Time Lower Bounds for Parallel Sorting on a Mesh-Connected Processor Array *

Yijie Han [1] ** and Yoshihide Igarashi [2]

[1]Department of Computer Science
University of Kentucky
Lexington, KY 40506

[2]Department of Computer Science
Gunma University
Kiryu, 376 Japan

Summary

We prove that $(1+\sqrt{6}/2)n \approx 2.22n$ is a time lower bound independent of indexing schemes for sorting n^2 items on an $n \times n$ mesh-connected processor array. We distinguish between indexing schemes by showing that there exists an indexing scheme which is provably worse than the snake-like row-major indexing for sorting. We also derive lower bounds for various indexing schemes. All these results are obtained by using *the chain argument* which we provide in this paper.

1 Introduction

Parallel sorting algorithms and their time complexities have been intensively studied[1-4,6,8-20]. Although there exist sorting algorithms of time complexity $O(\log N)$ [1,2,4,9,12], the structure of such algorithms are complicated and their realization is extremely difficult. A mesh-connected processor array is widely accepted as a realistic model of parallel computers. It consists of $N = n^2$ identical processors arranged in a square. A number of parallel sorting algorithms on the mesh-connected model have been reported[8,10,11,13-20]. Some of these algorithms are good in the practical sense and fast for realistic values of n. Schnorr and Shamir [18] and Ma et al.[10] have designed asymptotically fast sorting algorithms on the mesh-connected model. The time complexities of their algorithms are $3n + O(n^{3/4})$ steps and $4n + O(n^{3/4}\log n)$ steps, respectively.

Kunde[6] and Schnorr and Shamir [18] have proved a $3n - 2n^{1/2} - 3$ lower bound on the number of steps for sorting n^2 items into the snake-like row-major order. They used a technique called the joker-zone argument[6,18]. Since these discoveries, it becomes curious whether the snake-like row-major indexing is the best indexing scheme for sorting[18]. A question whether the distance bound of $2n$ is ultimately achievable by using some kind of super indexing schemes has also been raised[10].

*A preliminary version of this paper was presented on the 3rd International Workshop on Parallel Computation and VLSI Theory, Corfu Island, Greece, (June-July, 1988), Lecture Notes in Computer Science, Vol. 319, pp. 434-443(1988).

**Research supported in part by the University of Kentucky research initiation award.

The only known lower bound independent of indexing schemes for sorting on an $n \times n$ mesh-connected processor array was the distance bound of $2n$. No previous results showed that some indexing schemes are better than others.

The aim of this paper is to provide a general technique to derive lower bounds for various indexing schemes and to derive a good lower bound independent of indexing schemes. In this paper we show that $2.22n$ is a lower bound for any indexing scheme on the $n \times n$ mesh-connected processor array. We therefore answered the question posed by Ma et al.[10]. We use a tool which we call the chain argument to obtain this lower bound. This chain argument is a generalization of the joker-zone argument used by Kunde[6] and Schnorr and Shamir[18]. It combines the new idea of chaining with the joker-zone argument. By using the chain argument we give a constructive proof showing that $4n - 2(2n)^{1/2} - 3$ is a time lower bound for sorting by a certain indexing scheme. Combined with Schnorr and Shamir's result[18], this shows that some indexing schemes are provably worse than the snake-like row-major indexing for sorting. By using the chain argument we also derive lower bounds for various indexing schemes.

The present paper is divided into 6 sections. In section 2 we will define our mesh-connected model, illustrate commonly used indexing schemes and give some terminologies. In section 3, a basic tool, called the chain argument, is described. In section 4 we derive time lower bounds for various indexing schemes. We also show the existence of a poor indexing scheme. In section 5 we derive a good lower bound independent of indexing schemes.

