Average Case Analysis of Five Two-Dimensional Bubble Sorting Algorithms*

Serap A. Savari

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139 USA

Abstract

For each of five generalizations of the odd-even transposition sort to a sorting algorithm on a $\sqrt{N} \times \sqrt{N}$ mesh of processors, we demonstrate that with "high probability," the number of steps required to sort a random permutation of N numbers is $\Theta(N)$.

1 INTRODUCTION

The ***odd-even transposition sort***, or ***bubble sort***, is a simple and widely known algorithm for sorting N numbers on an N-cell linear array in at most N word steps. If we number the cells of the linear from left to right by $1, 2, \ldots, N$, then the algorithm can be described as follows. At odd steps, we compare the contents of cells 1 and 2, 3 and 4, etc., switching values if necessary so that the smaller value is stored in the leftmost cell. At even steps, we carry out the same operations for cells 2 and 3, 4 and 5, etc. A history of this algorithm and a proof that it requires at most N steps on any input can be found in [1]. It is also interesting to compute the average time needed to sort a random permutation of N numbers under the assumption that all $N!$ permutations are equally likely. It is not difficult to show that the average time needed to sort a random permutation is $\Omega(N)$ steps. This is because, at the end of the sorting procedure, the smallest number in the list must be stored in the leftmost cell. In a random permutation, the smallest number is equally likely to be initially contained in any of the cells $1, 2, \ldots, N$. If the smallest number begins in cell d, then at least $d-1$ steps are needed to bring it to cell 1, so the average running time for the entire algorithm is lower bounded by $\frac{1}{N}\sum_{d=1}^{N} d-1 = \frac{N-1}{2}$. In fact, the expected running time will be at least $N - O(\sqrt{N})$ since one of the $O(\sqrt{N})$ smallest items is likely to start in one of the rightmost $O(\sqrt{N})$ positions.

After understanding how the odd-even transposition sorting algorithm performs on a linear array, it is reasonable to investigate extensions of the bubble sort to two dimensional arrays. In particular, we would like to sort N numbers on a $\sqrt{N} \times \sqrt{N}$ mesh of processors. For convenience, we assume that $\sqrt{N} = 2n$ or $\sqrt{N} = 2n + 1$ for some integer n. We number the columns of the mesh $1, 2, \ldots, \sqrt{N}$ increasing from left to right and we similarly number the rows $1, 2, \ldots, \sqrt{N}$ increasing from top to bottom. Since processors now have four neighbors, there are many possibilities regarding the comparisons made at any step. We will first study two algorithms that seem to be the most "natural" extensions of the bubble sort to a two-dimensional array; for these algorithms, we will assume that $\sqrt{N} = 2n$. The goal of each of these sorting procedures is to finish with the input in row major order; i.e., the m^{th} smallest number will appear in row $\lfloor \frac{m-1}{2n} \rfloor + 1$ and column $[m - 1 \pmod{2n}] + 1$.

The first algorithm listed below begins with a row sort. i is assumed to be a non-negative integer.

1. At step $4i + 1$, each row acts as a linear array and performs an odd step of the bubble sort.

2. At step $4i + 2$, each column acts as a linear array and executes an odd step of the bubble sorting algorithm In the column sort comparisons, the smaller value is output in the top-most cell.

3. At step $4i + 3$, each row acts as a linear array and carries out an even step of the odd-even transposition sort. At the same time, the leftmost and rightmost columns execute a ***wrap-around comparison***; i.e., for $h = 1, 2, \ldots, 2n - 1$, a comparison is made between the h^{th} row of column $2n$ and the $h + 1^{\text{st}}$ row of column 1 and the smaller value is placed in the h^{th} row of column $2n$.

4. At step $4i + 4$, each column acts as a linear array and performs an even step of the bubble sort.

Why do we need the wrap-around comparisons? Suppose that we did not have them and the smallest $2n$ numbers were initially stored by the cells in column 1 Then the smallest $2n$ numbers will be forced to stay in the same column at each step and we would never get the desired ordering. The penalty of having a wrap-around comparison is that extra wires are required, but it is known that the sorting procedure above will correctly sort any set of inputs in $O(N) = O(n^2)$ steps because there is essentially an N-cell linear array embedded in the mesh of processors. In the worst case, this upper bound is met when the smallest $2n$ entries begin in the same column.

---

*This work was supported by an AT&T Bell Laboratories GRPW Fellowship.

ACM-SPAA'93-6/93/Velen,Germany.
© 1993 ACM 0-89791-599-2/93/0006/0336...$1.50

The other row major sorting algorithm that we will consider is similar to the first, except that it begins with a column sort. For any non-negative integer i, steps $2i+1$ and $2i+2$ of this algorithm are steps $2i+2$ and $2i+1$ of the first algorithm, respectively. This algorithm also has a worst case running time of $\Theta(N) = \Theta(n^2)$ steps and the worst case is attained when the smallest $\sqrt{N}$ entries of the mesh begin in the same column.

The other set of algorithms we will study finish with the input in a snakelike order. Here, at the end of the sorting procedure, the m^{th} smallest number will appear in row $R_m = \lfloor \frac{m-1}{\sqrt{N}} \rfloor + 1$ and column $\begin{cases} [m-1 \pmod{\sqrt{N}}]+1, & \text{if } R_m \text{ is odd} \\ \sqrt{N} - [m-1 \pmod{\sqrt{N}}], & \text{if } R_m \text{ is even} \end{cases}$

In order to explain these algorithms, it is necessary to define another procedure for sorting N numbers on an N-cell linear array.

Definition 1 *A* reverse bubble sort *is the same as the ordinary odd-even transposition sort except that when the contents of two cells are compared, the smaller value is stored in the* rightmost *cell.*

For the three algorithms described below, we again assume that i is a non-negative integer. The first algorithm we will investigate is listed below.

1. At step $4i+1$, each row acts as a linear array. The odd rows perform an odd step of the bubble sort and the even rows carry out an even step of the reverse bubble sort.

2. At step $4i+2$, each column acts as a linear array and executes an odd step of the bubble sort.

3. At step $4i+3$, each row acts as a linear array. The odd rows perform an even step of the bubble sort and the even rows carry out an odd step of the reverse bubble sort.

4. At step $4i+4$, each column acts as a linear array and executes an even step of the bubble sort.

The next algorithm has the same odd-numbered steps as the preceding sorting procedure and its even-numbered steps are:

1. At step $4i+2$, each column acts as a linear array. The odd columns execute an odd step of the bubble sort and the even columns carry out an even step of the odd-even transposition sort.

2. At step $4i+4$, each column acts as a linear array. The odd columns perform an even step of the bubble sort and the even columns carry out an odd step of the odd-even transposition sort.

The last algorithm that we shall examine has the same even-numbered steps are the second snakelike sorting procedure and its odd-numbered steps are

1. At step $4i+1$, each row acts as a linear array. The odd rows perform an odd step of the bubble sort and the even rows carry out an odd step of the reverse bubble sort.

2. At step $4i+3$, each row acts as a linear array. The odd rows execute an even step of the bubble sort and the even rows carry out an even step of the reverse bubble sort.

It is possible to show that the worst case running time of each of these algorithms is $\Theta(N) = \Theta(n^2)$ steps.

As with the case of bubble sorting on a linear array, it would be interesting to determine the average time needed by each algorithm to sort a permutation of N numbers, assuming that all $N!$ permutations are equally likely. If we once again lower bound the average number of steps required by each algorithm by the average number of steps needed to move the smallest number to the top, left cell of the mesh, the lower bound is $\Omega(\sqrt{N})$ steps since the diameter of the network is $2\sqrt{N}-2$. Is this bound tight? In this paper, we will show that these algorithms have an average case performance of $\Omega(N)$ steps and hence, the average-case performance is much worse than the diameter lower bound.

