Elektroniczny Biuletyn Informatyki Teoretycznej. 3-97

Correction Network

Marcin Kik [1]

Institute of Computer Science, University of Wrocław, ul. Przesmyckiego 20, PL-51-151 Wrocław, Poland, email: kik@tcs.uni.wroc.pl

Abstract

Suppose that there is a sorted sequence and at most k elements change their values, for some $k \ll n$. We present a comparator network of depth $\leq 4 \log n + O((\log k \log\log n)^2)$ that makes the sequence sorted again.

1 Introduction

The main application of comparators [3] is the construction of sorting networks. However they may be used in construction of the networks performing other tasks. Examples are selection [6] and merging [2] networks. Merging networks may be considered as the sorting networks for the proper subclass of inputs, i.e. the sequences with two sorted subsequences on some fixed positions. In many other practical applications we need to sort the sequences that have some properties that can be used by the sorting algorithm. Consider the situation where we have a sorted sequence of length n and some small number of its elements changes the value. We have to sort the sequence again. We could use any sorting network to do it. The asymptotically quickest known sorting network (the AKS network [1]) has the depth $c \log n$ where the constant c is very large. On the other hand, the Batcher sorting network [2], which seems to be the best one for practical applications has the depth $\Omega(\log^2 n)$. We present a network that exploits the property that the given input is a slightly damaged sorted sequence. We assume that the number of of elements that have been changed is not greater than some constant k. A correction network C of depth $2 \log n - 1$ by Schimmler and Starke [5] may be used for $k = 1$. For arbitrary k, the composition of k such networks C^k has to be used. A Schear-sort network [4], which is very suitable for VLSI technology, may be modified to solve this problem in time $O(\sqrt{n} + k \log k)$. The main result presented in this article is:

Theorem 1 *Let n and k be arbitrary positive integers such that $k < (n/6)^{1/(\log\log n+2)}$. Then there is a comparator network $N_{n,k}$ of depth $4 \log n + O((\log k \log\log n)^2)$ that sorts any input sequence of length n that is different from some sorted sequence at at most k positions.*

[1] Supported by KBN grant 8 S503 023 07

12 September 1997

If k is $o\left(2^{\sqrt{\log n}/\log\log n}\right)$, then the depth of $N_{n,k}$ is $4\log n + o(\log n)$.

2 Preliminaries

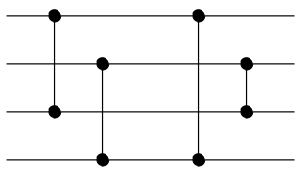
For $n \geq 1$ let $V_n = \{R_1, \ldots, R_n\}$ be a set of of n distinct elements. A *comparator network* N is a triple (n, d, L), where n and d are positive integers and $L = (L_1, L_2, \ldots, L_d)$ is a sequence of sets (called *layers)* of ordered pairs (called *comparators*) such that for each i, $1 \leq i \leq d$, $(V_n\,, L_i)$ is an directed graph with no loops and with each vertex incident to only one arc. The elements of V_n are called *registers*, n is called *input size* and d is called *depth* of N. An *input* for a network N is a sequence of length n. We say that the i-th element of an input is *stored* in register R_i. The sequence stored in the registers is transformed by the layers of the network according to following rules.

Let $s = (s_1, s_2, \ldots, s_n)$ be a sequence of n elements. Let L_k be any layer of network N. We denote by $L_k(s)$ the sequence $t = (t_1, t_2, \ldots, t_n)$ such that:

– if L_k has comparator (R_i, R_j), then $t_i = \min\{s_i, s_j\}$ and $t_j = \max\{s_i, s_j\}$;
– if register R_i is not incident to any comparator in L_k, then $t_i = s_i$.

The *output* of N for the input sequence s is the sequence $N(s) = L_d(L_{d-1}(\ldots L_1(s)\ldots))$.