2 Preliminaries

We consider a general model of a synchronous $n \times n$ mesh-connected processor array as given in [18]. The processor array is denoted by $M[1..n, 1..n]$. Each processor at location (i, j), $1 \le i, j \le n$, is denoted by $M[i, j]$. A subarray consisting of processors $M[i, j]$ such that $p \le i \le q$ and $r \le j \le t$ is denoted by $M[p..q, r..t]$. Processor $M[i, j]$ is directly connected to its neighbors $M[i, j-1]$, $M[i-1, j]$, $M[i+1, j]$ and $M[i, j+1]$, provided they exist. This model is shown in Fig. 1. All n^2 processors work in parallel with a single clock, but they may run different programs. As for sorting computation, the initial contents of $M[1..n, 1..n]$ are assumed to be a permutation of n^2 linearly ordered data items, where each processor has exactly one data item. The final contents of $M[1..n, 1..n]$ are the sorted sequence of the items in a specified order. In one step each processor can communicate with all of its directly-connected neighbors. The interchange of data items and the replacement of the data item in a processor with the data item of its directly connected processor can be done in one step. We should note that a data item can only be moved to one of its neighbors in one step. The computing time is defined to be the number of parallel steps of the basic operations to reach the final configuration of the processor array. As pointed out by Schnorr and Shamir[18], such a model is stronger than most models used for deriving upper bounds.

An *indexing scheme* I on array $M[1..n, 1..n]$ is a one-to-one mapping from $\{1, ..., n\} \times \{1, ..., n\}$ to $\{1, ..., n^2\}$, i.e. $I : (i, j) \to k$. Sorting by an indexing scheme I is to sort the items into the order defined by I.

The row-major indexing and the snake-like row-major indexing are commonly accepted ways to order the processor array. Other indexing schemes are also used[3]. They are

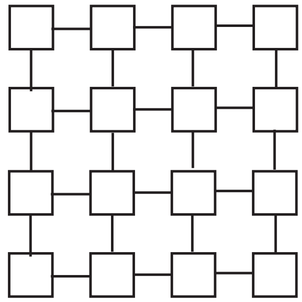

Fig. 1. A mesh-connected processor array

diagonal-major indexing, snake-like diagonal-major indexing, file-major indexing, and q-column-block-major indexing. These indexing schemes are shown in Fig. 2. The rules of indexing for these schemes may be clear from the illustrations. We briefly explain how processors are ordered by the file-major indexing and by the q-column-block-major indexing in the following. In the file-major indexing the whole processor array $M[1..n, 1..n]$ is divided into 4 subfiles $S_1 = M[1..n/2, 1..n/2]$, $S_2 = M[1..n/2, n/2+1..n]$, $S_3 = M[n/2+1..n, 1..n/2]$, $S_4 = M[n/2+1..n, n/2+1..n]$. Any processor in S_i proceeds any processor in S_j if $i < j$. The order of the processors in the same subfile is recursively defined in the same way. In the q-column-block-major indexing the i-th block is $M[1..n, iq+1..(i+1)q]$, $i = 0, 1, ..., n/q-1$, where n is assumed to be a multiple of q. The order of any processor in the i-th block proceeds the order of any processor of the j-th block if $i < j$. The order of processors in the same block is by the row-major indexing.

The distance between $M[i_1, j_1]$ and $M[i_2, j_2]$ is defined as $|i_1-i_2|+|j_1-j_2|$ and denoted by $d((i_1, j_1), (i_2, j_2))$. A zone S of $M[1..n, 1..n]$ is denoted by a subset of $\{1, ..., n\} \times \{1, ..., n\}$. The area of zone S is the number of processors in S. For convenience $M[1..n, 1..n]$ also denotes the zone of all processors in $M[1..n, 1..n]$. For an indexing scheme I the index of $M[i, j]$ is denoted by $I(i, j)$. For example, if I is the row-major indexing on $M[1..4, 1..4]$, $I(2, 4) = 8$. For an indexing scheme I, a chain C is a sequence of pairs $((i_1, j_1), (i_2, j_2), ... (i_c, j_c))$, $(i_k, j_k) \in \{1, ..., n\} \times \{1, ..., n\}$, $1 \le k \le c$, such that $I(i_1, j_1)$, $I(i_2, j_2)$, ..., $I(i_c, j_c)$ is a consecutive integer sequence in increasing order. In this case the length of C is $c - 1$. For the row-major indexing on $M[1..4, 1..4]$, $((2, 4), (3, 1), (3, 2), (3, 3))$ is a chain of length 3. It is obvious that no chain in $M[1..n, 1..n]$ could have its head and tail joined to form a circle.