2 ANALYSIS OF THE ROW MAJOR ORDERING ALGORITHMS

Consider a random permutation $\mathcal{A}$ of the numbers 1 to $4n^2$ in a $2n \times 2n$ grid with wrap-around wires. A lower bound on the number of steps needed to sort the entries of $\mathcal{A}$ is the number of steps needed to sort the cells of $\mathcal{A}^{01}$, where $\mathcal{A}^{01}$ is the matrix derived from $\mathcal{A}$ by substituting zeroes for the numbers 1 to $2n^2$ and substituting ones for the remaining numbers. We will focus upon the effects of the sorting algorithms on arbitrary 0-1 matrices and then we will apply the results to $\mathcal{A}^{01}$. Since half of the entries of $\mathcal{A}^{01}$ are zeroes and half are ones, we often observe that after the first row sort and column sort are executed, the odd-numbered columns tend to have more zeroes than ones and the even-numbered columns are likely to have more ones than zeroes. We note that when the sorting procedure on $\mathcal{A}^{01}$ is finished, the first n rows of each column consist entirely of zeroes and the bottom half of the matrix contains only ones. Hence, we are interested in investigating how these algorithms will even out the number of zeroes and ones in each column.

Definition 2 *For any* $0-1$ *matrix, let* $w_k(t)$ *and* $z_k(t)$ *denote the number of ones and zeroes, respectively, in column* k *immediately after the* t^{th} *sorting step.*

Definition 3 *The* weight *of a column is the number of ones in the column.*

We have the following results.

Lemma 1 *If step* t *is a column sort, then for all* k,

$$\begin{aligned} w_k(t) &= w_k(t-1) \\ z_k(t) &= z_k(t-1) \end{aligned}$$

Proof: Column sorts make no change in the weight of any column. Their only consequence is that they tend to move the zeroes of a column toward the top and the ones of the column toward the bottom. □

Lemma 2 *If step* t *is an odd row sort, then for all* $j \in \{1, \ldots, n\}$

$$\begin{aligned} w_{2j}(t) &\geq w_{2j-1}(t-1) \\ z_{2j-1}(t) &\geq z_{2j}(t-1) \end{aligned}$$

Proof: Let A_k denote column k immediately before step t. We observe that the zeroes of the even-numbered columns "travel together" and the ones of the odd-numbered columns "travel together" in the following sense: let $B_1, B_2, \ldots, B_{2n}$ be the new columns after step t. For any column vector C, let C^h represent the element in row h of column C. Then for all $j \in \{1,2,\ldots,n\}$ and $h \in \{1,2,\ldots,2n\}$,

- $A^h_{2j} = 0$ implies $B^h_{2j-1} = 0$,
- $A^h_{2j-1} = 1$ implies $B^h_{2j} = 1$.

Hence, an odd row sort causes the zeroes of the even-numbered columns to travel to the odd-numbered columns and shifts the weight of the odd-numbered columns to the even-numbered columns. Since the number of ones in A_{2j-1} is $w_{2j-1}(t-1)$, the weight of B_{2j} is $w_{2j}(t)$, and the number of zeroes in A_{2j} and B_{2j-1} are $z_{2j}(t-1)$ and $z_{2j-1}(t)$, respectively, the lemma follows. □

Lemma 3 *If step t is an even row sort, then*

$$\begin{aligned} w_{2j+1}(t) &\geq w_{2j}(t-1),\ j \in \{1,\ldots,n-1\} \\ z_{2j}(t) &\geq z_{2j+1}(t-1),\ j \in \{1,\ldots,n-1\} \\ w_1(t) &\geq w_{2n}(t-1)-1 \\ z_{2n}(t) &\geq z_1(t-1)-1 \end{aligned}$$

Proof: Let D_i and E_k represent column i immediately before step t and column k just after step t, respectively. Then for $j \in \{1,\ldots,n-1\}$ and $h \in \{1,\ldots,2n\}$,

- $D^h_{2j+1} = 0$ implies $E^h_{2j} = 0$,
- $D^h_{2j} = 1$ implies $E^h_{2j+1} = 1$.

Since the weight of D_{2j} is $w_{2j}(t-1)$, the number of ones in E_{2j+1} is $w_{2j+1}(t)$, and the number of zeroes in D_{2j+1} and E_{2j} are $z_{2j+1}(t-1)$ and $z_{2j}(t)$, we have demonstrated the first two inequalities of Lemma 3.

Next we consider the effect of the step t on the leftmost and rightmost columns. Here, for $h \in \{1,2,\ldots,2n-1\}$,

- $D_1^{h+1} = 0$ implies $E^h_{2n} = 0$,
- $D^h_{2n} = 1$ implies $E_1^{h+1} = 1$.

As in the proof of Lemma 2, we say that an even row sort causes the zeroes of column 1 to *travel* to column $2n$ and the ones of column $2n$ to travel to column 1. If $D_1^1 = 0$ and $D^{2n}_{2n} = 1$, then E_{2n} may have one fewer zero than D_1 and E_1 may have one less one than D_{2n}. Otherwise, E_{2n} will have at least as many zeroes as D_1 and the weight of E_1 will be at least as large as the weight of D_{2n}. Hence, we have established the last two inequalities of Lemma 3. □

From Lemmas 1-3, we have the following theorem.

Theorem 1 *For any mesh containing α zeroes and $N-\alpha$ ones, if after some odd row sorting step*

- *there is an odd-numbered column containing $x > \lceil \frac{\alpha}{\sqrt{N}} \rceil$ zeroes, then at least $(x - \lceil \frac{\alpha}{\sqrt{N}} \rceil - 1) \cdot 2\sqrt{N}$ additional steps will be required to complete the sorting*
- *there is an even-numbered column with weight $y > \lceil \frac{N-\alpha}{\sqrt{N}} \rceil$, then at least $(y - \lceil \frac{N-\alpha}{\sqrt{N}} \rceil - 1)\cdot 2\sqrt{N}$ more steps will be needed to finish the sorting.*

Proof: Suppose that after some odd row sorting step t_i, column $2j+1$ contains $x > \lceil \frac{\alpha}{\sqrt{N}} \rceil$ zeroes. From Lemma 1, we know that column sorts don't affect the number of zeroes and ones in a column; therefore, the set of zeroes of interest remains in the same column during a column sort. Lemma 3 indicates that at the next row sorting step, the zeroes that had been in column $2j+1 = [2j \,(\mathrm{mod}\ 2n)] + 1$ travel to column $[2j-1\,(\mathrm{mod}\ 2n)]+1$. Lemma 2 implies that at the following row sorting step, these zeroes are shifted left to column $[2j-2\,(\mathrm{mod}\ 2n)]+1$. Using induction, we see that this set of zeroes is moved one column to the left at each row sorting step, except at the step where is it wrapped around from column 1 to column $2n$. Thus, it will take $2\sqrt{N}$ steps for the set of zeroes to return to column $2j+1$. From Lemmas 2 and 3, we also have that the number of zeroes in the set does not decrease as it is shifted left although it may decrease by one in the wrap-around stage. Hence, we have that

$$z_{2j+1}(t_i + 2\sqrt{N}) \;\geq\; z_{2j+1}(t_i) - 1 \;=\; x-1.$$

Note that if the sorting algorithm is complete at step t_f, then for all $t \geq t_f$,

$$z_{2j+1}(t) = \begin{cases} \lceil \frac{\alpha}{\sqrt{N}} \rceil, & 2j+1 \leq \alpha - \sqrt{N}\lfloor \frac{\alpha}{\sqrt{N}} \rfloor \\ \lfloor \frac{\alpha}{\sqrt{N}} \rfloor, & 2j+1 > \alpha - \sqrt{N}\lfloor \frac{\alpha}{\sqrt{N}} \rfloor \end{cases}$$

Hence, $t_f - t_i \geq (x - \lceil \frac{\alpha}{\sqrt{N}} \rceil - 1)\cdot 2\sqrt{N}$.

