AUSTRALASIAN JOURNAL OF COMBINATORICS
Volume **31** (2005), Pages 85–106

Bubble sort with erroneous comparisons

Petros Hadjicostas

Department of Mathematics and Statistics
Texas Tech University, Box 41042
Lubbock, TX 79409-1042
U.S.A.
phadjico@math.ttu.edu

K. B. Lakshmanan

Department of Computer Science
SUNY Brockport
350 New Campus Drive
Brockport, NY 14420
U.S.A.
klakshma@brockport.edu

Abstract

In this paper, we analyze the familiar bubble sort algorithm and quantify the deviation of the output from the correct sorted order if the outcomes of one or more comparisons are in error. We thus generalize the results of Islam and Lakshmanan (1990). We show that the resulting disorder in the output of the bubble sort with $\lfloor n/2 \rfloor$ errors, where n is the size of the list sorted, is comparable to that of straight insertion sort, recursive merge sort and heapsort algorithms with just one error.

1 Introduction and review of the literature

Reliability of computer hardware and software is a major concern in many applications, whether they are in space exploration, patient monitoring, telephone call routing, or simple multi-media communication. Therefore, fault diagnosis, fault-tolerant computing, and distributed algorithms have become areas of active computer science research, involving sophisticated hardware and software techniques ([12] and [19]). In general, there are two main themes seen in the literature in this area: (a) the study of existing widely-used algorithms in order to determine how well they cope with failures, or (b) the development of new algorithms that are demonstrably fault-tolerant. In coping with faults, there are two approaches. The first is the reconfiguration approach in which faults are diagnosed, faulty components are isolated, and the system is reconfigured. The second approach requires devising robust algorithms that work correctly independent of the faulty behavior of some components, without explicitly singling out faulty ones.

Sorting and searching are two problems of fundamental nature studied quite extensively in combinatorics and computer science [10]. They arise in several contexts, e.g., in arranging entries in a database and later retrieving some of them. Because of their fundamental nature, sorting and searching algorithms are taught in the very first course in computing. A large number of algorithms have been developed, analyzed quite extensively, and classified. For serial computers (random access machines) that allow only one operation to be executed at a time, it is well known that $O(n \log n)$ comparisons are necessary for sorting n items in the worst case under the comparison tree model [10]. However, a number of interesting questions regarding sorting algorithms and the number of comparisons performed have been posed over the years, and have been answered partially:

(a) Is it possible to reduce the number of comparisons substantially if one settles for partially sorted outputs? ([17])

(b) If the number of comparisons that can be performed is limited, how should we use them to minimize the disorder of the output?

(c) If the list to be sorted is already partially sorted, how can the existing order be taken advantage of to reduce the number of comparisons? ([15])

(d) Is it possible to develop an alternative taxonomy of sorting algorithms based on factors other than the number of comparisons? ([16])

In recent years, inspired by the problems posed by the famous mathematicians Rényi and Ulam, a number of researchers have developed and analyzed variations of search, select and sort algorithms to cope with possible errors in comparisons (see [4], [8], and [22]). By an error, we mean the outcome of a binary comparison between two data elements is "no" when factually it should be "yes", and vice versa. Pelc [18] provides a comprehensive survey of the literature in this field.

Bagchi [3] and Lakshmanan *et al.* [11] have studied fault-tolerant algorithms for sorting with a worst-case upper bound on the number of erroneous comparisons. Feige *et al.* [8] have studied sorting under the noisy comparison tree model in which each node of the tree gives the wrong answer with some constant probability. Sorting networks present an interesting, but a different model of computation in which fault-tolerance has been explored in depth. A comparator, the basic building block, is a two-input, two-output element that can compare and place the larger of the two inputs on a specified output port and the smaller on the other output port. In contrast to serial computers, a network of comparators allows a number of operations to be done in parallel [5]. The size of the network is defined as the number of comparators, and is related to the overall cost. On the other hand, the depth of the network is defined as the maximum length of any path from the input to the output, and is related to the time to accomplish sorting. Ajtai *et al.* [1] were the first to present a network of aymptotically optimal size $O(n \log n)$ and depth $O(\log n)$. Yao and Yao [23] initiated the study of fault-tolerant sorting networks in which a faulty comparator simply outputs the two inputs without any processing, as if it is simply non-existent. Their analysis included two scenarios: one with a worst-case upper bound on the number of faulty comparators, and another with random faults. In the case of random faults, the objective is to ensure that the network sorts the output correctly with a high probability. The *passive fault* model of Yao and Yao have been extended in recent literature [13] to a *destructive fault* model in which a faulty comparator can output the two inputs in reversed order, or output one of the two inputs in both output ports. In this new model, *replicators* are therefore used to copy items. In [14], Leighton and Ma present tight bounds on the size of the fault-tolerant merging and sorting networks, analyzing both worst-case and random comparator faults.

In this paper, we analyze the familiar bubble sort algorithm in a new light: if the outcomes of one or more comparisons are in error, by how much will the output deviate from the correct sorted order? Obviously, crucial to all this work is the measure of disorder used in the analysis. Considerable work has been done in this area [7].

Let $a = (a_1, a_2, \ldots, a_n)$ be a list or finite sequence consisting of n *distinct* integers. Assume that the correct order for sorting is the ascending one. The degree of *disorder* of the list a can be quantified in a variety of ways (e.g., see [7], [10] and [15]): by the number of runs in a; the smallest number of elements in a that should be removed from a to leave it sorted; and the number of inversions in a. By a *run* in a we mean a non-descending sublist of consecutive elements in a, say $(a_i, a_{i+1}, \ldots, a_m)$, such that a_i is not preceded by a smaller number, and a_m is not followed by a larger number. For a sorted list a the number of runs is 1, while for a list a with n elements in reverse order the number of runs is n. The smallest number of integers that should be removed from a list a of n elements to leave it sorted is 0 for a sorted list, while this number equals $n-1$ for a list a in reverse order. By *inversion* in a we mean a pair of integers in a in the wrong order. For a sorted list a the number of inversions is 0, while for a sequence a in reverse order the number of inversions is $n(n-1)/2$. (For the notation used in this paper for the three measures of disorder, see Section 3.)

The three measures defined above are not related to each other in simple and obvious ways. For example, consider the following two sequences: $(1 + n/2, 2 + n/2, \ldots, n, 1, 2, \ldots, n/2)$ and $(2, 1, 4, 3, \ldots, n, n-1)$, where n is even. Both sequences require $n/2$ elements to be removed to leave them in ascending sorted order. But the first sequence has two runs and $n^2/4$ inversions, whereas the second one has $1 + n/2$ runs and $n/2$ inversions.

The problem considered is to quantify the degree of disorder of the output sequence of bubble sort, when some of its comparisons are erroneous. For a given length of an input sequence of positive integers and for a given number of erroneous comparisons, we calculate (or at least estimate) the worst and best case values of the three aforementioned measures of disorder for the possible output sequences obtained from bubble sort. We thus extend the results by Islam and Lakshmanan [9] for bubble sort.

One may intuitively guess that algorithms that sort efficiently, that is, those that use $O(n \log n)$ comparisons in the worst case will be more sensitive to errors than those that use $O(n^2)$ comparisons. Preliminary results by Islam and Lakshmanan [9] showed that the above intuitive conjecture might be incorrect. By analyzing sort algorithms under the assumption that exactly one comparison was in error, they found that the efficient merge sort algorithm is in fact less sensitive to errors than the straight insertion sort algorithm. In addition, even though both quicksort and merge sort use the divide-and-conquer strategy, there is a lot of difference in their performance when one comparison is in error. Computer simulation results in [9] show that the bubble sort is indeed the least sensitive to errors.

