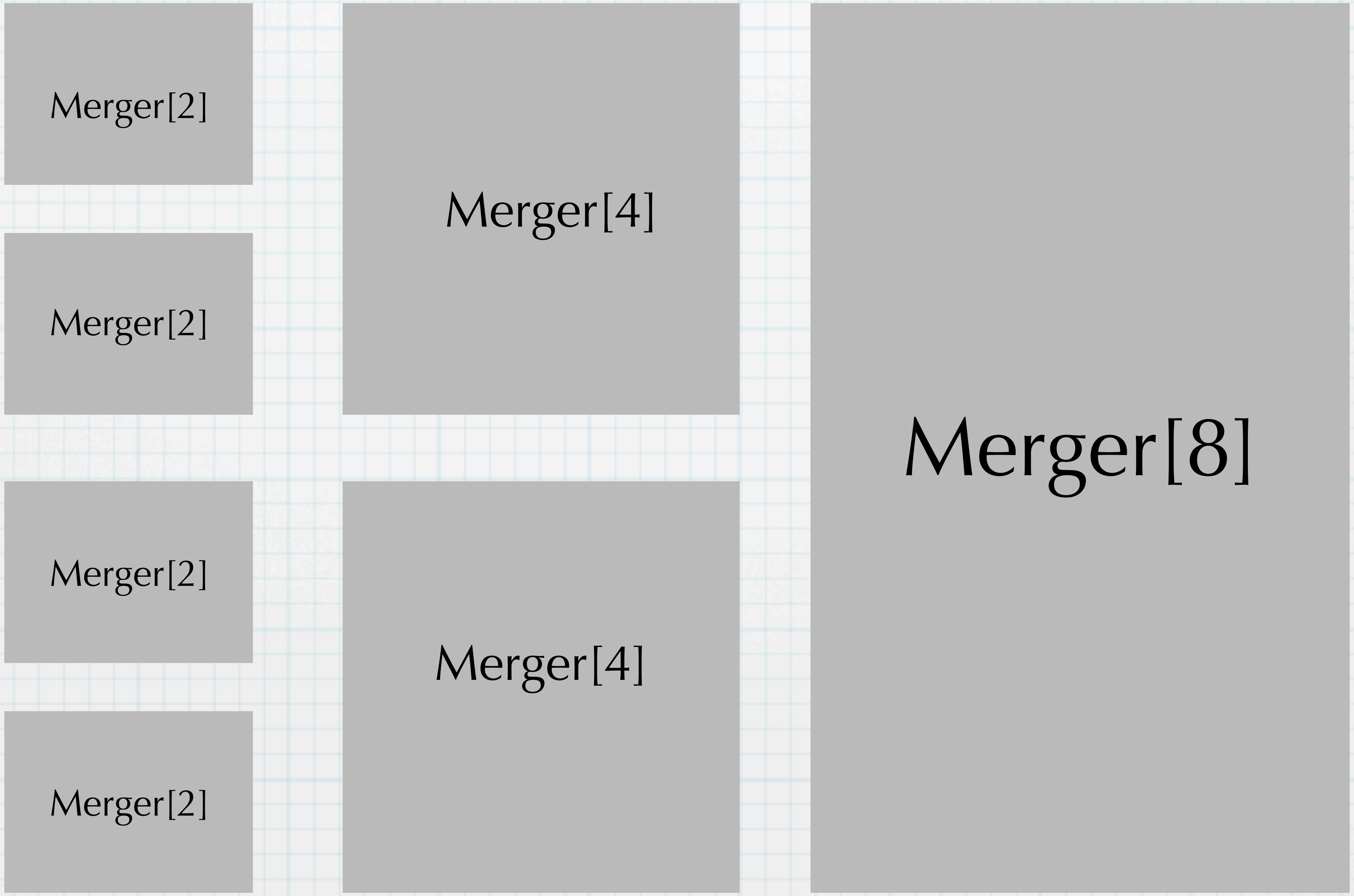
バイトニックソータからマージャへ

* ソートされた２つの列の方向を逆転させてマージするとバイトニック列ができる．
* 従ってバイトニックソータの最初のハーフクリーナを変形すれば，マージャが構成できる．



マージャを組み合わせればソータが得られる

* マージャを組み合わせればすべての0-1列をソートするネットワークが得られる．基本構造はマージソートと同じ．



マージャの段数

* Merge[2^n]の段数はnである．

* また，2^n個の要素をソートするのに必要なマージャはMerger[2^n], Merger[2^{n-1}],...., Merger[2] である．従って，必要な段数は,

$$n + (n-1) + \cdots + 1 = \frac{n(n+1)}{2}$$

であり，入力の要素数をNとおけば，段数は,

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

である．