3 The Chain Theorem

We shall frequently consider the computation performed in a duration called a sweep. As shown in Fig. 3(a), a sweep is defined by two lines and a direction. The two lines are parallel to a diagonal of the mesh-connected processor array and the direction is orthogonal to the lines. One of the lines is called the start line and the other is called the stop line. The computation performed during the sweep as specified in Fig. 3(a) is shown in Fig. 3(b).

1	**2**	**3**	**4**
5	**6**	**7**	**8**
9	**10**	**11**	**12**
13	**14**	**15**	**16**

a Row-major indexing

1	**2**	**3**	**4**
8	**7**	**6**	**5**
9	**10**	**11**	**12**
16	**15**	**14**	**13**

b Snake-like row-major indexing

1	**3**	**6**	**10**
2	**5**	**9**	**13**
4	**8**	**12**	**15**
7	**11**	**14**	**16**

c Diagonal-major indexing

1	**3**	**4**	**10**
2	**5**	**9**	**11**
6	**8**	**12**	**15**
7	**13**	**14**	**16**

d Snake-like diagonal-major indexing

1	**2**	**5**	**6**
3	**4**	**7**	**8**
9	**10**	**13**	**14**
11	**12**	**15**	**16**

e File-major indexing

1	**2**	**9**	**10**
3	**4**	**11**	**12**
5	**6**	**13**	**14**
7	**8**	**15**	**16**

f 2-column-block-major indexing

Fig. 2. Indexing schemes

The frontier of the computation is initially at the start line x and propagates along direction Z. When the frontier hits the extreme of the processor array (point A in Fig. 3(a)), it will be bounced back and then propagates backwards until it hits the stop line y. Referring to Fig. 3, the length of the sweep is defined to be the sum of the distance from line x to point A and the distance from point A to line y. Therefore the length of a sweep represents the minimum number of computing steps required to accomplish the sweep. The region enclosed by the start line x and corner D is called the stretching zone. This stretching zone is called the joker-zone in the joker-zone argument[6]. Our idea is to use the items in the stretching zone to stretch one position into a chain. This is demonstrated in the proof of the chain theorem. The region enclosed by stop line y and corner A is called the residing zone. We shall demonstrate the existence of a chain of a certain length in the residing zone. Notice that if the length of a sweep is no less than $2n-2$ then the area of residing zone is no less than the area of the stretching zone.

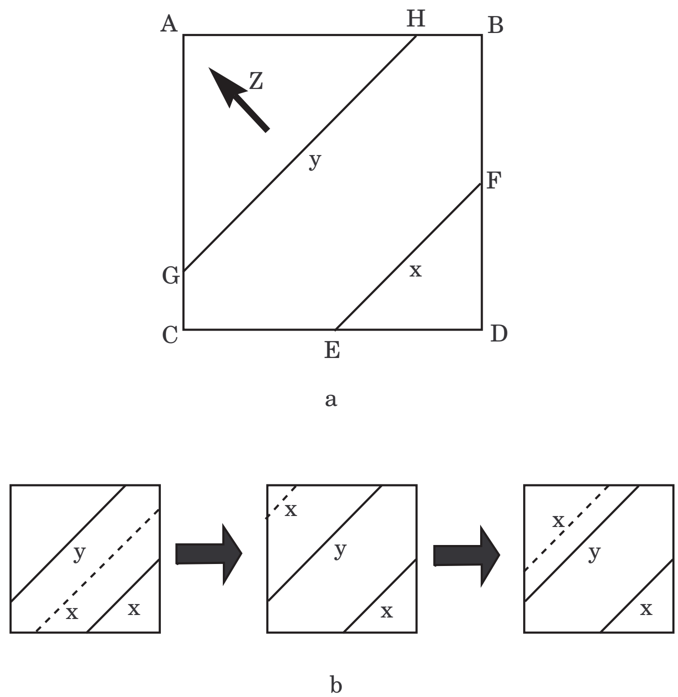

Fig. 3. A sweep

Our first theorem, called the chain theorem, relates the computing time of a sorting algorithm, the length of a sweep on mesh M and the length of a chain in the residing zone.
Theorem 1(Chain Theorem): *For any sorting algorithm of time complexity T and a sweep of length $T+1$, there exists a chain C of length S in the residing zone of the sweep, where S is the area of the stretching zone.*
Proof: Let $SORT$ be a sorting algorithm of time complexity T. Referring to Fig. 3(a), let k be the distance between line x and point A. During the first $k-1$ steps of $SORT$ the computation happens at point A is not affected by the initial contents in the stretching