A similar argument applies if there is some even-numbered column at time t that has a large weight. In this case, the set of ones that originate in that column are shifted right at the row sorting steps and travels from column $2n$ to column 1 at the wrap-around stage. □

An immediate consequence of Theorem 1 is

Corollary 1 *For both algorithms, the worst-case time to sort N numbers is at least $2N - 4\sqrt{N}$.*

Proof: Consider the mesh in which one column initially consists entirely of zeroes and the remaining cells of the matrix contain ones. Here, in terms of Theorem 1, $\alpha = X = \sqrt{N}$. □

Let us return to the matrix $\mathcal{A}^{01}$. For $j \in \{1,\ldots,n\}$, let Z_{2j-1} and W_{2j} represent the number of zeroes in column $2j-1$ and the weight of column $2j$, respectively, immediately after the first row sorting step is executed, and let

$$M = \max\{\max_{i\in\{1,\ldots,n\}} Z_{2i-1}, \max_{j\in\{1,\ldots,n\}} W_{2j}\} - n - 1.$$

Then we have

Corollary 2 *The number of steps needed to sort $\mathcal{A}^{01}$, and hence $\mathcal{A}$, is greater than $4nM$. Therefore, the average number of steps required for the two dimensional bubble sort with wrap-around wires is lower bounded by $4n \cdot E[M]$, where the expectation is taken over the set of random permutations.*

Proof: Apply Theorem 1 with $\alpha = \frac{N}{2} = 2n^2$. □

In view of the fact that the worst case performance of both of the algorithms is $O(N)$ steps, to demonstrate that the algorithms require $\Theta(N)$ steps on average, it suffices to show that $E[M] = \Omega(\sqrt{N})$, regardless of whether the first set of comparisons is a row sort or a column sort.

Lemma 4 *For the algorithm that begins with a row sorting step,* $E[M] \geq \frac{n}{2} + \frac{n}{8n^2-2} - 1$.

Proof: Since $M \geq Z_1 - n - 1$, we have that $E[M] \geq E[Z_1] - n - 1$. Let A_1 be column 1 immediately after the first row sort is performed and for $h \in \{1, \ldots, 2n\}$, let $z_h = \begin{cases} 1, & A_1^h = 0 \\ 0, & A_1^h = 1 \end{cases}$. Then

$$\begin{aligned} E[Z_1] &= E[\textstyle\sum_{h=1}^{2n} z_h] \\ &= \textstyle\sum_{h=1}^{2n} E[z_h] \\ &= 2nE[z_1] \\ &= 2n \cdot \mathrm{Prob}\{z_1 = 1\} \\ &= 2n \cdot \mathrm{Prob}\{([\mathcal{A}^{01}]_{1,1}, [\mathcal{A}^{01}]_{1,2}) \neq (1,1)\}. \end{aligned}$$

If we let $\mathcal{N}_1$ and $\mathcal{N}_2$ denote the number of matrices $\mathcal{A}^{01}$ such that $([\mathcal{A}^{01}]_{1,1}, [\mathcal{A}^{01}]_{1,2}) = (1,1)$ and the number of $2n \times 2n$ matrices with $2n^2$ zeroes and $2n^2$ ones, respectively, then

$$\mathrm{Prob}\{([\mathcal{A}^{01}]_{1,1}, [\mathcal{A}^{01}]_{1,2}) = (1,1)\} = \frac{\mathcal{N}_1}{\mathcal{N}_2}.$$

To evaluate $\mathcal{N}_2$, we note that we are looking for the number of ways to select the $2n^2$ out of $4n^2$ cells that initially store zeroes since the remaining cells will automatically hold ones. Similarly, to evaluate $\mathcal{N}_1$, we keep in mind that if $([\mathcal{A}^{01}]_{1,1}, [\mathcal{A}^{01}]_{1,2}) = (1,1)$, then the remaining $4n^2 - 2$ cells of $\mathcal{A}^{01}$ contain $2n^2$ zeroes and $2n^2 - 2$ ones. Hence,

$$E[z_1] = 1 - \frac{\binom{4n^2-2}{2n^2}}{\binom{4n^2}{2n^2}} = \frac{3}{4} + \frac{1}{16n^2 - 4}$$

and the lemma follows. □

Hence, we have

Theorem 2 *The average number of steps required to sort a random permutation of N numbers by the algorithm that begins with a row sorting step is lower bounded by* $\frac{N}{2} - 2\sqrt{N}$.

For the remainder of the paper, let $\mathcal{E}[\gamma, N]$ represent the event that the average number of steps needed to sort a random permutation of N numbers is less than γN.

Theorem 3 *For the algorithm that begins with a row sorting step, given any $\gamma < \frac{1}{2}$ and $\delta > 0$, there exists N_0 such that Prob $\{\mathcal{E}[\gamma, N]\} \leq \delta$ for all $N \geq N_0$.*

For the proofs of Theorems 3, 5, and 8, we will utilize the well-known Chebyshev inequality: for any random variable X,

$$\mathrm{Prob}[|X - E[X]| \geq t] \leq \frac{\mathrm{Var}(X)}{t^2}.$$

We will need a weaker consequence of this inequality, namely that for any random variable X and any $t > 0$,

$$\mathrm{Prob}[X \leq E[X] - t] \leq \frac{\mathrm{Var}(X)}{t^2}. \qquad (1)$$

Proof: Corollary 2 indicates that it is sufficient to establish that for any $\gamma < \frac{1}{2}$,

$$\lim_{n\to\infty} \mathrm{Prob}[M \leq \gamma n] = 0.$$

Since $M \geq Z_1 - n - 1$,

$$\mathrm{Prob}[M \leq \gamma n] \leq \mathrm{Prob}[Z_1 \leq (\gamma+1)n + 1]. \qquad (2)$$

From the proof of Lemma 4 and (1), we have that for all $t > 0$,

$$\mathrm{Prob}[Z_1 \leq \frac{3}{2}n + \frac{n}{8n^2-2} - t] \leq \frac{\mathrm{Var}(Z_1)}{t^2}.$$

Substituting $t = n(\frac{1}{2} - \gamma - \frac{1}{n} + \frac{1}{8n^2-2})$ into the preceding inequality gives

$$\mathrm{Prob}[Z_1 \leq (\gamma+1)n + 1] \leq \frac{\mathrm{Var}(Z_1)}{n^2\left(\frac{1}{2} - \gamma - o(1)\right)^2}. \qquad (3)$$

Using the same notation as in the proof of Lemma 4, we have that

$$\begin{aligned} \mathrm{Var}(Z_1) &= E\left(\left(\textstyle\sum_{h=1}^{2n} z_h\right)^2\right) - (E[Z_1])^2 \\ &= E\left(\textstyle\sum_{h=1}^{2n} z_h^2 + \sum_{h\neq h'} z_h z_{h'}\right) \\ &\quad - \left(\frac{3}{2}n + \frac{n}{8n^2-2}\right)^2 \\ &= \textstyle\sum_{h=1}^{2n} E[z_h] + \sum_{h\neq h'} E[z_h z_{h'}] \\ &\quad - \left(\frac{3}{2}n + \frac{n}{8n^2-2}\right)^2, \text{ since } z_h = 0 \text{ or } 1 \\ &= 2n \cdot E[z_1] + 2n(2n-1) \cdot E[z_1 z_2] \\ &\quad - \left(\frac{3}{2}n + \frac{n}{8n^2-2}\right)^2, \text{ by symmetry.} \end{aligned}$$

We have already seen that $E[z_1] = \frac{3}{4} + \frac{1}{16n^2-4}$. We note that

$$\begin{aligned} E[z_1 z_2] &= \mathrm{Prob}\{z_1 = z_2 = 1\} \\ &= 1 - \mathrm{Prob}\{z_1 = 0 \text{ or } z_2 = 0\}. \end{aligned}$$

For $i \in \{1,2\}$, the event $z_i = 0$ is equivalent to the event $([\mathcal{A}^{01}]_{i,1}, [\mathcal{A}^{01}]_{i,2}) = (1,1)$. Hence,

$$\begin{aligned} &\mathrm{Prob}\{z_1 = z_2 = 1\} \\ &= \textstyle\sum_{i=1}^{2} \mathrm{Prob}\{([\mathcal{A}^{01}]_{i,1}, [\mathcal{A}^{01}]_{i,2}) = (1,1)\} \\ &\quad - \mathrm{Prob}\{[\mathcal{A}^{01}]_{1,1} = [\mathcal{A}^{01}]_{1,2} = [\mathcal{A}^{01}]_{2,1} = [\mathcal{A}^{01}]_{2,2} = 1\} \\ &= 2 \cdot \frac{\binom{4n^2-2}{2n^2}}{\binom{4n^2}{2n^2}} - \frac{\binom{4n^2-4}{2n^2}}{\binom{4n^2}{2n^2}} \\ &= \frac{7}{16} - \frac{n^2 - \frac{3}{8}}{32n^4 - 32n^2 + 6}. \end{aligned}$$