Network N is called *standard* network if each comparator in each layer of N has the second component greater than the first one. Sometimes we use following graphical representation of standard network $N = (n, d, L)$. The registers of N are drawn as horizontal lines (in that case we call them *wires*). The wire R_{i+1} is drawn immediately below the wire R_i. If there is an comparator (R_i, R_j) in some layer of N it is drawn as the vertical line segment connecting the wire R_i with the wire R_j. All comparators of layer L_{k+1} are drawn to the right of all comparators of the layer L_k. For example the network $N = (4, 2, (\{(1,3), (2,4)\}, \{(1,4), (2,3)\}))$ has following graphical representation:



In this paper we consider sequences created by changing some values in some sorted sequence. A *k-damaged sequence* is a sequence that may be obtained from a sorted sequence by changing the values of at most k elements. A *k-damaged 0-1-sequence* is an sequence consisting of zeroes and ones that may be obtained from any sorted sequence consisting of zeroes and ones by changing the values of at most k elements.

The following lemma is an simple modification of the well known zero-one principle from [3]:

Lemma 2 *Comparator network N sorts all k-damaged sequences if and only if N sorts all k-damaged 0-1-sequences.*

Proof. (Sketch) Let $a = (a_1, a_2, \ldots, a_n)$ be a k-damaged input sequence for the network N and let $b = (b_1, b_2, \ldots, b_n)$ be the output $N(a)$ computed by network N for the input a. If b is not sorted, then there is an index i, $1 \le i \le n-1$, such that $b_i > b_{i+1}$. Let function f be defined as follows: $f(x) = 0$ if $x < b_i$ and $f(x) = 1$ if $x \ge b_i$. It is easy to show that for the input sequence $a' = (f(a_1), f(a_2), \ldots, f(a_n))$ the network N outputs the sequence $N(a') = (f(b_1), f(b_2), \ldots, f(b_n))$, which is not sorted, since $f(b_i) = 1$ and $f(b_{i+1}) = 0$. On the other hand a' is an k-damaged 0-1-sequence since it differs at at most k positions from $(f(c_1), \ldots, f(c_n))$, where $c = (c_1 \ldots, c_n)$ is a sorted sequence from which a can be produced by changing at most k elements.

Lemma 3 *Any k-damaged 0-1-sequence s of length n consisting of x zeroes and $n-x$ ones has following properties:*

(i) There are at most k ones on positions 1 to x.
(ii) There are at most k zeroes on positions $x+1$ to n.

Proof. Let t be a sorted zero-one sequence from which s can be produced with at most k changes. Then also t can be produced from s with at most k changes. If there are more than k ones on the positions 1 to x in sequence s, then there must be more than k zeroes on positions $x+1$ to n in s and we can not produce a sorted sequence t from s with only k changes, since more than k zeroes are followed by more than k ones in s. The second property can be shown in the same way.

Let s be a zero-one sequence of length n given on the input of the comparator network. If s contains x zeroes and $n-x$ ones, then we call the first x registers a *region for zeroes*, and the remaining registers a *region for ones*. We call all the ones that are out of the region for ones and all the zeroes that out of the region for zeroes *displaced* elements. Note that Lemma 3 says that any k-damaged 0-1-sequence has at most k displaced ones and at most k displaced zeroes.

Let $a^{(1)}, a^{(2)}, \ldots, a^{(m)}$ be zero-one sequences of length n. We denote by $\bigvee_{i=1}^{m} a^{(i)}$ a sequence $c = (c_1, \ldots, c_n)$ such that for $1 \le j \le n$: $c_j = \max\{a_j^{(i)} \mid 1 \le i \le m\}$. For two zero-one sequences of length n: a and b, we say that $a \le b$ if $a_j \le b_j$ for each j, $1 \le j \le n$.