zone (in Fig. 3(a) the stretching zone is ΔFED). Let a be the item which is at point A immediately after the $(k-1)$-st step of $SORT$. Item a is independent of the initial contents of any processor in the stretching zone, and its final sorted position must be in the residing zone since the time complexity of $SORT$ is T. By assigning different input values to the processors in the stretching zone, we can force (stretch) a into $S+1$ different sorted positions. These different positions are all within the residing zone and they form a chain of length S. □

The next theorem is immediate from the Chain Theorem and will be often used to show time lower bounds for various indexing schemes in the following sections.

Theorem 2: *For an indexing scheme I and a sweep of length T, if there is no chain of length S in the residing zone, where S is the area of the stretching zone, then there is no algorithm of time complexity $T-1$ for sorting the items on the mesh-connected processor array by indexing scheme I (i.e., T is a time lower bound for sorting by I).* □

4 Lower Bounds for Various Indexing Schemes

By using the chain argument we can derive new lower bounds for various indexing schemes. Theorems 3 and 4 in the following were proved by Kunde[6] and by Schnorr and Shamir[18]. Here we give alternative proofs by using the chain argument.

Theorem 3: *A lower bound for sorting by the row-major indexing is $3n-(2n)^{1/2}-2$ steps.*

Proof: We define the sweep (shown in Fig. 4, where $k=(2n)^{1/2}$) such that the stretching zone and the residing zone of the sweep are $\{M[i,j]|d((i,j),(n,1)) \leq (2n)^{1/2}-1\}$ and $\{M[i,j]|d((i,j),(1,n)) \leq n-1\}$, respectively. The length of the sweep is $3n-(2n)^{1/2}-2$. The area of the stretching zone is $k(k+1)/2 > n-1$. However, the longest chain in the residing zone has length $n-1$. By Theorem 2, a lower bound for sorting by the row-major indexing is $3n-(2n)^{1/2}-2$ steps. □

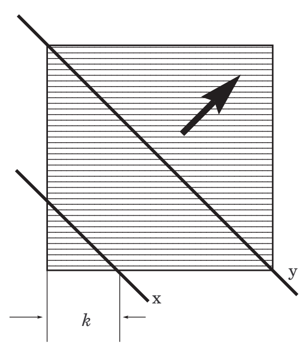

Fig. 4. A sweep for the raow-major indexing

Theorem 4: *A lower bound for sorting by the snake-like row-major indexing is $3n-2(n-2)^{1/2}-2$ steps.*

Proof: Consider the sweep (shown in Fig. 5, where $k=2(n-2)^{1/2}$) such that the stretching zone and the residing zone are $\{M[i,j]|d((i,j),(n,n)) \leq 2(n-2)^{1/2}-1\}$ and $\{M[i,j]|d((i,j),(1,1)) \leq n-1\}$. The length of the sweep is $3n-2(n-2)^{1/2}-2$. The area of the stretching zone is $2n+(n-2)^{1/2}-4$ which is greater than $2n-4$ when $n \geq 2$. However, the longest chain in the residing zone has length $2n-4$. By Theorem 2, a lower bound for sorting by the snake-like row-major indexing is $3n-2(n-2)^{1/2}-2$ steps. □

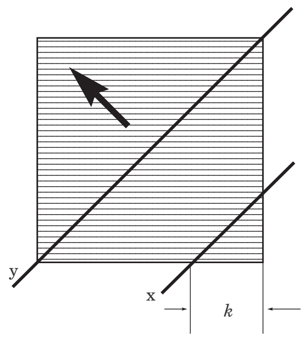

Fig. 5. A sweep for the snake-like row-major indexing

We use the chain argument to derive new lower bounds. These lower bounds are listed in Examples 1, 2 and Theorem 5. They can also be obtained by the joker-zone argument[6,18]. The idea here is to show how to characterize chains in the residing zone for various indexing schemes.