The organization of the paper is as follows: In Section 2, we give a brief review of bubble sort and explain what happens to the three measures of disorder as the algorithm progresses from one pass to the next one. In Section 3, we introduce the basic notation to be used in the paper. In Section 4, we prove some complementarity results for the number of runs and the number of inversions in the output sequence obtained after an erroneous execution of bubble sort operates on an input sequence. In particular, we give formulas that relate the maximum number of runs to the minimum number of runs, and the maximum number of inversions to the minimum number of inversions. In Sections 5, 6, 7, and 8 we give inequalities, equalities, and tables for the maximum number of runs, the maximum number of inversions, the maximum value of the smallest number of removals, and the minimum value of the smallest number of removals, respectively, for the output sequence obtained after an erroneous execution of bubble sort. Finally, Section 9 contains some concluding remarks.

2 Bubble sort

for $(i = 1; i < n; i = i + 1)$
    for $(j = n; j > i; j = j - 1)$
        if $(a_{j-1} > a_j)$
            swap a_{j-1} and a_j

Consider the bubble sort algorithm given above. If there are no errors in comparisons, at the beginning of the i^{th} pass, the smallest $i-1$ integers occupy their correct positions. During the i^{th} pass over the input list a, the algorithm "bubbles" the i^{th} smallest element in a to the i^{th} position (from the left) in the output sequence. The algorithm makes $n-1$ passes over the input list, with $n-i$ comparisons during the i^{th} pass. Hence it does exactly $n(n-1)/2$ comparisons in total. Moreover, in the absence of errors in comparisons, every swap between adjacent elements (that are not in the right order) removes one inversion. Hence, as the algorithm progresses, the number of inversions in the sequence goes down to 0 monotonically. In a list of distinct integers, the number of inversions is equal to the number of moves/swaps bubble sort performs when sorting the list. (In the context of Straight Insertion Sort, this is stated by [10], Section 5.2.1, p. 82.)

As the algorithm progresses, the minimum number of integers to be removed from the list to leave it sorted in ascending order goes down to 0 monotonically. On the other hand, a swap intended to remove an inversion may cause the number of runs to go up by one. Consider, for example, the sequence $(2, 4, 1, 3)$, which has two runs. Swapping 1 and 4 leads to three runs in the sequence. Still, the number of runs at the end of any pass will be no more than that at the end of the previous pass, eventually going down to 1.

3 Notation

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$. For each $n \in \mathbb{N}\backslash\{0, 1\}$, let $\mathcal{A}_n$ be the set of all lists with n *distinct* integers as elements. Let, also, $\mathcal{B}_n$ be the set of all executions of the bubble sort algorithm, $B : \mathcal{A}_n \to \mathcal{A}_n$, that can sort lists of length n, and can make up to $n(n-1)/2$ errors when making comparisons. This means that, for each $B \in \mathcal{B}_n$, the collection of comparisons where B is erring is uniquely associated with B. Let also $E(B)$ be the total number of errors B does.

For each $n \in \mathbb{N}\backslash\{0, 1\}$, $a \in \mathcal{A}_n$, and $B \in \mathcal{B}_n$, let $R(a, B)$ be the number of runs in the output list after B operates on a. Obviously, $1 \leq R(a, B) \leq n$. For each integer e with $0 \leq e \leq n(n-1)/2$, let

$$\begin{aligned} \mathrm{Mruns}(n, e) &= \max\{R(a, B) : \ a \in \mathcal{A}_n, \ B \in \mathcal{B}_n, \ E(B) = e\}; \\ \mathrm{mruns}(n, e) &= \min\{R(a, B) : \ a \in \mathcal{A}_n, \ B \in \mathcal{B}_n, \ E(B) = e\}. \end{aligned}$$

In other words, $\mathrm{Mruns}(n, e)$ and $\mathrm{mruns}(n, e)$ represent the worst and the best case scenario, respectively, for the number of runs in the output list obtained when an execution of bubble sort with exactly e errors operates on a list of integers with length n.

For each $n \in \mathbb{N}\backslash\{0, 1\}$, $a \in \mathcal{A}_n$, and $B \in \mathcal{B}_n$, let $RM(a, B)$ be the smallest number of integers that should be removed from the output sequence, after B operates on a, to leave it sorted. Obviously, $0 \leq RM(a, B) \leq n - 1$. For each integer e with $0 \leq e \leq n(n-1)/2$, let

$$\begin{aligned} \mathrm{Mrem}(n, e) &= \max\{RM(a, B) : \ a \in \mathcal{A}_n, B \in \mathcal{B}_n, E(B) = e\}; \\ \mathrm{mrem}(n, e) &= \min\{RM(a, B) : \ a \in \mathcal{A}_n, B \in \mathcal{B}_n, E(B) = e\}. \end{aligned}$$

Again, $\mathrm{Mrem}(n, e)$ and $\mathrm{mrem}(n, e)$ represent the worst and the best case scenarios, respectively.

For each $n \in \mathbb{N}\backslash\{0, 1\}$, $a \in \mathcal{A}_n$, and $B \in \mathcal{B}_n$, let $I(a, B)$ be the number of inversions in the output list after B operates on a. Obviously, $0 \leq I(a, B) \leq n(n-1)/2$. For each integer e with $0 \leq e \leq n(n-1)/2$, let

$$\begin{aligned} \mathrm{Minv}(n, e) &= \max\{I(a, B) : \ a \in \mathcal{A}_n, B \in \mathcal{B}_n, E(B) = e\}; \\ \mathrm{minv}(n, e) &= \min\{I(a, B) : \ a \in \mathcal{A}_n, B \in \mathcal{B}_n, E(B) = e\}. \end{aligned}$$

Thus, $\mathrm{Minv}(n, e)$ and $\mathrm{minv}(n, e)$ represent the worst and the best case scenarios, respectively.

4 Complementarity results

Let $n \in \mathbb{N}\backslash\{0,1\}$. If $a \in \mathcal{A}_n$, let ranks$(a) = (r_1, r_2, \ldots, r_n)$ be the list of ranks of the elements of a, where the smallest number gets the smallest rank. In other words, for $1 \le i \le n$, r_i is the rank of the i^{th} element of a. Obviously, ranks(a) is a permutation of the integers $1, 2, \ldots, n$. In addition, the list $(n+1-r_1, n+1-r_2, \ldots, n+1-r_n)$ is also a permutation of the first n positive integers. Consider the list $\overline{a} \in \mathcal{A}_n$, called the *complement* of a, which is created by putting in the i^{th} position the element of a whose rank is $n+1-r_i$. Then ranks$(\overline{a}) = (n+1-r_1, n+1-r_2, \ldots, n+1-r_n)$.

For a list $a \in \mathcal{A}_n$, let $R(a)$ and $I(a)$ be the number of runs and the numbers of inversions, respectively, in a. For a finite set A, let $\#A$ denote the number of elements of A. The proof of the following lemma is easy and hence is omitted.