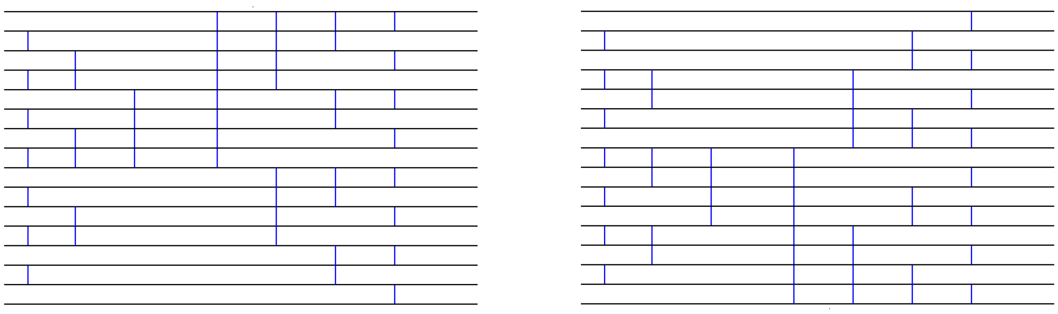
3 Networks I_n^1 and I_n^0

For any $n = 2^m$ there is a network I_n^1 of depth $2m-1$ and with $O(n)$ comparators that sorts any zero-one sequence that may be obtained from some sorted zero-one sequence by increasing one of its elements. The network I_n^1 is defined as follows:

$$I_2^1 = (2, 1, (\{(R_1, R_2)\})$$

Let $n = 2^m$ for some $m \ge 1$, and let $I_n^1 = (n, 2m-1, L)$ where $L = (L_1, L_2, \ldots, L_{2m-1})$. Then $I_{2n}^1 = (2n, 2m+1, L')$ where $L' = (L'_1, L'_2, \ldots, L'_{2m+1})$ and:

(i) $L'_1 = \{(R_{2i}, R_{2i+1}) \mid 1 \le i \le n-1\}$,
(ii) $L'_{2m+1} = \{(R_{2i-1}, R_{2i}) \mid 1 \le i \le n\}$,
(iii) $L'_{t+1} = \{(R_{2i-1}, R_{2j-1}) \mid (i,j) \in L_t\}$, for $1 \le t \le 2m-1$.



I^1_{16} I^0_{16}

Fig. 1. The networks I^1_n and I^0_n for $n = 16$.

Remark 4 *For $2 \leq k \leq 2m$, all comparators in the layers L'_k are incident to odd numbered registers only and the network I^1_{2n} without the first and the last layers and without even registers is identical to I^1_n.*

We will use following property of I^1_n:

Lemma 5 *Let k be arbitrary positive integer not greater than n. Let $a = (a_1, a_2, \ldots, a_n)$ be a zero-one sequence such that only one of the elements a_i, for $1 \leq i \leq k$, is equal to one. Then for input a the network I^1_n outputs the sequence $b = I^1_n(a)$ with all the ones contained on positions $\{k, \ldots, n\}$, i.e. $\min\{i \mid b_i = 1\} \geq k$.*

Proof. The proof is by induction on m, where $n = 2^m$. The case for $m = 1$ is obvious. We assume that the property is proven for some $m \geq 1$ and that $n = 2^m$. Let $I^1_{2n} = (n, 2m+1, (L_1, L_2, \ldots, L_{2m+1}))$. Let $a = (a_1, a_2, \ldots, a_{2n})$ be a zero-one sequence such that for some k, $1 \leq k < n$, only one of the elements a_i, for $1 \leq i \leq k$, is equal to one. Let $a^{(0)} = a$ and for $1 \leq t \leq 2m+1$ let $a^{(t)} = L_t(a^{(t-1)})$. For $0 \leq t \leq 2m+1$ let $i_t = \min\{i \mid a^{(t)}_i = 1\}$. We have to show that $i_{2m+1} \geq k$. If $i_0 = k$, then $i_{2m} \geq k$ because I^1_{2n} is a standard network. Consider the case $i_0 < k$. Then i_1 is odd. Let $a' = (a'_1, a'_2, \ldots, a'_n)$ be such that $a'_i = a^{(1)}_{2i-1}$, for $1 \leq i \leq n$. Let $k' = \lceil k/2 \rceil$. Note that only one of the elements $a'_1, \ldots, a'_{k'}$ is equal to 1 and that for all i such that $2 \leq 2i \leq k$, $a^{(1)}_{2i} = 0$. Let $b' = (b'_1, b'_2, \ldots, b'_n)$, where $b'_i = a^{(2m)}_{2i-1}$. By remark 4 we have $b' = I^1_n(a')$ and for $2 \leq 2i \leq k$: $a^{(2m)}_{2i} = a^{(1)}_{2i} = 0$. By induction hypothesis $\min\{i \mid b'_i = 1\} \geq k'$, i.e. $i_{2m} \geq 2\lceil k/2 \rceil - 1$. If k is even and $i_{2m} = 2\lceil k/2 \rceil - 1 = k-1$, then due to the comparator $(k-1, k)$ from L_{2m+1} we have $i_{2m+1} = k$. In all remaining cases $i_{2m} \geq k$, and of course $i_{2m+1} \geq i_{2m}$.