Example 1: A lower bound for sorting by the diagonal-major indexing or the snake-like diagonal-major indexing is $3n-4$ steps.

Proof: We choose the sweep (shown in Fig. 6) such that the stretching zone and the residing zone are $\{M[i,j] | i \geq j\}$ and $\{M[i,j] | (i,j) \neq (n,1)\}$, respectively. That is, the stretching zone is the lower-left triangle while the residing zone consists of all processors except the one at point C. The length of the longest chain in the residing zone is equal to the area of the stretching zone minus 1. The sweep has length $3n-4$. By Theorem 2, a lower bound for sorting by these indexing schemes is $3n-4$ steps. □

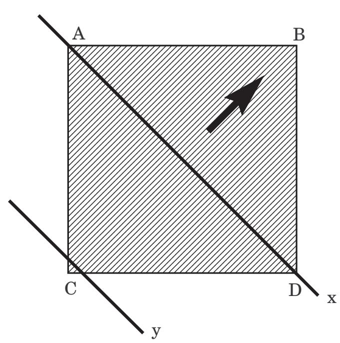

Fig. 6. A sweep for the diagonal-major and the snake-like diagonal-major indexings

Example 2: A lower bound for sorting by the file-major indexing is $2.75n-4$ steps.

Proof: Consider the sweep (shown in Fig. 7.) such that the stretching zone and the residing zone are $\{M[i,j] | i \leq j+n/4-1\}$ and $\{M[i,j] | (i,j) \neq (1,n)\}$, respectively. The length of the sweep is $2.75n - 4$. The area of the stretching zone is $(23n^2 - 12n)/32$. However, the longest chain in the residing zone has length $n^2(1/2 + 1/8 + ...) < (23n^2 - 12n)/32$. Hence, by Theorem 2, a lower bound for sorting by the file-major indexing is $2.75n - 4$ steps. □

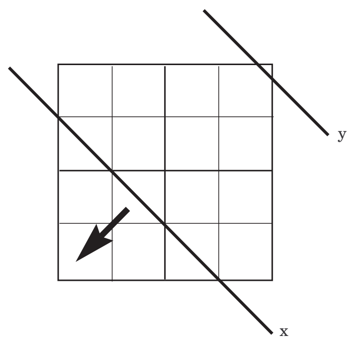

Fig. 7. A sweep for the subfile-major indexing

We demonstrate the variation of the length of chains in the (n/k)-column-block major indexing scheme by proving the following theorem.

Theorem 5: *Let n be a multiple of k. A lower bound for sorting by the* (n/k)*-column-block-major indexing is* $3n + 2n/k - (2/k)^{1/2}n - 4$ *steps.*

Proof: Consider the sweep (shown in Fig. 8.) such that the stretching zone and the residing zone are $\{M[i,j] | d((i,j),(n,1)) \leq (2/k)^{1/2}n - 1\}$ and $\{M[i,j] | d((i,j),(n,1)) \geq (n/k)(k-2)+1\}$, respectively. The length of the sweep is $3n + 2n/k - (2/k)^{1/2}n - 4$. The length of the longest chain in the residing zone is smaller than the area of the stretching zone. By Theorem 2, a lower bound for the (n/k)-column-block-major indexing is $3n + 2n/k - (2/k)^{1/2}n - 4$ steps. □

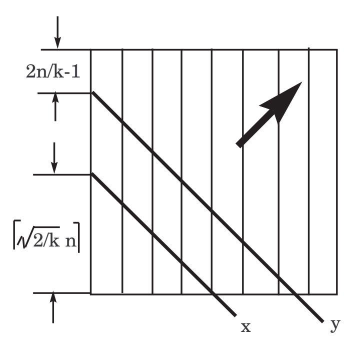

Fig. 8. A sweep for the (n/k)-column-block-major indexing

The relation between lower bounds and values of k is summarized in Table 1 and illustrated in Fig. 9.

By using the chain argument we show that there exists an indexing scheme with a lower bound greater than $3n$.