Lemma 4.1 *$R(\overline{a}) + R(a) = n + 1$ and $I(\overline{a}) + I(a) = n(n-1)/2$.*

Let $\mathcal{D}_{1n}$ be the set of all atomic operations $OP - C - SI : \mathcal{A}_n \to \mathcal{A}_n$ (for operation compare and swap if inversion is discovered) that compares two items in a sequence in $\mathcal{A}_n$, and if it is discovered that they constitute an inversion, it swaps the items (the smallest number goes to the left). For each $OP - C - SI \in \mathcal{D}_{1n}$, there is a unique doubleton set $\{i, j\} \subseteq \{1, 2, \ldots, n\}$ (with $i < j$) such that for each $a \in \mathcal{A}_n$, the elements a_i and a_j are compared. Define also the *complement* operation $\overline{OP - C - SI} : \mathcal{A}_n \to \mathcal{A}_n$ (with associated doubleton set $\{i, j\}$, where $1 \le i < j \le n$) such that for each $a \in \mathcal{A}_n$, it compares a_i and a_j, and swaps them if and only if there is no inversion (i.e., $a_i < a_j$).

Let $\mathcal{D}_{2n} = \{\overline{OP - C - SI} : OP - C - SI \in \mathcal{D}_{1n}\}$, and $\mathcal{D}_n = \mathcal{D}_{1n} \cup \mathcal{D}_{2n}$. For each $S \in \mathcal{D}_n$, the *complement* $\overline{S}$ can be defined in the obvious way.

Lemma 4.2 *For any $n \in \mathbb{N}\backslash\{0,1\}$, $a \in \mathcal{A}_n$ and $S \in \mathcal{D}_n$, we have $S(a) = \overline{S}(\overline{a})$.*

Proof: Let ranks$(a) = (s_1, \ldots, s_n)$. Then ranks$(\overline{a}) = (n+1-s_1, \ldots, n+1-s_n)$. Assume the atomic operation $S : \mathcal{A}_n \to \mathcal{A}_n$ compares the i^{th} and j^{th} element of any sequence in $\mathcal{A}_n$, where $1 \le i < j \le n$. Note that $S(a)$ is a permutation of a and $\overline{S}(\overline{a})$ is a permutation of $\overline{a}$.

If S is in error, then $S \in \mathcal{D}_{2n}$ and $\overline{S}$ is not in error. In such a case, for $k \in \{1, 2, \ldots, n\}$:

$$\text{ranks}(S(a))_k = \begin{cases} s_k & \text{if } k \neq i, j \\ \max(s_i, s_j) & \text{if } k = i \\ \min(s_i, s_j) & \text{if } k = j \end{cases},$$

and

$$\text{ranks}(\overline{S}(\overline{a}))_k = \begin{cases} n+1-s_k & \text{if } k \neq i, j \\ \min(n+1-s_i, n+1-s_j) & \text{if } k = i \\ \max(n+1-s_i, n+1-s_j) & \text{if } k = j \end{cases}.$$

Since

$$\max(s_i, s_j) + \min(n+1-s_i, n+1-s_j) = n+1$$

and

$$\min(s_i, s_j) + \max(n+1-s_i, n+1-s_j) = n+1,$$

it follows that $\overline{S}(\overline{a})$ is the complement of $S(a)$.

If S is not in error, then $S \in \mathcal{D}_{1n}$ and we can arrive to a similar conclusion by a similar argument. This concludes the proof of the lemma. □

If $B \in \mathcal{B}_n$ is an execution of bubble sort, define its *complement*, $\overline{B} \in \mathcal{B}_n$, to be the execution of bubble sort that errs in comparison k of pass i (where $1 \leq i \leq n-1$ and $1 \leq k \leq n-i$) if and only if B does not err there. It follows that $E(\overline{B}) = n(n-1)/2 - E(B)$. Using finite induction and Lemma 4.2 we can easily show the following result.

Lemma 4.3 *If $a \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$, then the output list we get when $\overline{B}$ operates on $\overline{a}$ is the complement of the output list we get when B operates on a.*

Recall that, for each $n \in \mathbb{N}\backslash\{0,1\}$, $a \in \mathcal{A}_n$, and $B \in \mathcal{B}_n$, the integers $R(a,B)$ and $I(a,B)$ are the number of runs and number of inversions, respectively, in the output list after B operates on a. Combining Lemmas 4.1 and 4.3, we can prove the following result.

Corollary 4.4 *For $a \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$, $R(\overline{a}, \overline{B}) = n+1-R(a,B)$ and $I(\overline{a}, \overline{B}) = n(n-1)/2 - I(a,B)$.*

For integers e, k, m with $0 \leq e, m \leq n(n-1)/2$, and $1 \leq k \leq n$, define:

$$\begin{aligned} SR(n,e;k) &= \{(\text{ranks}(a), B) : \ a \in \mathcal{A}_n, \ B \in \mathcal{B}_n, \ E(B) = e, \ R(a,B) = k\}; \\ SI(n,e;m) &= \{(\text{ranks}(a), B) : \ a \in \mathcal{A}_n, \ B \in \mathcal{B}_n, \ E(B) = e, \ I(a,B) = m\}. \end{aligned}$$

Let P_n be the set of permutations of the first n positive integers, and let $\mathcal{B}_n^e$ be the set of all executions of bubble sort that make exactly e errors. For given n and e, the sets $SR(n,e;k)$, $k = 1, 2, \ldots, n$, partition the set $P_n \times \mathcal{B}_n^e$, which has $n!\binom{n(n-1)/2}{e}$ elements. The same happens with the sets $SI(n,e;m)$, $m = 0, 1, \ldots, n(n-1)/2$. Note, however, that some of these sets may be empty. Define

$$\text{Nruns}(n,e;k) = \#SR(n,e;k), \quad \text{Ninv}(n,e;m) = \#SI(n,e;m).$$

We are now ready to prove the main result of the section.

Theorem 4.5 *If e, k, m are integers with $0 \leq e, m \leq n(n-1)/2$, and $1 \leq k \leq n$, then*

$$\begin{aligned} \text{Nruns}(n,e;k) &= \text{Nruns}(n, n(n-1)/2 - e; \ n+1-k); \\ \text{Ninv}(n,e;m) &= \text{Ninv}(n, n(n-1)/2 - e; \ n(n-1)/2 - m). \end{aligned}$$

Proof: Define the mapping $f : P_n \times \mathcal{B}_n \to P_n \times \mathcal{B}_n$ by $f(a, B) = (\overline{a}, \overline{B})$ for all $(a, B) \in P_n \times \mathcal{B}_n$. It is easy to show that f is well-defined and one-to-one. For $A \subseteq P_n \times \mathcal{B}_n$, let $f(A)$ be the image of A under f. It follows from Corollary 4.4 that:

$$\begin{aligned} f(SR(n,e;k)) &= \{(\text{ranks}(\overline{a}), \overline{B}) : a \in \mathcal{A}_n,\ B \in \mathcal{B}_n,\ E(B) = e,\ R(a,B) = k\} \\ &= \{(\text{ranks}(a'), B') : a' \in \mathcal{A}_n,\ B' \in \mathcal{B}_n,\ E(B') = n(n-1)/2 - e, \\ &\quad\ R(a', B') = n + 1 - k\}. \end{aligned}$$

Therefore, $f(SR(n,e;k)) = SR(n, n(n-1)/2 - e; n+1-k)$. By a similar argument we can show that $f(SI(n,e;m)) = SI(n, n(n-1)/2 - e; n(n-1)/2 - m)$. Since f is one-to-one, the statement of the theorem follows easily. □

Using the notation of Section 3, we have:

$$\begin{aligned} \text{Mruns}(n,e) &= \max\{k : \text{Nruns}(n,e;k) > 0\}, \\ \text{mruns}(n,e) &= \min\{k : \text{Nruns}(n,e;k) > 0\}, \\ \text{Minv}(n,e) &= \max\{m : \text{Ninv}(n,e;m) > 0\}, \\ \text{minv}(n,e) &= \min\{m : \text{Ninv}(n,e;m) > 0\}. \end{aligned}$$