Hence, $E[z_1 z_2] = \frac{9}{16} + \frac{n^2 - \frac{3}{8}}{32n^4 - 32n^2 + 6}$ and so

$$\begin{aligned} \mathrm{Var}(Z_1) &= \frac{3n}{8} - \frac{64n^6 - 12n^5 - 76n^4 + 19n^3 + 21n^2 - \frac{9n}{2}}{(8n^2-2)^2(4n^2-3)} \\ &= n\left(\frac{3}{8} - o(1)\right). \end{aligned}$$

The previous equation, combined with (2) and (3), implies

$$\mathrm{Prob}[M \leq \gamma n] = \frac{\frac{3}{8} - o(1)}{n\left(\frac{1}{2} - \gamma - o(1)\right)^2},$$

completing the proof. □

Theorem 4 *The average number of steps required to sort a random permutation of N numbers by the algorithm that begins with a column sorting step is lower bounded by $\frac{3N}{8} - 2\sqrt{N}$.*

Proof: Because of Corollary 2, it is sufficient to show that for the algorithm that begins with a column sorting step,

$$E[M] \geq \frac{3}{8}n + \frac{n^3 - \frac{9}{8}n}{16n^4 - 16n^2 + 3} - 1.$$

After a column sort and a row sort have been performed, $\mathcal{A}^{01}$ has been mapped into a matrix which will be denoted by A: A can be partitioned into n^2 blocks of the form $\begin{pmatrix} a_{2h-1,2j-1} & a_{2h-1,2j} \\ a_{2h,2j-1} & a_{2h,2j} \end{pmatrix}$ with the property that none of the elements in a block have been compared with any elements of a different block. We have the following mapping of initial blocks to blocks immediately after the first row sort:

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \longmapsto \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\left\{\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\right\} \longmapsto \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\left\{\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}\right\} \longmapsto \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$$

$$\left\{\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}\right\} \longmapsto \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$$

$$\left\{\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\right\} \longmapsto \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \longmapsto \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Since $M \geq Z_1 - n - 1$, we have that $E[M] \geq E[Z_1] - n - 1$. For $h \in \{1, \ldots, n\}$, let $z_h = \begin{cases} 2, & A_1^{2h-1} = A_1^{2h} = 0 \\ 1, & A_1^{2h-1} = 0,\ A_1^{2h} = 1 \\ 0, & A_1^{2h-1} = A_1^{2h} = 1 \end{cases}$

Then

$$E[Z_1] = E[\sum_{h=1}^{n} z_h] = \sum_{h=1}^{n} E[z_1] = nE[z_1], \quad \text{by symmetry.}$$

The probability of any block $\begin{pmatrix} [\mathcal{A}^{01}]_{1,1} & [\mathcal{A}^{01}]_{1,2} \\ [\mathcal{A}^{01}]_{2,1} & [\mathcal{A}^{01}]_{2,2} \end{pmatrix}$ with z zeroes and $4 - z$ ones is $\dfrac{\binom{4n^2-4}{2n^2-z}}{\binom{4n^2}{2n^2}}$.

These probabilities are:

- $z = 0$ or 4 : $\frac{1}{16} - \frac{3n^2 - \frac{21}{8}}{32n^4 - 32n^2 + 6}$
- $z = 1$ or 3 : $\frac{1}{16} - \frac{\frac{3}{16}}{16n^4 - 16n^2 + 3}$
- $z = 2$: $\frac{1}{16} + \frac{n^2 - \frac{3}{8}}{32n^4 - 32n^2 + 6}$

Hence,

$$\begin{aligned} \text{Prob}\{z_1 = 2\} &= \left(\frac{1}{16} - \frac{3n^2 - \frac{21}{8}}{32n^4 - 32n^2 + 6}\right) \\ &\quad + 4 \cdot \left(\frac{1}{16} - \frac{\frac{3}{16}}{16n^4 - 16n^2 + 3}\right) \\ &\quad + 2 \cdot \left(\frac{1}{16} + \frac{n^2 - \frac{3}{8}}{32n^4 - 32n^2 + 6}\right) \\ &= \frac{7}{16} - \frac{n^2 - \frac{3}{8}}{32n^4 - 32n^2 + 6} \\ \text{Prob}\{z_1 = 1\} &= 4 \cdot \left(\frac{1}{16} + \frac{n^2 - \frac{3}{8}}{32n^4 - 32n^2 + 6}\right) \\ &\quad + 4 \cdot \left(\frac{1}{16} - \frac{\frac{3}{16}}{16n^4 - 16n^2 + 3}\right) \\ &= \frac{1}{2} + \frac{1}{8n^2 - 2} \\ \text{Prob}\{z_1 = 0\} &= \frac{1}{16} - \frac{3n^2 - \frac{21}{8}}{32n^4 - 32n^2 + 6} \end{aligned}$$

and thus,

$$E[z_1] = \frac{11}{8} + \frac{n^2 - \frac{9}{8}}{16n^4 - 16n^2 + 3}.$$

Hence,

$$\begin{aligned} E[M] &\geq E[Z_1] \\ &= n\left(\frac{11}{8} + \frac{n^2 - \frac{9}{8}}{16n^4 - 16n^2 + 3}\right) - n - 1 \\ &= \frac{3}{8}n + \frac{n^3 - \frac{9}{8}n}{16n^4 - 16n^2 + 3} - 1. \ \square \end{aligned}$$

Theorem 5 *For the algorithm that begins with a column sorting step, given any $\gamma < \frac{3}{8}$ and $\delta > 0$, there exists N_1 such that Prob $\{\mathcal{E}[\gamma, N]\} \leq \delta$ for all $N \geq N_1$.*

Proof: As in the proof of Theorem 3, it is sufficient to show that for all $\gamma < \frac{3}{8}$,

$$\lim_{n \to \infty} \text{Prob}[Z_1 \leq (\gamma + 1)n + 1] = 0. \qquad (4)$$

From the proof of Theorem 4 and (1), we have that for all $t > 0$,

$$\text{Prob}[Z_1 \leq \frac{11}{8}n + \frac{n^3 - \frac{9}{8}n}{16n^4 - 16n^2 + 3} - t] \leq \frac{\text{Var}(Z_1)}{t^2}.$$

Substituting $t = n(\frac{3}{8} - \gamma - \frac{1}{n} + \frac{n^2 - \frac{9}{8}}{16n^4 - 16n^2 + 3})$ into the preceding inequality gives

$$\text{Prob}[Z_1 \leq (\gamma + 1)n + 1] \leq \frac{\text{Var}(Z_1)}{n^2\left(\frac{3}{8} - \gamma - o(1)\right)^2}. \qquad (5)$$

Using the same notation as in the proof of Theorem 4, we have that

$$\text{Var}(Z_1) = E\left(\left(\sum_{h=1}^{n} z_h\right)^2\right) - (E[Z_1])^2$$

$$
\begin{aligned}
&= \textstyle\sum_{h=1}^{n} E[z_h^2] + \sum_{h\neq h'} E[z_h z_{h'}] \\
&\quad - \left(\frac{11}{8}n + \frac{n^3 - \frac{9}{8}n}{16n^4 - 16n^2 + 3}\right)^2 \\
&= n \cdot E[z_1^2] + n(n-1)\cdot E[z_1 z_2] \\
&\quad - \left(\frac{11}{8}n + \frac{n^3 - \frac{9}{8}n}{16n^4 - 16n^2 + 3}\right)^2,
\end{aligned}
$$

$$
\begin{aligned}
\text{where } E[z_1^2] &= 4\cdot\left(\frac{7}{16} - \frac{n^2 - \frac{3}{8}}{32n^4 - 32n^2 + 6}\right) \\
&\quad + 1\cdot\left(\frac{1}{2} + \frac{1}{8n^2 - 2}\right) \\
&\quad + 0\cdot\left(\frac{1}{16} - \frac{3n^2 - \frac{21}{8}}{32n^4 - 32n^2 + 6}\right) \\
&= \frac{9}{4} - \frac{3}{64n^4 - 64n^2 + 12} \\
\text{and } E[z_1 z_2] &= \text{Prob}\{z_1 = z_2 = 1\} \\
&\quad + 2\cdot\text{Prob}\{z_1 = 1,\ z_2 = 2\} \\
&\quad + 2\cdot\text{Prob}\{z_1 = 2,\ z_2 = 1\} \\
&\quad + 4\cdot\text{Prob}\{z_1 = z_2 = 2\}.
\end{aligned}
$$