Theorem 6: *There exists an indexing scheme with lower bound of* $4n - 2(2n)^{1/2} - 3$ *steps*

Table 1. Lower bounds for (n/k)-column-block-major indexing schemes

n/k	Lower bound	n/k	Lower bound
$n/2$	$3n$	$n/10$	$2.752n$
$n/3$	$2.850n$	$n/16$	$2.771n$
$n/4$	$2.792n$	$n/32$	$2.812n$
$n/6$	$2.775n$	$n/64$	$2.854n$
$n/8$	$2.750n$	$n/128$	$2.890n$

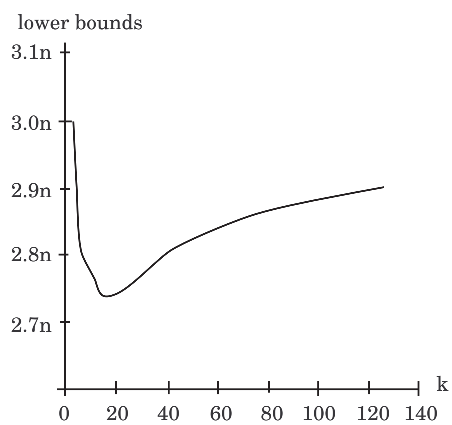

Fig. 9. Lower bounds for (n/k)-column-block-major indexing

for sorting.
Proof: Consider the sweep defined in Fig. 10(a). Notice that the start line is immediate to the lower right of the stop line. Dividing the sorted chain of length $n^2 - 1$ evenly into $S + 1$ chains as shown in Fig. 10(b). Define indexing scheme I such that the S dividing points are located outside the residing zone (i.e., in the stretching zone) and the others are located in the residing zone. The longest chain in the residing zone has length no greater than $\lceil \frac{n^2}{S+1} \rceil$. Let $S = n$. Then the length of the sweep is $4n - 2(2n)^{1/2} - 3$. By Theorem 2, $4n - 2(2n)^{1/2} - 3$ is a lower bound for sorting by indexing scheme I. □

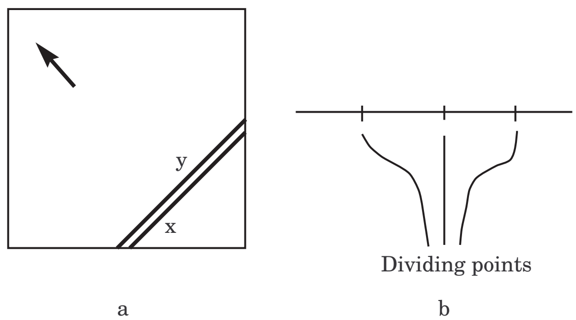

Fig. 10. A poor indexing scheme

Since $3n$ is the known upper bound for sorting by the snake-like row-major indexing[18], we have proved that there exists an indexing scheme which is worse than the snake-like row-major indexing for sorting.

5 A Lower Bound for Any Sorting Order

We apply the chain theorem to derive a lower bound independent of indexing schemes. Here we are interested in a lower bound of the form $2n + \epsilon$, where $\epsilon = \delta n$ and $\delta > 0$. In proving such a lower bound, we can ignore lower order terms because these terms do not contribute to the main term $2n + \epsilon$. In particular, when we say that the area of a zone is a we mean the area is between $a - o(n^2)$ and $a + o(n^2)$.

We first show a weeker lower bound of $2.125n$. The proof of this lower bound is easier to understand and may give more intuition on the application of our chain argument.
Proposition 1: A lower bound independent of indexing schemes for sorting is $2.125n$.
Proof: Suppose that there is an algorithm for sorting in time $2.125n$. Define a sweep of length $2.125n$ as shown in Fig. 11(a). By the chain theorem there is a chain C_1 of length $> 3n^2/8$ in the upper right triangle ΔABD. Now define the other three sweeps as shown in Fig. 11(b). By the chain theorem there is a chain C_2 in ΔBDC, a chain C_3 in ΔDCA and a chain C_4 in ΔCAB. Each of these chains has length $> 3n^2/8$. Consider the triangle ΔBOD as shown in Fig. 11(c). Both chains C_1 and C_2 have more than $3n^2/8 - n^2/4 = n^2/8$ points in ΔBOD, therefore they must join in ΔBOD. For the same reason C_2 and C_3 must join in ΔDOC, C_3 and C_4 must join in ΔCOA, C_4 and C_1 must join in ΔAOB. We

thus obtain a circle. This contradicts the circleless property of any indexing scheme. We therefore conclude that there is a residing zone in which no chain of length $3n^2/8$ exists and, by Theorem 2, a lower bound for sorting is $2.125n$. □