Corollary 4.6 *If* $n \in \mathbb{N}\backslash\{0,1\}$ *and* e *is an integer with* $0 \le e \le n(n-1)/2$*, then:*

$$\text{mruns}(n,e) + \text{Mruns}(n, n(n-1)/2 - e) = n + 1;$$

$$\text{minv}(n,e) + \text{Minv}(n, n(n-1)/2 - e) = n(n-1)/2.$$

Proof: By Theorem 4.5,

$$\begin{aligned} \text{mruns}(n,e) &= \min\{k : \text{Nruns}(n,e;k) > 0\} \\ &= \min\{k : \text{Nruns}(n, n(n-1)/2 - e; n+1-k) > 0\} \\ &= n + 1 - \max\{l : \text{Nruns}(n, n(n-1)/2 - e; l) > 0\} \\ &= n + 1 - \text{Mruns}(n, n(n-1)/2 - e). \end{aligned}$$

A similar proof can be given for the second equality. □

The previous corollary allows us to prove results about $\text{mruns}(n,e)$ and $\text{minv}(n,e)$ by utilizing results about $\text{Mruns}(n,e)$ and $\text{Minv}(n,e)$, respectively. For this reason, in Sections 5 and 6 we do not state or prove any results about $\text{mruns}(n,e)$ and $\text{minv}(n,e)$.

Unfortunately, similar complementary results do not hold for $\text{mrem}(n,e)$ and $\text{Mrem}(n,e)$, and so in Sections 7 and 8 we prove separate results for each of these quantities.

5 Maximum number of runs

The following theorem gives some inequalities regarding $\text{Mruns}(n,e)$.

Theorem 5.1 *Let* $n \in I\!N \backslash \{0,1\}$. *Then:*

(a) For any integer e *with* $0 \le e \le n(n-1)/2$*:* $\text{Mruns}(n,e) \le \min(n, e+1)$.

(b) For any integer e *with* $0 \le e \le n(n-1)/2$*:*

$$(i)\ \text{Mruns}(n,e) \le \text{Mruns}(n+1,e) \quad (ii)\ \text{Mruns}(n,e) \le \text{Mruns}(n+1,e+1).$$

(c) For any integer e *with* $0 \le e \le n(n+2)/8$*:*

$$\text{Mruns}(n,e) \ge \left\lfloor \frac{\sqrt{8e+1}-1}{2} \right\rfloor .$$

(d) For any integer e *with* $0 \le e \le n(n-1)/2 - 1$, *if* $\text{Mruns}(n,e) = n$, *then* $\text{Mruns}(n,\epsilon) = n$ *for all integers* ϵ *with* $e < \epsilon \le n(n-1)/2$.

Proof: (a) It suffices to show that $\text{Mruns}(n,e) \le e+1$ for $0 \le e \le n-2$. We use finite induction on e. If $e = 0$, then $\text{Mruns}(n,0) = 1 = 0+1$. Let ϵ be an integer with $1 \le \epsilon \le n-2$, and assume for any integer e with $0 \le e < \epsilon$, we have $\text{Mruns}(m,e) \le e+1$ for all $m \in \mathbb{N}\backslash\{0,1\}$ with $e \le m-2$. (In such a case, the inequality $\text{Mruns}(m,e) \le e+1$ would then hold for all $m \in \mathbb{N}\backslash\{0,1\}$ with $e \le m(m-1)/2$.) Let $a \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$ with $E(B) = \epsilon$ be given. Assume that, when B operates on a, the first error happens in pass i (where $1 \le i \le n-1$). This means that the smallest $i-1$ integers from a have been "bubbled" to the left to their correct positions. If $i = n-1$, then $R(a,B) = 2 \le \epsilon + 1$.

Assume $1 \le i \le n-2$. Let a' be the list of length $n-i$ that results after i passes of B on a (and after ignoring the i elements of a that have been placed by B in positions from 1 through i). Let B' be the part of B that consists of passes $i+1, i+2, \ldots, n-1$. Then $a' \in \mathcal{A}_{n-i}$ and $B' \in \mathcal{B}_{n-i}$ with $E(B') < \epsilon$. By the induction hypothesis,

$$R(a',B') \le \text{Mruns}(n-i, E(B')) \le E(B') + 1 \le \epsilon.$$

It can be easily proved that $R(a,B) \le R(a',B') + 1$. Hence $R(a,B) \le \epsilon + 1$ for all $a \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$ with $E(B) = \epsilon$. Hence, $\text{Mruns}(n,\epsilon) \le \epsilon + 1$, and by the induction theorem the inequality has been proven.

(b)(i) By definition, there is a list $a \in \mathcal{A}_n$ and an execution $B \in \mathcal{B}_n$ of bubble sort such that $E(B) = e$ and $R(a,B) = \text{Mruns}(n,e)$. The list a may be chosen so that it is a permutation of the numbers $1, 2, \ldots, n$. Let $a' = (a, n+1)$ and let $B' \in \mathcal{B}_{n+1}$ be an execution of bubble sort that does *not* err in the first comparison of each pass, and for $1 \le i \le n-1$ and $2 \le k \le n+1-i$ it errs in comparison k of pass i if and only if B errs in comparison $k-1$ of pass i. Then $\text{Mruns}(n,e) = R(a,B) = R(a',B') \le \text{Mruns}(n+1,e)$.

(ii) Choose $a \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$ as in part (i). Define $a' = (a, n+1)$ and let $B' \in \mathcal{B}_{n+1}$ be defined as in part (i), but now assume that it also errs in the only comparison of the last pass. Then $R(a',B') = R(a,B)$ or $R(a,B)+1$, and so

$$\text{Mruns}(n,e) = R(a,B) \le R(a',B') \le \text{Mruns}(n+1,e+1).$$

(c) Let $a = (n, n-1, \dots, 1)$. Let $k(e)$ be the largest integer k such that $k(k+1)/2 \le e$. In other words,

$$k(e) = \left\lfloor \frac{\sqrt{8e+1} - 1}{2} \right\rfloor .$$

Since $e \le n(n+2)/8$, it follows that $k(e) \le n/2$. Let B be an execution of the bubble sort algorithm that errs exactly e times as follows: For each pass i with $1 \le i \le k(e)$, B errs in the last i comparisons of that pass; thereafter it errs $e - [k(e)(k(e)+1)/2]$ times somewhere in the remaining passes. (This execution exists since $i \le n - i$ for $1 \le i \le k(e)$.) Then the first $k(e)$ elements of the output sequence are $n, n-1, \dots, n-k(e)+1$. Hence the output sequence has at least $k(e)$ runs, and so $\text{Mruns}(n, e) \ge k(e)$.

(d) Let $0 \le e \le n(n-1)/2 - 1$ with $\text{Mruns}(n, e) = n$. We will show that $\text{Mruns}(n, e+1) = n$, from which the statement of the theorem follows. Choose a list $a \in \mathcal{A}_n$ and an execution $B \in \mathcal{B}_n$ such that $R(a, B) = \text{Mruns}(n, e) = n$ and $E(B) = e$. Without loss of generality, we may assume that a is a permutation of the integers $1, 2, \dots, n$. Since $e \ne n(n-1)/2$, there is at least one comparison of B that is not in error. Let i (where $1 \le i \le n-1$) be the last pass of B in which such a comparison exists. In passes $i+1, \dots, n-1$ (if there are any left), all the comparisons are in error. Let $B' \in \mathcal{B}_n$ be the execution of bubble sort that is identical to B, except that in the last comparison of pass i in which B is not in error, B' is in error. Denote this comparison by k (where $1 \le k \le n - i$).