The Figure 1 shows I^1_n and I^0_n for $n = 16$, where I^0_n is a network dual to I^1_n. $I^0_n = (n, 2m-1, (L'_1, \ldots, L'_{2m-1}))$ with $L'_t = \{(R_i, R_j) \mid (R_{n-j+1}, R_{n-i+1}) \in L_t\}$, where L_t is the t-th layer of I^1_n. We will use I^1_n and I^0_n in the construction of our network.

4 Construction of the correction network

We assume that the input is k-damaged 0-1-sequence a, where $3k(2\log n)^{\log k} < n/2$ (which is equivalent to $k < (n/6)^{1/(\log\log n + 2)}$). Let x be the number of zeroes in a. Below we describe the construction of the correction network $N_{n,k} = (n, D, (L_1, \dots, L_D))$.

First we arrange the registers in the matrix $n_1 \times n_2$ in row-major order (i.e. n_1 is the number of rows and n_2 is the number of columns). We assume that $n = n_1 \cdot n_2$ where n_1 is the power of two. The parameter n_2 will be defined later. The rows are numbered from one to n_1 and the columns are numbered from one to n_2. The i-th row contains registers $R_{(i-1)\cdot n_2+1}, \dots, R_{i\cdot n_2}$. The j-th column contains the registers $R_{j+(k-1)\cdot n_2}$, for $1 \le k \le n_1$. Note that the rows $1, \dots, x'-1$, where $x' = \lfloor x/n_1 \rfloor + 1$, are contained in the region for zeroes and the rows $x'+1, \dots, n_1$ are contained in the region for ones. The row x' may intersect both regions.

4.1 Phase 1

For each row separately we build a Batcher sorting network [2] that sorts it from left to right. Let B_{n_2} be a Batcher sorting network for input size n_2. Let d be the depth of B_{n_2}. The first d layers of our network are defined as follows: $L_t = \bigcup_{i=1}^{n_1} L_{i,t}$, where $L_{i,t}$ is the t-th layer of B_{n_2} with the set of registers substituted by the i-th row of our matrix. This defines the first $d \approx 0.5 \log^2 n_2$ layers of comparators. Note that after applying this part of the network we have following configuration (see Figure 2): In the sub-matrix consisting of the rows $1, \dots, x'-1$ all the (displaced) ones are in the k rightmost columns, and in the sub-matrix consisting of the rows $x'+1, \dots, n_1$ all the (displaced) zeroes are in the k leftmost columns.

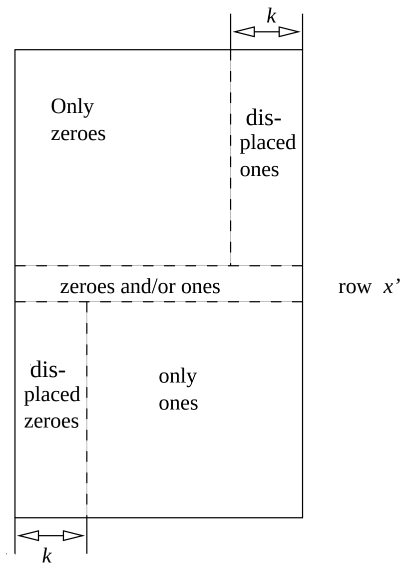


Fig. 2. The configuration in the matrix of registers after applying subnetwork of Phase 1.