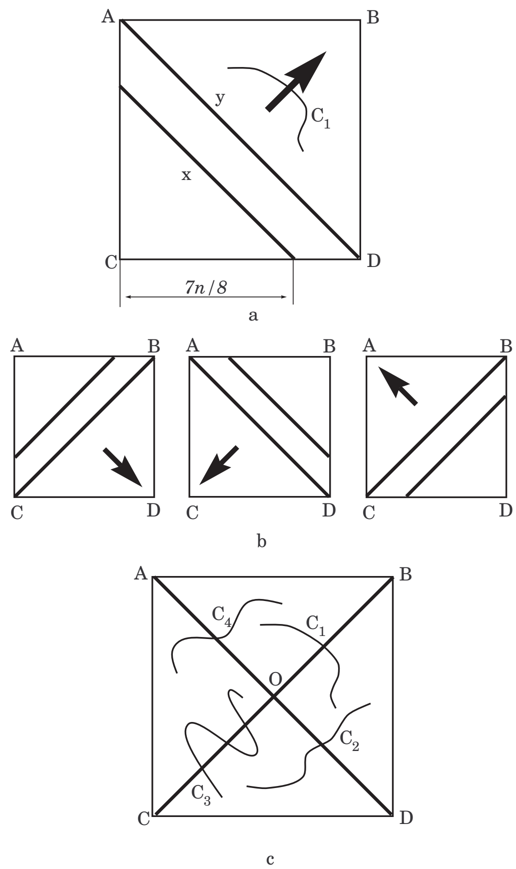

Fig. 11. A sketch for Proposition 1

Now we show the lower bound of $2.22n$.

Theorem 7: *A lower bound independent of index schemes for sorting is* $2.22n$.

Proof: Let I be an arbitrary but fixed indexing scheme. Referring to Fig. 12, the area of square $EFGH$ is $2k^2$, therefore there is a point (i, j) satisfying:

(a). (i, j) is located outside square $EFGH$;

(b). $n^2/2 - k^2 \le I(i, j) \le n^2/2 + k^2$.

Without loss of generality we may let a be point (i, j) as shown in Fig. 12. Because of property (b), the longest chain in polygon $PQDBA$ has length at most $\max\{I(i, j), n^2 - I(i, j)\} \le n^2/2 + k^2$.

Define a sweep such that ST, PQ and Z are the start line, the stop line and the direction of the sweep, respectively. The length of the sweep is $2n + \epsilon$. The area of the stretching zone (polygon $SACDT$) is $n^2 - (n - k + \epsilon)^2/2$.

By Theorem 2, a lower bound of $2n+\epsilon$ will be held when $n^2 - (n-k+\epsilon)^2/2 \ge n^2/2 + k^2$.

Solving the equation $n^2 - (n - k + \epsilon)^2/2 = n^2/2 + k^2$ we obtain

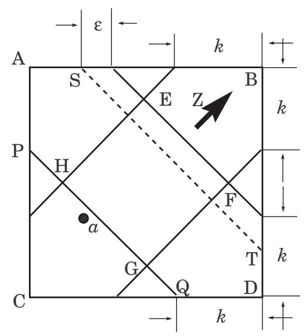

Fig. 12. A sketch for Theorem 7

$$\epsilon = \sqrt{n^2 - 2k^2} - n + k.$$

Let

$$\frac{d\epsilon}{dk} = -2k(n^2 - k^2)^{-1/2} + 1 = 0.$$

We derive $k = n/\sqrt{6}$. Therefore the maximum value of ϵ is $\sqrt{n^2 - n^2/3} - n + n/\sqrt{6} = (\sqrt{6}/2 - 1)n \approx 0.22n$. □

We would like to mention that recent studies have raised the lower bound proved here. In particular, Kunde has observed a lower bound of $2.25n$ [7] and Han et al. have proved a lower bound of $2.27n$ [5].