Since $R(a, B) = n$, the output of the operation of B on a is the list $(n, n-1, \dots, 2, 1)$. If $i \ge 2$, it follows from the definition of B' that after $i-1$ passes, the integers $n, n-1, \dots, n-i+2$ have been placed in positions $1, 2, \dots, i-1$, respectively. We claim that (for $i \ge 1$), the number $n-i+1$ will be put in position i after pass i. This claim, along with the fact that all the comparisons in the remaining passes (if any) are in error, implies that the final output of the operation of B' on a will be $(n, n-1, \dots, 2, 1)$, whence part (d) of the theorem follows.

At the end of pass i of execution B, the number $n-i+1$ is placed in position i. Since in an error-free comparison the smallest number moves to the left and the largest to the right, and since comparison k of pass i of B is not in error, and since $n-i+1$ is the largest of the numbers $1, 2, \dots, n-i, n-i+1$, it follows that: (i) Comparison k is not the last comparison of pass i of B (i.e., $k \ne n-i$); and (ii) before pass i starts, $n-i+1$ is in the left of the two numbers that will be compared in comparison k. In execution B', comparison k of pass i is in error, while all the other comparisons of pass i of B' are the same (in terms of their error condition) with the corresponding comparisons in execution B. This means that $n-i+1$ will stay to the left of the numbers that are compared in comparison k, and being the largest of the numbers 1 through $n-i+1$, it will move to position i (since all the comparisons after k in pass i of B' are in error). This proves the previous claim. □

The bounds obtained in the previous theorem are not always achieved. We have $\text{Mruns}(4, 3) = 3$ (see Table 1), which is not equal to $\min(n, e+1) = 4$. Also, we have

Mruns(4, 2) = 3, which is not equal to

$$\lfloor(\sqrt{8e+1}-1)/2\rfloor = 1$$

(although $e \leq n(n+2)/8$). The following theorem gives some bounds or values for Mruns(n, e) for various values of n and e.

Theorem 5.2 *Let $n \in \mathbb{N}\backslash\{0,1\}$ and $e \in \mathbb{N}$. Then:*

(a) $\mathrm{Mruns}(n,1) = 2$.

(b) $\mathrm{Mruns}(n,2) = 3$ *for* $n \geq 3$.

(c) $\mathrm{Mruns}(n,3) = 4$ *for* $n \geq 5$.

(d) $4 \leq \mathrm{Mruns}(n,4) \leq 5$ *for* $n \geq 5$.

(e) If $n \geq 2e$*, then* $\mathrm{Mruns}(n,e) = e+1$.

(f) If $n \geq 3$ *and* k *is an integer such that* $1 \leq k \leq (n-1)/2$*, then*

$$\left\lfloor\frac{n}{2}\right\rfloor - k + 1 \;\leq\; \mathrm{Mruns}\left(n, \left\lfloor\frac{n}{2}\right\rfloor + k\right) \;\leq\; \left\lfloor\frac{n}{2}\right\rfloor + k + 1.$$

(g) $\mathrm{Mruns}(n,e) = n$ *for all integers* $e \geq \lfloor n^2/4\rfloor$.

Proof: (a) It was proven in [9].

(b) It follows from part (a) of Theorem 5.1 that $\mathrm{Mruns}(n,2) \leq \min(n, 2+1) = 3$ for $n \geq 3$. To finish the proof of (b), consider the list $a = (n, n-1, n-2, \ldots, 2, 1)$. Assume that B is the execution of the bubble sort algorithm that errs exactly two times as follows: During the last comparison of the first pass, and during the single comparison of the last pass. Thus, for $n \geq 4$ the output list is $(n, 1, \ldots, n-3, n-1, n-2)$, while for $n = 3$ the output list is $(3, 2, 1)$. In such a case, the number of runs in the output list is three, and so $\mathrm{Mruns}(n,2) \geq 3$ for $n \geq 3$. Hence, $\mathrm{Mruns}(n,2) = 3$.

(c) Assume that $n \geq 5$. It follows from part (a) of Theorem 5.1 that $\mathrm{Mruns}(n,3) \leq 4$. To show that $\mathrm{Mruns}(n,3) = 4$, let

$$a = (n, n-1, 1, 2, \ldots, n-3, n-2),$$

and let $B \in \mathcal{B}_n$ be the execution of bubble sort that errs in the last comparison of passes 1, 2 and $n-1$. Then the output list is

$$(n, 2, 1, 3, 4, \ldots, n-3, n-1, n-2)$$

if $n > 5$, and $(5, 2, 1, 4, 3)$ if $n = 5$. Thus, $\mathrm{Mruns}(n,3) \geq R(a,B) = 4$, which proves this part of the theorem.

(d) It follows from part (a) of Theorem 5.1 that $\mathrm{Mruns}(n,4) \leq 5$ for $n \geq 5$. If $n \geq 6$, let

$$a = (n, n-1, n-2, 1, 2, \ldots, n-3),$$

and let $B \in \mathcal{B}_n$ be the execution of the bubble sort that errs in the last comparison of passes 1, 2, $n-2$ and $n-1$. Then the output list is

$$(n, 2, 1, 3, 4, \ldots, n-4, n-2, n-1, n-3)$$

if $n > 7$; it is $(7,2,1,3,5,6,4)$ if $n = 7$; and it is $(6,2,1,4,5,3)$ if $n = 6$. Thus, $\mathrm{Mruns}(n,4) \geq R(a,B) = 4$ for $n \geq 6$.

If $n = 5$, let $a' = (5,4,3,1,2)$, and let $B' \in \mathcal{B}_5$ be the execution of the bubble sort that errs in the last comparison of passes 1 and 2, and in the first comparison of passes 3 and 4. Then the output list is $(5,2,1,4,3)$, which shows that $\mathrm{Mruns}(5,4) \geq R(a',B') = 4$.

Combining all cases, we can prove the claim in part (d) of the theorem.

(e) The case $e = 0$ is trivial. If $e = 1$ or 2, the equality has been proven in parts (a) and (b). Assume $3 \leq e \leq n/2$. By part (a) of Theorem 5.1, $\mathrm{Mruns}(n,e) \leq e+1$. To prove that $\mathrm{Mruns}(n,e) = e+1$, let $a = (n, n-1, n-2, \ldots, 2, 1)$, and let $B \in \mathcal{B}_n$ be the execution of bubble sort that errs in the last comparison of passes $1,3,5,\ldots,2e-1$. (Note that $5 \leq 2e-1 \leq n-1$.) If a_i is the output list after pass i $(1 \leq i \leq n-1)$, then:

$$
\begin{aligned}
a_1 &= (n,1,n-1,n-2,\ldots,2)\\
a_3 &= (n,1,3,2,n-1,n-2,\ldots,4)\\
a_5 &= (n,1,3,2,5,4,n-1,n-2,\ldots,6)\\
&\vdots\\
a_{2e-1} &= (n,1,3,2,5,4,7,\ldots,2e-3,2e-4,2e-1,2e-2,n-1,n-2,\ldots,2e)\\
&\vdots\\
a_{n-1} &= (n,1,3,2,5,4,7,\ldots,2e-3,2e-4,2e-1,2e-2,2e,2e+1,\ldots,n-1)
\end{aligned}
$$

It can be easily seen that a_{n-1} has exactly $e+1$ runs, which shows that $\mathrm{Mruns}(n,e) = e+1$.