The event $z_1 = z_2 = 1$ means that $\begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ A_{3,1} & A_{3,2} \\ A_{4,1} & A_{4,2} \end{pmatrix}$ is either $\begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}$, $\begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$, $\begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}$, or $\begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$. There are $16\cdot\binom{4n^2-8}{2n^2-4} + 16\cdot\binom{4n^2-8}{2n^2-3} + 16\cdot\binom{4n^2-8}{2n^2-3} + 16\cdot\binom{4n^2-8}{2n^2-2}$ many matrices $\mathcal{A}^{01}$ satisfying $z_1 = z_2 = 1$. Hence,

$$\text{Prob}\{z_1 = z_2 = 1\} = \frac{1}{4} + \frac{4n^4 - 11n^2 + \frac{15}{4}}{64n^6 - 144n^4 + 92n^2 - 15}$$

The event $z_1 = 1, z_2 = 2$ implies $\begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ A_{3,1} & A_{3,2} \\ A_{4,1} & A_{4,2} \end{pmatrix}$ is either $\begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$, or $\begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$. Hence, it is straightforward to show that

$$
\begin{aligned}
&\text{Prob}\{z_1 = 1,\ z_2 = 2\} = \text{Prob}\{z_1 = 2,\ z_2 = 1\} \\
&\qquad = \frac{7}{32} + \frac{\frac{17}{2}n^4 - \frac{97}{8}n^2 + \frac{15}{32}}{64n^6 - 144n^4 + 92n^2 - 15}.
\end{aligned}
$$

Similarly, $z_1 = z_2 = 2$ if and only if $\begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ A_{3,1} & A_{3,2} \\ A_{4,1} & A_{4,2} \end{pmatrix}$ is $\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$, or $\begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$. Hence,

$$
\begin{aligned}
&\text{Prob}\{z_1 = z_2 = 2\} \\
&= \frac{49}{256} - \frac{43n^6 - \frac{1227}{8}n^4 + 170n^2 - \frac{15015}{256}}{256n^8 - 1024n^6 + 1376n^4 - 704n^2 + 105}.
\end{aligned}
$$

Therefore,

$$E[z_1 z_2] = \frac{121}{64} - \frac{20n^6 - \frac{219}{2}n^4 + 241n^2 - \frac{12495}{64}}{256n^8 - 1024n^6 + 1376n^4 - 704n^2 + 105}.$$

and hence,

$$\text{Var}(Z_1) = n\left(\frac{23}{64} - o(1)\right).$$

The previous equation, combined with (4) and (5) implies

$$\text{Prob}[Z_1 \leq (\gamma+1)n + 1] \leq \frac{\frac{23}{64} - o(1)}{n\left(\frac{3}{8} - \gamma - o(1)\right)^2},$$

proving Theorem 5. □

3 ANALYSIS OF THE SNAKELIKE SORTING ALGORITHMS

We assume that $\sqrt{N} = 2n$ in this section. The analysis for $\sqrt{N} = 2n + 1$ is similar and outlined in the appendix. $\mathcal{A}$ and $\mathcal{A}^{01}$ are defined as before. We begin by considering the first snakelike sorting procedure. For any $0-1$ matrix, we utilize the following definitions for this algorithm:

Definition 4 *Let $Z_1(i)$ denote the number of zeroes in the odd-numbered columns and the even-numbered rows of column $2n$ immediately after step $4i+1$.*

Definition 5 *Let $Z_2(i)$ denote the number of zeroes in the odd-numbered columns and the odd-numbered rows of column $2n$ just after step $4i+2$.*

Definition 6 *Let $Z_3(i)$ denote the number of zeroes in the even-numbered columns and the odd-numbered rows of column 1 right after step $4i+3$.*

Definition 7 *Let $Z_4(i)$ denote the number of zeroes in the even-numbered columns and the even-numbered rows of column 1 immediately after step $4i+4$.*

We have the following relationships among $Z_1(i)$, $Z_2(i)$, $Z_3(i)$ and $Z_4(i)$:

Lemma 5 $Z_2(i) \geq Z_1(i)$.

Proof: Let A_{2n}, B_{2n} represent column $2n$ immediately before and after step $4i+2$ is executed, respectively. Then for $h \in \{1, 2, \ldots, n\}$, $A_{2n}^{2h} = 0$ implies $B_{2n}^{2h-1} = 0$. Hence, the number of zeroes in the odd-numbered rows of column $2n$ just after step $4i+2$ is no less than the number of zeroes in the even-numbered rows of column $2n$ immediately after step $4i+1$. To complete the proof, we observe that a column sort will have no effect on the number of zeroes in the odd-numbered columns. □

Lemma 6 $Z_3(i) \geq Z_2(i)$.

Proof: Let C_k and D_l represent column k immediately before step $4i+3$ and column l just after step $4i+3$, respectively. For $h \in \{1, \ldots, n\}$, $C_1^{2h-1} = D_1^{2h-1}$ and $C_{2n}^{2h-1} = D_{2n}^{2h-1}$ because the contents of cells of the form $(2h-1, 1)$ and $(2h-1, 2n)$ are not compared with the contents of other cells during step $4i+3$. Therefore, the number of zeroes in the odd-numbered rows of columns 1 and $2n$ is the same before and after step $4i+3$. We observe that for $h \in \{1, \ldots, n\}$,

- $C_{2j+1}^{2h-1} = 0$ implies $D_{2j}^{2h-1} = 0$, $j \in \{1, \ldots, n-1\}$
- $C_{2j-1}^{2h} = 0$ implies $D_{2j}^{2h} = 0$, $j \in \{1, \ldots, n\}$

Hence, the number of zeroes in columns $2, 4, \ldots, 2n-2$ and the even-numbered rows of column $2n$ immediately after step $4i+3$ is greater than or equal to the number of zeroes in columns $3, 5, \ldots, 2n-1$ and the even-numbered rows of column 1 just before step $4i+3$ was performed, finishing the proof. □

Lemma 7 $Z_4(i) \geq Z_3(i) - 1$.

Lemma 8 $Z_1(i+1) \geq Z_4(i)$.

We omit the proofs of these lemmas because they are like the proofs of Lemmas 5 and 6.

Let $f(\alpha, N) = \lceil \frac{\alpha}{2} + \frac{\alpha}{2\sqrt{N}} \rceil$. From Lemmas 5-8, we have the following theorem.

Theorem 6 *For any mesh containing α zeroes and $N - \alpha$ ones, if after the first step the number of zeroes in the odd-numbered columns plus the number of zeroes in the even-numbered rows of column $\sqrt{N}$ is $x > f(\alpha, N)$, then at least $4(x - f(\alpha, N) - 1)$ additional steps will be required to complete the sorting.*

Proof: The preceding four lemmas imply that for all non-negative integers i,

$$Z_1(i+1) \geq Z_1(i) - 1.$$

When the sorting algorithm is complete, each column will either have $\lceil \frac{\alpha}{\sqrt{N}} \rceil$ or $\lfloor \frac{\alpha}{\sqrt{N}} \rfloor$ zeroes and so the number of zeroes in the odd-numbered columns and the even-numbered rows of column $2n$ is at most $f(\alpha, N) = \lceil \frac{\alpha}{2} + \frac{\alpha}{2\sqrt{N}} \rceil$. Hence, there is some minimal i_f such that for all $i \geq i_f$,

$$Z_1(i) \leq f(\alpha, N)$$

and the theorem follows. □

If we apply Theorem 6 to $\mathcal{A}^{01}$, we have the following result.