6 Concluding Remarks

Our main results shown in this paper are a lower bound for any sorting order and the existence of a poor indexing scheme. These results were obtained by using the chain argument. Although it has been shown in this paper that it is impossible to sort in $2n$ steps, the question remains open whether there exist sorting algorithms with less than $3n$ time.

Acknowledgement

We wish to thank F. D. Lewis and M. Truszczynski for comments and discussions on this paper.

References

1. Ajtai, M., J. Komlòs, and E. Szemerèdi. An $O(N \log N)$ sorting network. Proc. 15th ACM Symp. on Theory of Computing, 1-9(1983).

2. Bilardi, G., and Preparata, F.: A minimum area VLSI architecture for $O(\log N)$ time sorting, Proc. 16th Annual ACM Symp. on Theory of Computing, 67-70(1984).

3. Bitton, D., Dewitt, D. T., Hsiao, D. K., and Menon, J.: A taxonomy of parallel sorting, ACM Computing Survey, 16, 287-318(1984).

4. Cole, R.: Parallel merge sort. 27th Symp. on Foundations of Comput. Sci., IEEE, 511-516(1986).

5. Han, Y., Y. Igarashi, M. Truszczynski.: Indexing functions and time lower bounds for sorting on a mesh-connected computer. Technical Report No. 114-88, Dept. of Computer Science, University of Kentucky.

6. Kunde, M.: Lower bounds for sorting on mesh-connected architectures. Acta Informatica, 24, 121-130(1987).

7. Kunde, M.: Bounds for l-selection and related problems on grids of processors. Proceedings of the Fourth International Workshop on Parallel Processing by Cellular Automata and Arrays, Berlin, G.D.R., (Oct. l988).

8. Lang, H.-W., Schimmler, M., Schmech, H. and Schröder, H.: Systolic sorting on a mesh-connected network, IEEE Trans. Comput., C-34, 652-658(1985).

9. Leighton, T.: Tight bounds on the complexity of parallel sorting. IEEE Trans. Comput., C-34, 344-354, 1985.

10. Ma, Y., Sen, S. and Scherson, I. D.: The distance bound for sorting on mesh-connected processor arrays is tight. 27th Symp. on Foundations of Comput. Sci., IEEE, 255-263(1986).

11. Nassimi, D., and Sahni, S.: Bitonic sort on a mesh-connected parallel computer, IEEE Trans. Comput., C-27(1979).

12. Reif, J. H. and Valiant L. G.: A logarithmic time sort for linear size networks, J. ACM, 34, 60-76(1987).

13. Sado, K. and Igarashi, Y.: A divide-and-conquer method of the parallel sort, IECE of Japan, Tech. Commit. of Automata and Languages, AL84-68, 41-50(1985).

14. Sado, K. and Igarashi, Y.: A fast pseudo-merge sorting algorithm, IECE of Japan, Tech. Commit. of Automata and Languages, AL84-68, 41-50(1985).

15. Sado, K. and Igarashi, Y.: Fast parallel sorting on a mesh-connected processor array, Proc. of Japan-U.S. Joint Seminar, Discrete Algorithms and Complexity Theory(Johnson, D.S. et al. eds), Academic Press, New York, 161-183(1986).

16. Sado, K. and Igarashi, Y.: Some parallel sorts on a mesh-connected processor array, J. Parallel and Distributed Computing, Vol. 3, pp. 389-410, 1986.

17. Scherson, I. D., Sen, S. and Shamir, A.: Shear sort: A true two-dimensional sorting technique for VLSI networks, Proc. 1986 Int. Conf. on Parallel Processing, 903-908(1986).

18. Schnorr, C. P. and Shamir, A.: An optimal sorting algorithm for mesh-connected computers, Proc. 18-th ACM Symp. on Theory of Computing, 255-263(1986).

19. Thompson, C. D. and Kung, H. T.: Sorting on a mesh-connected parallel computer, Commun. ACM, 20, 263-271(1977).

20. Thompson, C. D.: The VLSI complexity of sorting, IEEE Trans. Comput. C-32, 1171-1184(1983).