(f) The right inequality follows from part (a) of Theorem 5.1. To prove the left inequality, note that part (b)(ii) of Theorem 5.1 implies that $\mathrm{Mruns}(n-2k, \lfloor n/2 \rfloor - k) \leq \mathrm{Mruns}(n, \lfloor n/2 \rfloor + k)$. Since $n - 2k \geq 2(\lfloor n/2 \rfloor - k)$, it follows from part (e) of this theorem that $\mathrm{Mruns}(n-2k, \lfloor n/2 \rfloor - k) = \lfloor n/2 \rfloor - k + 1$, which proves the left inequality.

(g) Let $k = \lfloor n/2 \rfloor$. Let $a = (n, n-1, \ldots, n-k+1, 1, 2, \ldots, n-k)$. Consider an execution B of bubble sort in which errors are introduced during the i^{th} pass as follows: If $i \leq k$, the first $n-k-1$ comparisons are error-free, while the remaining $k-i+1$ comparisons are erroneous. If $i > k$, all the comparisons are erroneous. Then the output list is $(n, n-1, \ldots, 1)$. Note that, if n is even, the total number of comparisons is $n(n-1)/2 = k(2k-1)$, while the total number of errors is $n(n-1)/2 - k(n-k-1) = k(2k-1) - k(k-1) = k^2 = \lfloor n^2/4 \rfloor$. If n is odd, then the total number of comparisons is $n(n-1)/2 = k(2k+1)$, while the total number of errors is $n(n-1)/2 - k(n-k-1) = k(2k+1) - k^2 = k^2 + k = (n^2-1)/4 = \lfloor n^2/4 \rfloor$. It follows that $\mathrm{Mruns}(n, \lfloor n^2/4 \rfloor) = n$. It follows from part (d) of Theorem 5.1 that $\mathrm{Mruns}(n,e) = n$ for all integers $e \geq \lfloor n^2/4 \rfloor$. $\square$

The following example shows that $\lfloor n^2/4 \rfloor$ is *not* always the smallest number of errors e for which $\mathrm{Mruns}(n,e) = n$. Let $n = 8$ and $e = 14 < 16 = 8^2/4$. Let $a = (8,7,5,6,1,2,3,4)$, and let $B \in \mathcal{B}_8$ be the following execution of bubble sort: In pass 1, comparisons 4, 6, 7 are erroneous; in pass 2, comparisons 4, 6 are erroneous; in pass 3, comparisons 4, 5 are erroneous; in pass 4, comparison 4 is erroneous; in pass 5, comparisons 1, 2, 3 are erroneous; in pass 6, comparisons 1, 2 are erroneous; and finally in pass 7, comparison 1 is erroneous. The ouput sequence is $(8, 7, 6, 5, 4, 3, 2, 1)$, which means $\text{Mruns}(n, e) = R(a, B) = 8 = n$.

Table 1 gives the maximum number of runs for small values of n. They have been calculated either using the results of the section, or using a C++ program, which is available from the second author.

Table 1: Values of $\text{Mruns}(n, e)$ when n is small.

	e																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
n	2	1	2														
	3	1	2	3	3												
	4	1	2	3	3	4	4	4									
	5	1	2	3	4	4	4	5	5	5	5	5					
	6	1	2	3	4	5	5	5	5	6	6	6	6	6	6	6	6

6 Maximum number of inversions

The following theorem gives some results regarding $\text{Minv}(n, e)$.

Theorem 6.1 *Let* $n \in I\!N \backslash \{0, 1\}$. *Then:*

(a) $\text{Minv}(n, 1) = n - 1$.

(b) $\text{Minv}(n, 2) = 2n - 4$ *for* $n \geq 4$.

(c) For any integer e *with* $0 \leq e \leq n(n-1)/2$*:*

$$\max(e(n-e), 0) \leq \text{Minv}(n, e) \leq e(n-2) + 1.$$

(d) For any integer e *with* $0 \leq e \leq n(n+2)/8$*:*

$$\text{Minv}(n, e) \geq \frac{k(e)[2n - k(e) - 1]}{2},$$

where

$$k(e) = \left\lfloor \frac{\sqrt{8e+1} - 1}{2} \right\rfloor.$$

(e) For any integer e *with* $0 \leq e \leq n(n-1)/2 - 1$, *if* $\text{Minv}(n, e) = n(n-1)/2$, *then* $\text{Minv}(n, \epsilon) = n(n-1)/2$ *for all integers* ϵ *with* $e < \epsilon \leq n(n-1)/2$.

(f) $\text{Minv}(n, e) = n(n-1)/2$ *for all integers* $e \geq \lfloor n^2/4 \rfloor$.

(g) $\text{Minv}(n, e+1) \geq \text{Minv}(n, e)$ *for all integers* e *with* $0 \leq e < n(n-1)/2$.

Proof: (a) It was proven in [9].

(b) Let $n \geq 4$. From part (c) of this theorem (with $e = 2$) we get $2n - 4 \leq \mathrm{Minv}(n, 2) \leq 2n-3$. Assume $\mathrm{Minv}(n, 2) = 2n-3$. We shall arrive to a contradiction.

We may choose $a \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$ with $E(B) = 2$ such that $I(a, B) = \mathrm{Minv}(n, 2) = 2n - 3$, and such that a is a permutation of the integers $1, 2, \ldots, n$. For $j \in \{1, 2, \ldots, n\}$, let $I(j)$ be the number of inversions, in each of which integer j is the largest number, that occur in the output list obtained after the operation of B on a. Then $0 \leq I(j) \leq n - 1$, and for two distinct integers $j, j' \in \{1, 2, \ldots, n\}$ we have $I(j) + I(j') \leq (n - 1) + (n - 2) = 2n - 3$. It is then clear that the only way to have $I(a, B) = 2n - 3$ and $E(B) = 2$ is when the output list is $(n, n - 1, 1, 2, \ldots, n - 2)$. It is impossible for both errors to occur in the first pass, for otherwise in the output list the last $n - 1$ integers would be sorted. Therefore, one error occurs in pass 1 (and involves n), and the other one occurs in pass 2 (for otherwise the output list would not be as claimed above). The first two integers of the output list obtained after the first pass should be n and 1 (in that order). It is clear then that the second error should involve 1 and $n - 1$. Since $n - 1$ is the second largest integer of the list a and since $n \geq 4$, we need at least two errors in the second pass to move $n - 1$ to the second position. This is a contradiction.

(c) The left inequality needs to be proven only when $1 \leq e \leq n - 1$. We let $a = (n, n - 1, \ldots, n - e + 1, 1, 2, \ldots, n - e)$, and B be the execution of bubble sort that errs in comparison $n - e$ of pass i for $i = 1, 2, \ldots, e$. Then the output list is $(n - e + 1, \ldots, n - 1, n, 1, 2, \ldots, n - e)$, and so $\mathrm{Minv}(n, e) \geq I(a, B) = e(n - e)$.

To show the right inequality, use finite induction on e. The proof of this inequality is similar to the proof of part (a) of Theorem 5.1, and hence is omitted.

(d) Let $a = (n, n - 1, \ldots, 1)$. Let B be the execution of bubble sort (with errors) that is described in the proof of part (c) of Theorem 5.1. Then the first $k(e)$ elements of the output list are $n, n-1, \ldots, n-k(e)+1$, and so the output sequence has at least $(n-1)+(n-2)+\ldots+(n-k(e))$ inversions. Thus, $\mathrm{Minv}(n, e) \geq k(e)[2n-k(e)-1]/2$.

(e) It follows from part (d) of Theorem 5.1 and from the fact that $\mathrm{Minv}(n, w) = n(n - 1)/2$ if and only if $\mathrm{Mruns}(n, w) = n$ (where $0 \leq w \leq n(n - 1)/2$).