4.2 Phase 2

Let T_d denote a tree with the edges labeled with positive integers defined as follows: T_0 contains only one vertex (the root) and no edges. For $d \ge 0$, T_{d+1} (see Figure 3) consists of two trees T'_d, where T'_d denotes T_d with all edges' labels increased by one. The root of

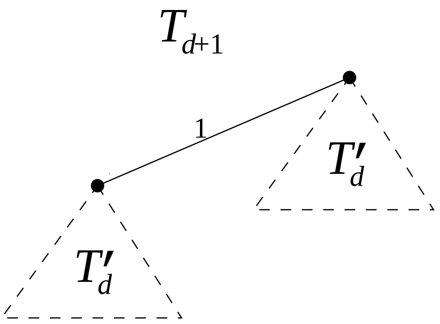


Fig. 3. The tree T_{d+1}.

the first T'_d is connected with additional edge labeled 1 as a leftmost child of the root of the second T'_d. The root of the second T'_d is the root of T_{d+1}. Note that any node may be incident to many edges with different labels. Let T^t_d denote the subtree of T_d consisting of the vertices on the *levels* (the distance from the root) not greater than t. The nodes of T^t_d are ordered (starting from zero) in preorder, i.e. the root is the node with the number zero, and the following numbers are assigned to the subtrees rooted at it's children from left to right. Let α^t_d denote the number of vertices in T^t_d. Since each vertex has at most d children we have: $\alpha^t_d \leq \sum_{i=0}^{t} d^i \leq d^t(1+1/(d-1))$. We define the number of columns in the matrix of registers as $n_2 = 2k \cdot \alpha^{\lceil \log k \rceil}_{2 \cdot \log n_1 - 1}$. Note that $n_2 \leq 3k(2 \log n)^{\log k}$ for n big enough. Since now let α denote $\alpha^{\lceil \log k \rceil}_{2 \cdot \log n_1 - 1}$.

To describe this phase of the construction of the network, we assume that the columns of the matrix of registers are the nodes of a graph G that consists of two disjoint forests: the *left* and the *right forest*. The left forest will be used to move up the displaced zeroes while the right forest will be used to move down the displaced ones. Each one of the two forests consists of k trees $T^{\lceil \log k \rceil}_{2 \cdot \log n_1 - 1}$. The $n_2/2$ columns from the left half of the matrix are the nodes of the left forest. The k leftmost columns are the roots of the trees. The column $i \cdot k + j + 1$ is the i-th node of the tree rooted at column $j+1$, where $0 \leq i \leq \alpha - 1$ and $0 \leq j \leq k-1$. Symmetrically the $n_2/2$ columns from the right half of the matrix are the nodes of the right forest and the k rightmost columns are the roots of the trees. The column $n_2 - (i \cdot k + j)$ is the i-th node of the three rooted at column $n_2 - j$, where $0 \leq i \leq \alpha - 1$ and $0 \leq j \leq k-1$.

Recall that we have already defined the networks $I^1_{n_1}$ and $I^0_{n_1}$ and that L_d is the last layer from the previous phase. We place the comparators in the next $4 \log n_1 - 2$ layers. For $1 \leq l \leq 2 \cdot \log n_1 - 1$ the layers L_{d+2l-1} and L_{d+2l} are defined as follows:

- For each column C from the left (respectively right) forest, which is a root of some tree or is connected with its parent by an edge with a label less than l, if in the l-th layer of $I^0_{n_1}$ (respectively $I^1_{n_1}$) there is a comparator (R_{r_1}, R_{r_2}), then there is a comparator between r_1-th and r_2-th register of C in the layer L_{d+2l-1}, i.e. a comparator $(R_{j+(r_1-1)n_2}, R_{j+(r_2-1)n_2})$, where j is the number of C.
- For each column C, which is a parent of some edge e labeled by l, if C is from the left (respectively right) forest and in the l-th layer of $I^0_{n_1}$ (respectively $I^1_{n_1}$) there is an comparator (r_1, r_2), then there is an comparator $(R_{j'+(r_1-1)n_2}, R_{j+(r_2-1)n_2})$ (respectively $(R_{j+(r_1-1)n_2}, R_{j'+(r_2-1)n_2})$) in the layer L_{d+2l}, where j is the index of C and j' is the index of a column connected with C by the edge e.