Corollary 3 *The average number of steps required for this algorithm is lower bounded by $4\left(E[Z_1(0)] - f(\frac{N}{2}, N) - 1\right)$, where the expectation is taken over the set of random permutations.*

Since the worst case performance of this algorithm is O(N) steps, to demonstrate that it requires $\Theta(N)$ steps on average, it is enough to show that $E[Z_1(0)] - \frac{N}{4} = \Omega(N)$.

Lemma 9 $E[Z_1(0)] = \frac{3}{8}N + \frac{\sqrt{N}}{8} + \frac{\sqrt{N}}{8(\sqrt{N}+1)}$.

Proof: Let A be the matrix immediately after the first row sort is performed and for $h, j \in \{1, \ldots, \sqrt{N}\}$, let $z_{h,j} = \begin{cases} 1, & A_j^h = 0 \\ 0, & A_j^h = 1 \end{cases}$. Then

$$\begin{aligned} E[Z_1(0)] &= E\left(\sum_{h=1}^{\sqrt{N}} \sum_{j=1}^{\frac{\sqrt{N}}{2}} z_{h,2j-1} + \sum_{h=1}^{\frac{\sqrt{N}}{2}} z_{2h,\sqrt{N}}\right) \\ &= \left(\sum_{h=1}^{\frac{\sqrt{N}}{2}} E[z_{2h-1,1}] + \sum_{h=1}^{\sqrt{N}} \sum_{j=2}^{\frac{\sqrt{N}}{2}} E[z_{h,2j-1}]\right) \\ &\quad + \left(\sum_{h=1}^{\frac{\sqrt{N}}{2}} E[z_{2h,1}] + \sum_{h=1}^{\frac{\sqrt{N}}{2}} E[z_{2h,\sqrt{N}}]\right) \\ &= \left(\frac{N}{2} - \frac{\sqrt{N}}{2}\right) E[z_{1,1}] + \sqrt{N} E[z_{2,1}], \end{aligned}$$

by symmetry.

Since $A_{2,1} = [\mathcal{A}^{01}]_{2,1}$, $E[z_{2,1}] = \frac{1}{2}$. As in the proof of Lemma 4,

$$E[z_{1,1}] = \text{Prob}\{([\mathcal{A}^{01}]_{1,1}, [\mathcal{A}^{01}]_{1,2}) \neq (1,1)\} = \frac{3}{4} + \frac{1}{4N-4}$$

and the lemma follows. □

Hence, we have

Theorem 7 *The average number of steps required to sort a random permutation of N numbers by the first two dimensional snakelike bubble sorting algorithm is lower bounded by $\frac{N}{2} - \frac{\sqrt{N}}{2} - 7$.*

We have the following stronger result.

Theorem 8 *For the first snakelike sorting algorithm, given any $\gamma < \frac{1}{2}$, and $\delta > 0$, there exists $\dot{N}$ such that Prob $\{\mathcal{E}[\gamma, N]\} \leq \delta$ for all $N \geq \dot{N}$.*

Proof: Theorem 6 indicates that it is sufficient to establish that for any $\gamma < \frac{1}{2}$,

$$\begin{aligned} &\lim_{n\to\infty} \text{Prob}\left\{4\left(Z_1(0) - n^2 - \frac{n}{2} - 1\right) \leq \gamma \cdot 4n^2\right\} \\ &\quad = \lim_{n\to\infty} \text{Prob}\left\{Z_1(0) \leq n^2(\gamma+1) + \frac{n}{2} + 1\right\} = 0. \quad (6) \end{aligned}$$

Lemma 9 and (1) imply that for all $t > 0$,

$$\text{Prob}\left\{Z_1(0) \leq \frac{3}{2}n^2 + \frac{n}{4} + \frac{n}{8n+4} - t\right\} \leq \frac{\text{Var}[Z_1(0)]}{t^2}$$

Setting $t = n^2\left(\frac{1}{2} - \gamma - \frac{1}{4n} - \frac{7n+4}{8n^3+4n^2}\right)$ into the preceding inequality gives

$$\text{Prob}\left\{Z_1(0) \leq n^2(\gamma+1) + \frac{n}{2} + 1\right\} \leq \frac{\text{Var}[Z_1(0)]}{n^4\left(\frac{1}{2} - \gamma - o(1)\right)^2}. \quad (7)$$

If we maintain the notation from the proof of Lemma 9 and let

$$\begin{aligned}\mathcal{Z}_1 &= \textstyle\sum_{h=1}^{n} z_{2h-1,1} + \sum_{h=1}^{2n}\sum_{j=2}^{n} z_{h,2j-1}\\ \mathcal{Z}_2 &= \textstyle\sum_{h=1}^{n} z_{2h,1} + \sum_{h=1}^{n} z_{2h,2n},\end{aligned}$$

we find that

$$\begin{aligned}\mathrm{Var}[Z_1(0)] &= E\left(\left(\textstyle\sum_{h=1}^{2n}\sum_{j=1}^{n} z_{h,2j-1} + \sum_{h=1}^{n} z_{2h,2n}\right)^2\right)\\ &\quad - (E[Z_1(0)])^2\\ &= E\left((\mathcal{Z}_1+\mathcal{Z}_2)^2\right) - \left(\frac{3}{2}n^2 + \frac{n}{4} + \frac{n}{8n+4}\right)^2\\ &= E\left(\mathcal{Z}_1{}^2\right) + 2E(\mathcal{Z}_1\mathcal{Z}_2) + E\left(\mathcal{Z}_2{}^2\right)\\ &\quad - \left(\frac{3}{2}n^2 + \frac{n}{4} + \frac{n}{8n+4}\right)^2.\end{aligned}$$

We have that

$$\begin{aligned}E\left(\mathcal{Z}_1{}^2\right) &= (2n^2-n)E[z_{1,1}^2]\\ &\quad + [(2n^2-n)^2 - (2n^2-n)]\cdot E[z_{1,1}z_{1,3}].\end{aligned}$$

In the proof of Lemma 9, we saw that

$$E[z_{1,1}] = E[z_{1,1}^2] = \frac{3}{4} + \frac{1}{16n^2-4}.$$

With an argument exactly like the one used in the proof of Theorem 3, it is straightforward to show that

$$E[z_{1,1}z_{1,3}] = \mathrm{Prob}\{z_{1,1}=z_{1,3}=1\} = \frac{9}{16} + \frac{n^2-\frac{3}{8}}{32n^4-32n^2+6}.$$

Hence,

$$E\left(\mathcal{Z}_1{}^2\right) = \frac{9}{4}n^4 - \frac{9}{4}n^3 + \frac{17}{16}n^2 - \frac{5}{16}n + \frac{n}{8n+4} + \frac{3}{8}\cdot\frac{n^2-n}{8n^2-6}.$$

We also find that

$$\begin{aligned}2E(\mathcal{Z}_1\mathcal{Z}_2) &= 2(2n^2-n)\cdot 2nE[z_{1,1}z_{2,1}]\\ &= (8n^3-4n^2)\mathrm{Prob}\{z_{1,1}=z_{2,1}=1\}\\ &= (8n^3-4n^2)[1-\mathrm{Prob}\{z_{1,1}=0 \text{ or } z_{2,1}=0\}].\end{aligned}$$

We know that

$$\begin{aligned}\mathrm{Prob}\{z_{1,1}=0 \text{ or } z_{2,1}=0\} &= \textstyle\sum_{i=1}^{2}\mathrm{Prob}\{z_{i,1}=0\}\\ &\quad - \mathrm{Prob}\{z_{1,1}=z_{2,1}=0\}.\end{aligned}$$

We have seen in the proofs of Lemmas 4 and 9 that $\mathrm{Prob}\{z_{1,1}=0\} = \frac{1}{4} - \frac{1}{16n^2-4}$ and $\mathrm{Prob}\{z_{2,1}=0\} = \frac{1}{2}$.

$$\begin{aligned}&\mathrm{Prob}\{z_{1,1}=z_{2,1}=0\}\\ &\quad= \mathrm{Prob}\{[\mathcal{A}^{01}]_{1,1} = [\mathcal{A}^{01}]_{1,2} = [\mathcal{A}^{01}]_{2,1} = 1\}\\ &\quad= \frac{\binom{4n^2-3}{2n^2}}{\binom{4n^2}{2n^2}} = \frac{1}{8} - \frac{3}{32n^2-8}.\end{aligned}$$