(f) It follows from part (g) of Theorem 5.2 and the fact that $\mathrm{Minv}(n, w) = n(n - 1)/2$ if and only if $\mathrm{Mruns}(n, w) = n$ (where $0 \leq w \leq n(n - 1)/2$).

(g) Choose $a \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$ with $E(B) = e$ such that $\mathrm{Minv}(n, e) = I(a, B)$. Let $b = (b_1, \ldots, b_n)$ be the output list after the operation of B on a. Determine integer i such that $1 \leq i \leq n-1$, pass i contains at least one non-erroneous comparison, and passes $i+1, \ldots, n-1$ (if any) contain only erroneous comparisons. Such an integer i exists since $e < n(n - 1)/2$. Note that $b_{i+1}, \ldots, b_n$ are in reverse order in the output sequence.

Construct an execution $B' \in \mathcal{B}_n$ of bubble sort such that it is identical to execution B except that one more error is introduced in the last non-erroneous comparison of pass i. As a result, in the new execution B', all comparisons from the point of introduction of the new comparison are erroneous. Let $c = (c_1, \ldots, c_n)$ be the output sequence for this new execution. Clearly $b_j = c_j$ for $j = 1, \ldots, i - 1$ (since B and

B' are identical until pass $i-1$). Also, $c_i \geq b_i$, and $c_{i+1}, \ldots, c_n$ are in reverse order (since all the comparisons from the point of introduction of the new comparison are erroneous).

For a permutation $d_1, \ldots, d_n$ of the integers $1, 2, \ldots, n$, the contribution of any element d_j to the number of inversions is the number of values to its right that are less than d_j. So, for the output sequence c the number of inversions resulting from $c_1, \ldots, c_{i-1}$ must be the same as from $b_1, \ldots, b_{i-1}$ in the output sequence b. Since $c_i \geq b_i$, the contribution from c_i is not less than that from b_i. Since both $b_{i+1}, \ldots, b_n$ and $c_{i+1}, \ldots, c_n$ are in reverse order, their total contribution to inversions is the same.

Hence $\mathrm{Minv}(n, e+1) \geq I(a, B') \geq I(a, B) = \mathrm{Minv}(n, e)$. □

Table 2 gives the maximum number of inversions for small values of n.

Table 2: Values of $\mathrm{Minv}(n, e)$ when n is small.

n \ e	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1														
3	0	2	3	3												
4	0	3	4	5	6	6	6									
5	0	4	6	7	8	9	10	10	10	10	10					
6	0	5	8	9	11	12	13	14	15	15	15	15	15	15	15	15

7 Maximum value of the smallest number of removals

The following theorem gives some inequalities regarding $\mathrm{Mrem}(n, e)$, which is the maximum value of the smallest number of integers needed to be removed to leave the output list sorted. (The output list is obtained after an execution of the bubble sort with e errors operates on a list of length n.)

Theorem 7.1 *Let* $n \in I\!N \backslash \{0, 1\}$ *and* $e \in I\!N$.

(a) If $0 \leq e \leq n-1$, *then* $\min(n-e, e) \leq \mathrm{Mrem}(n, e) \leq e$.

(b) For any integer e *with* $0 \leq e \leq n(n-1)/2$:

(i) $\mathrm{Mrem}(n, e) \leq \mathrm{Mrem}(n+1, e)$ (ii) $\mathrm{Mrem}(n, e) \leq \mathrm{Mrem}(n+1, e+1)$.

(c) If $0 \leq e \leq n(n+2)/8$, *then:*

$$\mathrm{Mrem}(n, e) \geq \left\lfloor \frac{\sqrt{8e+1}-1}{2} \right\rfloor .$$

(d) For any integer e *with* $0 \leq e \leq n(n-1)/2 - 1$, *if* $\mathrm{Mrem}(n, e) = n-1$, *then* $\mathrm{Mrem}(n, \epsilon) = n-1$ *for all integers* ϵ *with* $e < \epsilon \leq n(n-1)/2$.

Proof: (a) For $e = 0$ the left inequality is trivially true, so assume $1 \le e \le n-1$. Let $a = (n, n-1, \ldots, n-e+1, 1, 2, \ldots, n-e)$, and B be the execution of bubble sort that errs in comparison $n-e$ of pass i for $i = 1, 2, \ldots, e$. Then the output list is $(n-e+1, \ldots, n-1, n, 1, 2, \ldots, n-e)$, and so $\mathrm{Mrem}(n, e) \ge RM(a, B) = \min(n-e, e)$. (Recall from Section 3 that $RM(a, B)$ is the smallest number of integers that should be removed from the output sequence, after B operates on a, to leave it sorted.)

To show the right inequality, we use finite induction on e. The proof of this inequality is similar to the proof of part (a) of Theorem 5.1, and hence is omitted.

(b)(i) By definition, there is a list $a \in \mathcal{A}_n$ and an execution $B \in \mathcal{B}_n$ of bubble sort such that $E(B) = e$ and $RM(a, B) = \mathrm{Mrem}(n, e)$. The list a may be chosen so that it is a permutation of the numbers $1, 2, \ldots, n$. Define $a' \in \mathcal{A}_{n+1}$ and $B \in \mathcal{B}_{n+1}$ as in the proof of part (b)(i) of Theorem 5.1. Then $\mathrm{Mrem}(n, e) = RM(a, B) = RM(a', B') \le \mathrm{Mrem}(n+1, e)$.

(ii) Let $a \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$ be such that $E(B) = e$ and $RM(a, B) = \mathrm{Mrem}(n, e)$. Assuming, without loss of generality, that a is a permutation of the numbers $1, 2, \ldots, n$, let $a' = (a, n+1)$, and define $B' \in \mathcal{B}_{n+1}$ as in the proof of part (b)(ii) of Theorem 5.1. Let b_n be the output list of the operation of B on a, and b_{n+1} be the output list of the operation of B' on a'. Let

$$\begin{aligned} K(b_n) &= \{X \subseteq \{1, 2, \ldots, n\} : \#X = RM(a, B)\}; \\ K(b_{n+1}) &= \{Y \subseteq \{1, 2, \ldots, n, n+1\} : \#Y = RM(a', B')\}. \end{aligned}$$

Thus, if X (with $\#X = RM(a, B)$) is removed from b_n, the rest of the list is left sorted. Similarly, if Y (with $\#Y = RM(a', B')$) is removed from b_{n+1}, the rest of the list is left sorted.

Assume c is the last element of b_n. Since there is an error in the last comparison of B', the last two elements of b_{n+1} are $n+1$ and c (in that order). (Note that, if $n+1$ is removed from b_{n+1}, the rest of the list is just b_n.) For each $Y \in K(b_{n+1})$, we have either $n+1 \in Y$ or $c \in Y$. We consider two cases: (A) $n+1 \in Y_0$ for some $Y_0 \in K(b_{n+1})$; and (B) $n+1 \notin Y$ for all $Y \in K(b_{n+1})$.

In case (A), let $X_0 = Y_0 - \{n+1\}$. Then by removing X_0 from b_n, the rest of the latter list is left sorted. This means $\#X_0 \ge RM(a, B)$, which implies $RM(a', B') = \#X_0 + 1 \ge RM(a, B) + 1$. On the other hand, by removing $n+1$ from b_{n+1}, we are left with b_n, and then by removing any $X \in K(b_n)$ we are left with a sorted list. Thus, $RM(a', B') \le \#X + 1 = RM(a, B) + 1$. It follows that $RM(a', B') = RM(a, B) + 1$.