Note that if the l-th layer of $I^1_{n_1}$ contains comparator (R_i, R_j), then for each column C from the right forest that is a parent of some edge e labeled l there are two comparators connected to the i-th register of C in the layers L_{d+2l-1} and L_{d+2l} respectively. The first one has its lower endpoint at the j-th register of C while the the second one has its lower endpoint at the j-th register of column C' connected to C by the edge e. The idea is that if the displaced one from i-th register of C is blocked by another displaced one from j-th register of C in the layer L_{d+2l-1}, then it has in the layer L_{d+2l} a second chance of reaching the row j but this time in the column C'. Note that the column C' does not contain any displaced ones (above the row x') until the layer L_{d+2l} is applied. Therefore the one from i-th register of C can not be blocked by a displaced one (above the row x') in the layer L_{d+2l}.

Lemma 6 *After applying the layers of the first and the second phase all displaced zeroes are in rows x' and $x'+1$ and all displaced ones are in rows x' and $x'-1$.*

Proof. We prove the part for displaced ones. (The proof for displaced zeroes is analogous.) Consider the situation just before second phase (Figure 2). All the displaced ones in the rows above x' are in the k rightmost columns. Each of these columns is a root of one of the trees from the right forest. By the definition, there are no comparators between registers from the columns being nodes of distinct trees. For any two trees T_1, T_2 from the right forest let M_1 and M_2 denote the sub-matrices consisting of the columns being nodes of T_1 and T_2 and let N_1, N_2 denote the subnetworks of $(n, 4\log n_1 - 2, (L_{d+1}, \ldots, L_{d+4\log n_1 - 2}))$ restricted to the registers of M_1 and M_2 respectively. Then N_1 and N_2 are isomorphic. Therefore it is sufficient to consider the configuration of the zeroes and ones in the columns of the tree T rooted at the rightmost column.

Let $a^{(0)}$ denote a sequence that is an output of the previous phase (i.e. of $(n, d, (L_1, \ldots, L_d))$). For $1 \le t \le 4\log n_1 - 2$ let $a^{(t)} = L_{d+t}(a^{(t-1)})$. Let $c^{(i,t)}$ be a sequence of length n_1 that is the contents of i-th column after application of L_{d+t} (i.e. $c_j^{(i,t)} = a_{i+(j-1)n_2}^{(t)}$). Let M be a sub-matrix consisting of the nodes of tree T. Let $m_1, m_2, \ldots, m_\alpha$ be the indices of the columns included in M, where column m_1 is a root of T. Let k' be the number of displaced ones in $a^{(0)}$ that are in column m_1 above the row x', say in the rows $p_1, p_2, \ldots, p_{k'}$. For $1 \le i \le k'$ let $s^{(i,0)}$ be a sequence of length n_1 such that $s_j^{(i,0)} = 1$ if and only if $j = p_i$ or $j \ge x'$. For $1 \le t \le 2 \cdot \log n_1 - 1$ let $s^{(i,t)} = L'_t(s^{(i,t-1)})$, where L'_t is the t-th layer of $I^1_{n_1}$.

Proposition 7 *For $0 \le l \le \lceil \log k \rceil$ if m_i is an index of a column that is a node of T at level l, then there are at most $k'/2^l$ ones above the row x' in column m_i in all sequences $a^{(t)}$, for $0 \le t \le 4\log n_1 - 2$ (i.e. $\sum_{j=1}^{x'-1} c_j^{(m_i,t)} \le k'/2^l$).*