Hence,

$$2E(\mathcal{Z}_1\mathcal{Z}_2) = 3n^3 - \frac{3}{2}n^2 + \frac{n}{4} - \frac{n}{8n+4}.$$

Finally,

$$\begin{aligned}E\left(\mathcal{Z}_2{}^2\right) &= 2nE[z_{2,1}^2] + (4n^2-2n)E[z_{2,1}z_{4,1}]\\ &= 2n\cdot\frac{1}{2} + (4n^2-2n)\cdot\left(\frac{3}{4} + \frac{1}{16n^2-4}\right)\\ &= 3n^2 - \frac{n}{2} + \frac{n}{4n+2}.\end{aligned}$$

Therefore,

$$\begin{aligned}\mathrm{Var}[Z_1(0)] &= \frac{17}{8}n^2 - \frac{7}{16}n + \frac{11n^2+6n}{(8n+4)^2} + \frac{3}{8}\cdot\frac{n^2-n}{8n^2-6}\\ &= n^2\left(\frac{17}{8} + o(1)\right).\end{aligned}$$

The previous expression, combined with (6) and (7) implies

$$\mathrm{Prob}\left\{4\left(Z_1(0) - n^2 - \frac{n}{2} - 1\right) \le \gamma\cdot 4n^2\right\} \le \frac{\frac{17}{8}+o(1)}{n^2(\frac{1}{2}-\gamma-o(1))^2},$$

completing the proof. □

The analysis of the second snakelike sorting procedure is nearly replicates the preceding analysis, so we will merely outline the results here.

Definition 8 *Let $Y_1(i)$ denote the number of zeroes in the odd-numbered columns immediately after step $4i+1$, or equivalently, just after step $4i+2$.*

Definition 9 *Let $Y_2(i)$ denote the number of zeroes in columns $2,4,\ldots,2n-2$, the odd-numbered rows of column 1, and the even-numbered rows of column $2n$ just after step $4i+3$.*

Definition 10 *Let $Y_3(i)$ denote the number of zeroes in columns $2,4,\ldots,2n-2$, the even-numbered rows of column 1, and the odd-numbered rows of column $2n$ just after step $4i+4$.*

We have the following results.

Lemma 10
- $Y_2(i) \ge Y_1(i)$.
- $Y_3(i) \ge Y_2(i) - 1$.
- $Y_1(i+1) \ge Y_3(i)$.

Theorem 9 *For any mesh containing α zeroes and $N-\alpha$ ones, if after the first step the number of zeroes in the odd-numbered columns is $x > \lceil\frac{\alpha}{2}\rceil$, then at least $4\left(x - \lceil\frac{\alpha}{2}\rceil - 1\right)$ additional steps will be required to complete the sorting. Therefore, the average number of steps required for the second snakelike bubble sorting algorithm is lower bounded by $4\left(E[Y_1(0)] - \frac{N}{4} - 1\right)$, where the expectation is taken over all $\sqrt{N}\times\sqrt{N}$ 0−1 matrices with $\frac{N}{2}$ zeroes.*

Lemma 11 $E[Y_1(0)] = \frac{3}{8}N - \frac{\sqrt{N}}{8} + \frac{\sqrt{N}}{8(\sqrt{N}+1)}$.

Theorem 10 *The average number of steps required to sort a random permutation of N numbers by the second two dimensional snakelike bubble sorting algorithm is lower bounded by $\frac{N}{2} - \frac{\sqrt{N}}{2} - 4$.*

Theorem 11 *For the second snakelike sorting algorithm, given any $\gamma < \frac{1}{2}$ and $\delta > 0$, there exists N^* such that Prob $\{\mathcal{E}[\gamma,N]\} \le \delta$ for all $N \ge N^*$.*

Although we have not done so here, it is not difficult to show that the sorting procedures we have investigated until this point all satisfy the property that the average time for the smallest element to move to the top, left cell is $\Theta(\sqrt{N})$ steps. To conclude this paper, we will establish that the last snakelike sorting procedure needs $\Theta(N)$ steps with "high probability" for the smallest element to move to the top, left cell; hence, this algorithm also takes $\Theta(N)$ steps with "high probability" to sort a random permutation of N numbers.

Definition 11 *Let $(j(i), k(i))$ denote the cell containing the smallest element of the mesh immediately after step $2i$. $(j(0), k(0))$ denotes the cell initially storing the smallest entry.*

Lemma 12 *If $(j(2i), k(2i))$ is the cell occupied by the m^{th} smallest element at the end of the sorting procedure, then $(j(2i+1), k(2i+1))$ is the cell occupied by either the m^{th} or $m-1^{st}$ smallest element when the sort is complete.*

Proof: There are three cases to consider:
Case 1: $j(2i) \equiv k(2i) \pmod 2$. Neither step $4i+1$ nor step $4i+2$ moves the smallest entry of the mesh. Hence,

$$(j(2i+1), k(2i+1)) = (j(2i), k(2i)),$$

the location of the m^{th} smallest element at the end of the sorting procedure.
Case 2: $j(2i) \equiv 0 \pmod 2$, $k(2i) \equiv 1 \pmod 2$. Step $4i+1$ moves the smallest element to $(j(2i), k(2i)+1)$, and step $4i+2$ leaves it there. Hence,

$$(j(2i+1), k(2i+1)) = (j(2i), k(2i)+1),$$

the location of the $m-1^{\text{st}}$ smallest element when the sort is complete.
Case 3: $j(2i) \equiv 1 \pmod 2$, $k(2i) \equiv 0 \pmod 2$. Step $4i+1$ shifts the smallest element to $(j(2i), k(2i)-1)$, and step $4i+2$ doesn't move it. Hence,

$$(j(2i+1), k(2i+1)) = (j(2i), k(2i)-1),$$

the location of the $m-1^{\text{st}}$ smallest element at the end of the sorting procedure. □

Lemma 13 *If $(j(2i+1), k(2i+1))$ is the cell occupied by the m^{th} smallest element at the end of the sorting procedure, then $(j(2i+2), k(2i+2))$ is the cell occupied by the $m-1^{st}$ smallest element when the sort is complete.*

Proof: From the preceding lemma, we see that there are only two cases to consider:
Case 1: $j(2i+1) \equiv k(2i+1) \equiv 0 \pmod 2$.
Subcase 1a: $k(2i+1) \neq 2n$. Here, step $4i+3$ moves the smallest element to $(j(2i+1), k(2i+1)+1)$ and step $4i+4$ leaves it there. Hence,

$$(j(2i+2), k(2i+2)) = (j(2i+1), k(2i+1)+1).$$

Subcase 1b: $k(2i+1) = 2n$. Here, step $4i+3$ does not shift the smallest element, but step $4i+4$ causes it to travel to

$$(j(2i+1)-1, 2n) = (j(2i+2), k(2i+2)).$$

Case 2: $j(2i+1) \equiv k(2i+1) \equiv 1 \pmod 2$.
Subcase 2a: $k(2i+1) \neq 1$. Here, step $4i+3$ moves the smallest element to $(j(2i+1), k(2i+1)-1)$ and step $4i+4$ leaves it there. Hence,

$$(j(2i+2), k(2i+2)) = (j(2i+1), k(2i+1)-1).$$

Subcase 2b: $k(2i+1) = 1$. Here, step $4i+3$ does not shift the smallest element, but step $4i+4$ causes it to travel to

$$(j(2i+1)-1, 1) = (j(2i+2), k(2i+2)).$$

In each case, cell $(j(2i+2), k(2i+2))$ holds the $m-1^{\text{st}}$ smallest entry when the sort is complete. □

Hence, we have

Theorem 12 *The third snakelike sorting procedure requires $\Theta(N)$ steps with "high probability."*

Proof: Lemmas 12 and 13 imply that if the cell initially storing the smallest entry is the cell occupied by the m^{th} smallest entry at the end of the sorting procedure, then at least $2m-3$ steps are required to bring it to the top, left cell. Since the smallest entry is equally likely to be initially contained in any of the cells of the mesh, for any $\delta > 0$, the probability that the algorithm needs fewers than δN steps is upper bounded by $\frac{\delta}{2} + \frac{5}{2N}$. □

4 CONCLUSION

We have investigated five generalizations of the odd-even transposition sort to a $\sqrt{N} \times \sqrt{N}$ mesh of processors. The first two sorting procedures finish with the input in row major order and require wrap-around wires between the leftmost and rightmost columns. We demonstrated a lower bound on the number of steps required to complete the sort on any $0-1$ matrix based on the number of zeroes and ones in each column at the end of the first row sorting step. We then used this bound to establish a $\Theta(N)$ bound on the worst case and average case performance of these algorithms. We concluded our investigation of these algorithms by proving that with "high probability," each requires $\Theta(N)$ steps to sort a random permutation of N numbers.