Proof of Proposition 7 (Induction on l.) The only node at level 0 is the root of T that contains k' ones above the row x'. Let column m_i be a node at level l, $l \ge 1$. Let column m_h be a parent of a column m_i. Let l' be a label of an edge of T connecting column m_h with m_i. For $t < 2l'$ we have $\sum_{j=1}^{x'-1} c_j^{(m_i,t)} = 0$ since there is no comparator between column m_i and any other column in the layers $L_{d+1}, \ldots, L_{d+t}$. By induction hypothesis $\sum_{j=1}^{x'-1} c_j^{(m_h,2l'-1)} \le k'/2^{l-1}$. For $j \le x'-1$ if $c_j^{(m_i,2l')} = 1$, then following conditions have to take place: There is a comparator $(R_{m_h+(j'-1)n_2}, R_{m_j+(j-1)n_2})$ in $L_{d+2l'}$ for some $j' < j$, and $c_{j'}^{(m_h,2l'-1)} = 1$ and there is a comparator $(R_{m_h+(j'-1)n_2}, R_{m_h+(j-1)n_2})$ in $L_{d+2l'-1}$ and $c_j^{(m_h,2l'-1)} = 1$. Since all such pairs $\{j, j'\}$ are pairwise disjoint, there can be at most $\sum_{j=1}^{x'-1} c_j^{(m_h,2l'-1)}/2 \le k'/2^l$ of them. For $t > 2l'$ there are no comparators in L_{d+t} with maximum endpoint in column m_i and minimum endpoint in another column. This completes the proof of Proposition 7.

Corollary 8 *Any leaf of T at level $\lceil \log k \rceil$ may contain at most 1 one above the row x'.*

Proposition 9 *For each t, $0 \le t \le 2 \cdot \log n_1 - 1$, holds $\bigvee_{i=1}^{\alpha} c^{(m_i,2t)} \le \bigvee_{j=1}^{k'} s^{(j,t)}$.*

Proof of Proposition 9 For $t = 0$ the proposition follows from the fact that all ones in the sub-matrix M above the row x' are located in the column m_1 and from the definitions of sequences $c^{(i,0)}$ and $s^{(j,0)}$. Assume that for some $t \ge 0$, $\bigvee_{i=1}^{\alpha} c^{(m_i,2t)} \le \bigvee_{j=1}^{k'} s^{(j,t)}$. Let m_i be one of the columns of T. If $c_r^{(m_i,2t)} = 1$ for some $r < x'$, then there is some j, $1 \le j \le k'$, such that $s_r^{(j,t)} = 1$. If there is no comparator $(R_r, R_{r'})$, with $r < r' < x'$, in the layer L'_{t+1} of $I^1_{n_1}$, then $s_r^{(j,t+1)} = 1$. Otherwise $s_{r'}^{(j,t+1)} = 1$ and there is an comparator $(R_{m_i+(r-1)n_2}, R_{m_i+(r'-1)n_2})$ in L_{d+2t+1} and either m_i is a leaf at level $\lceil \log k \rceil$ of T (there are no comparators inside leaves at lower levels) or there is a comparator $(R_{m_i+(r-1)n_2}, R_{m_g+(r'-1)n_2})$ in L_{d+2t+2}, where m_g is a child of column m_i connected with it by edge labeled $t+1$. If m_i is a leaf at level $\lceil \log k \rceil$, then by Corollary 8 we have $c_{r'}^{(m_i,2t)} = 0$; hence $c_r^{(m_i,2t+1)} = c_r^{(m_i,2t+2)} = 0$. If m_i is not a leaf of T, then $c_{r'}^{(m_g,2t+1)} = 0$ (because column m_g does not receive any displaced ones above the row x' before the application of L_{d+2t+2}) and it follows that $c_r^{(m_i,2t+2)} = 0$ (even if $c_{r'}^{(m_i,2t)} = 1$).

The Lemma 6 follows from Proposition 9 and from the fact that, by Lemma 5, for all i, $1 \le i \le k'$, $\min\{j \mid s_j^{(i,2\cdot\log n_1-1)} = 1\} \ge x' - 1$.

4.3 Phase 3

After executing previous phase we have only zeroes above row $x' - 1$, only ones below row $x' + 1$ and perhaps some dirty region in rows $x' - 1$, x' and $x' + 1$. The next layers are defined as follows. For $1 \le t \le d$ let $L_{d+4\log n_1-2+t} = L_t$. (Those layers sort the rows again.) Let M_{2n_2} be a Batcher merging network [2,3] that merges two sorted sequences placed in registers $R_1, \ldots, R_{n_1}$ and $R_{n_1+1}, \ldots, R_{2n_1}$. Let c denote the depth of M_{2n_2}. For $1 \le t \le c$ let

$$L_{(2d+4\log n_1-2)+t} = \bigcup_{i=1}^{n_1/2} L'_{2i-1,t}$$

$$L_{(2d+4\log n_1-2)+c+t} = \bigcup_{i=1}^{n_1/2-1} L'_{2i,t}$$

$$L_{(2d+4\log n_1-2)+2c+t} = \bigcup_{i=1}^{n_1/2} L'_{2i-1,t},$$

where $L'_{j,t}$ denotes the t-th layer of M_{2n_2} with the set of registers substituted by the registers from rows j and $j+1$ of our matrix.

Lemma 10 *The output of the layer $L_{(2d+4\log n_1-2)+3c}$ is a sorted sequence.*

Proof. For $0 \leq t \leq 3$ let a_t, b_t, c_t denote the numbers of ones in the output of $L_{2d+4\log n_1 - 2 + t\cdot c}$ in the rows $x'-1$, x', $x'+1$ respectively. (We omit the cases: $x' = 1$ and $x' = n_1$.) We have

$$0 \leq a_t, b_t, c_t \leq n_2 \tag{1}$$

and since the row x' is on the border between the regions for zeroes and for ones, we have

$$n_2 \leq a_t + b_t + c_t \leq 2n_2. \tag{2}$$

Assume that x' is even (the case for odd x' is symmetrical). First we merge rows $x'-1$ and x' and we have $b_1 = \min\{n_2, a_0 + b_0\}$ and by equations 1, 2 we have $b_1 + c_0 \geq n_2$. Therefore after merging the rows x' and $x'+1$, the row $x'+1$ is filled with ones (i.e. $c_2 = n_2$). Since $c_2 = n_2$ we have by equation 2: $a_2 + b_2 \leq n_2$ and after the next merging of the rows $x'-1$ and x' the row $x'-1$ is filled with zeroes. Since all the rows remain sorted, the whole output is also sorted.

5 Conclusions

The total depth of the network is $D = 4\log n_1 - 2 + 2d + 3c$, where $d \approx 0.5\log^2 n_2$ is a depth of Batcher sorting network for input size n_2 and $c = \log(2n_2)$ is the depth of Batcher merging network for input size $2n_2$. Thus $D \approx 4\log n + 2\log^2(3k(2\log n)^{\log k}) + 3\log(6k(2\log n)^{\log k}) = 4\log n + 2(\log k \log\log n)^2 + o((\log k \log\log n)^2)$. This completes the proof of Theorem 1. As mentioned in introduction, the depth of $N_{n,k}$ is $4\log n + o(\log n)$ if k is $o\left(2^{\sqrt{\log n}/\log\log n}\right)$. It seems to be interesting problem to find efficient networks of logarithmic depth for larger values of k.

6 Acknowledgment

Thanks to Mirek Kutyłowski for presenting the problem and many helpful discussions on the subject.

References

[1] M. Ajtai, J. Komlós and E. Szemerédi, Sorting in $c\log n$ parallel steps. *Combinatorica* **3** (1983) 1–19.

[2] K. E. Batcher, Sorting networks and their applications, In *AFIPS Conf. Proc. 32*, 307–314, 1968.

[3] D. E. Knuth, *The art of Computer Programming*, Vol. 3:*Sorting and Searching* (Addison-Wesley, 1973).

[4] I. D. Scherson, S. Sen, and A. Shamir, Shear-sort: A true two-dimensional sorting technique for VLSI networks, in *Proc. IEEE International Conference on Parallel Processing*, 1986, 903–908.

[5] M. Schimmler and C. Starke, A Correction Network for N-Sorters, *SIAM J. Comput.* **6**, No. 6, 1989.

[6] A. C. Yao, Bounds on selection networks. *SIAM J. Comput.* **9**, No. 3, 1980.