The other three algorithms we considered finished with the input in a snakelike order. For the first two of these three algorithms, we studied how the zeroes and ones "travel" in certain well-defined patterns. With this information, we were able to establish a $\Theta(N)$ bound on the average case performance of these algorithms and proved that with "high probability," each needs $\Theta(N)$ steps to sort a random permutation of N numbers. Our approach to demonstrating a $\Theta(N)$ step bound on the time required, with "high probability," by the third snakelike sorting procedure was to examine the path taken by the smallest entry as it moves to the top, left cell of the mesh.

Acknowledgment

I am grateful to Dr. F. T. Leighton for his guidance and encouragement throughout the course of this work.

References

[1] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes.* Morgan Kaufmann, San Mateo, California, 1992.

[2] W. Feller, *An Introduction to Probability Theory and its Applications*, Vol. 1, Wiley, New York, 1950 (3rd ed., 1968).

APPENDIX

We begin by outlining our approach to the analysis of the first two snakelike sorting procedures when $\sqrt{N} = 2n+1$. $\mathcal{A}$ is defined as before; we redefine $\mathcal{A}^{01}$ as the matrix derived from $\mathcal{A}$ by substituting zeroes for the smallest $2n^2+2n+1$ entries of the mesh and substituting ones for the remaining entries.

For the first snakelike sorting algorithm, we redefine $Z_1(i)$ and $Z_2(i)$; we maintain the earlier definitions of $Z_3(i)$ and $Z_4(i)$.

Definition 12 *Let $Z_1(i)$ denote the number of zeroes in columns $1,3,5,\ldots,2n-1$, and in the even-numbered rows of column $2n+1$ immediately after step $4i+1$.*

Definition 13 *Let $Z_2(i)$ denote the number of zeroes in columns $1,3,5,\ldots,2n-1$, and in the odd-numbered rows of column $2n+1$ just after step $4i+2$.*

With these $Z_j(i)$, $j \in \{1,2,3,4\}$, Lemmas 5-8 again apply with similar proofs. The analog of Theorem 6, Corollary 3, and Lemma 9 are listed below; we omit most of the proofs because of their great resemblance to earlier ones.

Theorem 13 *For any mesh containing α zeroes and $N-\alpha$ ones, if after the first step the number of zeroes in columns $1,3,5,\ldots,2n-1$, and the even-numbered rows of column $2n+1$ is $x > \lceil\frac{\alpha(N-1)}{2N}\rceil$, then at least $4\left(x - \lceil\frac{\alpha(N-1)}{2N}\rceil - 1\right)$ additional steps will be required to complete the sorting.*

Corollary 4 *The average number of steps required for this algorithm is lower bounded by $4\left(E[Z_1(0)] - \lceil\frac{N^2-1}{4N}\rceil - 1\right)$, where the expectation is taken over the set of random permutations.*

Lemma 14 $E[Z_1(0)] = \frac{3}{8}N - \frac{\sqrt{N}}{8} + \frac{N-\sqrt{N}-2}{8N}$.

Proof: Using the same notation as in the proof of Lemma 9, we have that

$$\begin{aligned}
E[Z_1(0)] &= E\left(\sum_{h=1}^{\sqrt{N}}\sum_{j=1}^{\frac{\sqrt{N}-1}{2}} z_{h,2j-1} + \sum_{h=1}^{\frac{\sqrt{N}-1}{2}} z_{2h,\sqrt{N}}\right) \\
&= \sum_{h=1}^{\frac{\sqrt{N}+1}{2}} E[z_{2h-1,1}] + \sum_{h=1}^{\sqrt{N}}\sum_{j=2}^{\frac{\sqrt{N}-1}{2}} E[z_{h,2j-1}] \\
&\quad + \sum_{h=1}^{\frac{\sqrt{N}-1}{2}} E[z_{2h,\sqrt{N}}] + \sum_{h=1}^{\frac{\sqrt{N}-1}{2}} E[z_{2h,1}] \\
&= \left(\frac{N}{2} - \frac{\sqrt{N}}{2}\right)E[z_{1,1}] + \frac{\sqrt{N}-1}{2}E[z_{2,1}],
\end{aligned}$$

by symmetry.

Since $A_{2,1} = [\mathcal{A}^{01}]_{2,1}$, $E[z_{2,1}] = \frac{N+1}{2N}$.

$$\begin{aligned}
E[z_{1,1}] &= \text{Prob}\{([\mathcal{A}^{01}]_{1,1},[\mathcal{A}^{01}]_{1,2}) \neq (1,1)\} \\
&= 1 - \frac{\begin{pmatrix} 4n^2+4n-1 \\ 2n^2+2n+1 \end{pmatrix}}{\begin{pmatrix} 4n^2+4n+1 \\ 2n^2+2n+1 \end{pmatrix}} = \frac{3}{4} + \frac{3}{4N}
\end{aligned}$$

and the lemma follows. □

Theorems 7 and 8 still apply and have proofs like the ones given earlier.

For the second snakelike sorting algorithm, it turns out that the preceding analysis for the first snakelike sorting algorithm is applicable here; i.e., we can use the same definitions and theorems with some minor variations in the proofs.

We conclude the appendix by studying the last snakelike sorting procedure in the case where $\sqrt{N} = 2n+1$. We maintain the earlier notation $(j(i), k(i))$ for non-negative integers i. We omit the proof of Lemma 16; it is very similar to the proof of Lemma 13.

Lemma 15 *If $(j(2i), k(2i))$ is the cell occupied by the m^{th} smallest entry at the end of the sorting procedure, then $(j(2i+1), k(2i+1))$ is the cell occupied by either the m^{th} or $m-1^{st}$ smallest entry when the sort is complete.*

Proof: There are three cases to consider:
Case 1: $j(2i) \equiv k(2i) \pmod 2$.
Neither step $4i+1$ nor step $4i+2$ moves the smallest entry of the mesh. Hence,

$$(j(2i+1), k(2i+1)) = (j(2i), k(2i)),$$

the location of the m^{th} smallest element at the end of the sorting procedure.
Case 2: $j(2i) \equiv 0 \pmod 2$, $k(2i) \equiv 1 \pmod 2$.
Subcase 2a: $k(2i) \neq \sqrt{N}$. Here, step $4i+1$ moves the smallest element to $(j(2i), k(2i)+1)$, and step $4i+2$ leaves it there. Hence,

$$(j(2i+1), k(2i+1)) = (j(2i), k(2i)+1),$$

the location of the $m-1^{st}$ smallest element when the sort is complete.
Subcase 2b: $k(2i) = \sqrt{N}$. Here, step $4i+1$ does not shift the smallest element, but step $4i+2$ causes it to travel to

$$(j(2i+1), k(2i+1)) = (j(2i+1)-1, 2n),$$

the location of the $m-1^{st}$ smallest element at the end of the sorting procedure.
Case 3: $j(2i) \equiv 1 \pmod 2$, $k(2i) \equiv 0 \pmod 2$.
Step $4i+1$ shifts the smallest element to $(j(2i), k(2i)-1)$, and step $4i+2$ doesn't move it. Hence,

$$(j(2i+1), k(2i+1)) = (j(2i), k(2i)-1),$$

the location of the $m-1^{st}$ smallest element at the end of the sorting procedure. □

Lemma 16 *If $(j(2i+1), k(2i+1))$ is the cell occupied by the m^{th} smallest entry at the end of the sorting procedure, then $(j(2i+2), k(2i+2))$ is the cell occupied by the $m-1^{st}$ smallest entry when the sort is complete.*

Finally, Theorem 12 and its proof still apply.