In case (B), $c \in Y$ for all $Y \in K(b_{n+1})$. Choose $Y_1 \in K(b_{n+1})$, and note that $Y_1 \subseteq \{1, 2, \ldots, n\}$. By removing Y_1 from b_{n+1}, we get a sorted list: apart from $n+1$, this is the same sorted list we would have got if we were removing Y_1 from b_n. Hence $RM(a', B') = \#Y_1 \ge RM(a, B)$. Now choose $Y_2 \in K(b_n)$. If $c \in Y_2$, then by removing Y_2 from b_{n+1}, we get a sorted list, i.e., $RM(a, B) = \#Y_2 \ge RM(a', B')$ (which implies $RM(a, B) = RM(a', B')$). If $c \notin Y_2$, then by removing $Y_2 \cup \{c\}$ from b_{n+1} we get a sorted list, i.e., $RM(a, B) + 1 = \#Y_2 \ge RM(a', B')$ (and so $RM(a, B) + 1 \ge RM(a', B') \ge RM(a, B)$).

Combining all cases, we conclude that $RM(a', B') = RM(a, B)$ or $RM(a, B) + 1$. It follows that: $\text{Mrem}(n, e) = RM(a, B) \leq RM(a', B') \leq \text{Mrem}(n + 1, e + 1)$.

(c) Let $a = (n, n-1, \ldots, 1)$. Let B be the execution of bubble sort (with errors) that is described in the proof of part (c) of Theorem 5.1. Then the first $k(e)$ elements of the output list are $n, n-1, \ldots, n-k(e)+1$. To leave the output list sorted one must either remove at least all the first $k(e)$ elements of it, or remove $n-1$ elements from it. This implies that $\text{Mrem}(n, e) \geq RM(a, B) \geq \min(k(e), n-1) = k(e)$ (since $n \geq 2$ and $e \leq n(n+2)/8$.)

(d) Let $0 \leq e \leq n(n-1)/2 - 1$ with $\text{Mrem}(n, e) = n - 1$. Let ϵ be such that $e < \epsilon \leq n(n-1)/2$. Then there is $a \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$ such that $E(B) = e$ and $RM(a, B) = n-1$. Without loss of generality, we may assume that a is a permutation of the integers $1, 2, \ldots, n$. It follows that the output list after B operates on a is $(n, n-1, \ldots, 1)$, i.e., $R(a, B) = n$. This implies that $\text{Mruns}(n, e) = n$, and so by Theorem 5.1, part (d), $\text{Mruns}(n, \epsilon) = n$. This in turn implies $\text{Mrem}(n, \epsilon) = n - 1$. □

The following theorem gives some values of $\text{Mrem}(n, e)$ for various values of n and e.

Theorem 7.2 *Let* $n \in \mathbb{N} \setminus \{0, 1\}$. *Then:*

(a) $\text{Mrem}(n, 1) = 1$.

(b) $\text{Mrem}(n, 2) = 2$ *for* $n \geq 3$.

(c) For any integer e *with* $0 \leq e \leq n/2$, $\text{Mrem}(n, e) = e$.

(d) $\text{Mrem}(n, e) = n - 1$ *for all integers* $e \geq \lfloor n^2/4 \rfloor$.

Proof: (a) It was proved in [9].

(b) For $n \geq 4$, this part follows from part (c). Assume $n = 3$. Then $\text{Mrem}(3, 2) \leq n - 1 = 2$. If $a = (3, 2, 1)$, and $B \in \mathcal{B}_3$ is the execution of bubble sort that errs in the last comparison of each of its two passes, then the output list is a itself. Then $2 = RM(a, B) \leq \text{Mrem}(3, 2)$. Hence $\text{Mrem}(3, 2) = 2$.

(c) It follows from part (a) of Theorem 7.1.

(d) It follows from part (g) of Theorem 5.2 and the fact that $\text{Mrem}(n, w) = n - 1$ if and only if $\text{Mruns}(n, w) = n$ (where $0 \leq w \leq n(n-1)/2$). □

Table 3 gives the maximum value of the smallest number of removals for small values of n.

8 Minimum value of the smallest number of removals

The following theorem gives some results regarding $\text{mrem}(n, e)$, which is the minimum value of the smallest number of integers needed to be removed to leave the output list sorted. (The output list is obtained after an execution of the bubble sort with e errors operates on a list of length n.)

Theorem 8.1 *Assume* $n \in \mathbb{N} \setminus \{0, 1\}$. *Then:*

Table 3: Values of $\mathrm{Mrem}(n, e)$ when n is small.

		e = 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
n	2	0	1														
	3	0	1	2	2												
	4	0	1	2	2	3	3	3									
	5	0	1	2	3	3	3	4	4	4	4	4					
	6	0	1	2	3	4	4	4	4	5	5	5	5	5	5	5	5

(a) $\mathrm{mrem}(n, 0) = 0$ *and* $\mathrm{mrem}(n, n(n-1)/2) = n - 1$.

(b) For any integer e *with* $0 < e \leq n(n-1)/2$, *if* $\mathrm{mrem}(n, e) = 0$, *then* $\mathrm{mrem}(n, \epsilon) = 0$ *for all integers* ϵ *with* $0 \leq \epsilon < e$.

(c) For any integer e *with* $(n-1)(n-2)/2 \leq e \leq n(n-1)/2$,

$$\mathrm{mrem}(n, e) \geq e - (n-1)(n-2)/2.$$

(d) For any integer e *with* $0 \leq e \leq n(n-1)/2$: $\mathrm{mrem}(n+1, e) \leq \mathrm{mrem}(n, e)$.

(e) For any integer e *with* $0 \leq e \leq n(n-1)/2$, *if* $\mathrm{mrem}(n, e) = 0$, *then* $\mathrm{mrem}(m, e) = 0$ *for all integers* m *with* $m > n$.

Proof: (a) The proof of this part of the theorem is easy, and hence is omitted.

(b) Choose $a \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$ with $E(B) = e$ such that $RM(a, B) = \mathrm{mrem}(n, e) = 0$. Then the output sequence, after B operates on a, is sorted, i.e., $R(a, B) = 1$. By Corollary 4.4, $R(\overline{a}, \overline{B}) = n + 1 - R(a, B) = n$. Since $E(\overline{B}) = n(n-1)/2 - e$, $\mathrm{Mruns}(n, n(n-1)/2 - e) = n$. Assume $0 \leq \epsilon < e$. Since $n(n-1)/2 - e < n(n-1)/2 - \epsilon \leq n(n-1)/2$, by Theorem 5.1, part (d), $\mathrm{Mruns}(n, n(n-1)/2 - \epsilon) = n$. Then there is $a' \in \mathcal{A}_n$ and $B' \in \mathcal{B}_n$ with $E(B') = n(n-1)/2 - \epsilon$ such that $R(a', B') = n$. By Corollary 4.4, $R(\overline{a'}, \overline{B'}) = n + 1 - R(a', B') = 1$, i.e., $RM(\overline{a'}, \overline{B'}) = 0$. Since $E(\overline{B'}) = \epsilon$, $\mathrm{mrem}(n, \epsilon) = 0$.

(c) We use finite backward induction on e. For $e = n(n-1)/2$, the inequality is true because $\mathrm{mrem}(n, n(n-1)/2) = n - 1 = n(n-1)/2 - (n-1)(n-2)/2$.

Let ϵ be an integer such that $(n-1)(n-2)/2 \leq \epsilon < n(n-1)/2$, and assume $\mathrm{mrem}(n, e) \geq e - (n-1)(n-2)/2$ for all $n \in \mathbb{N} \backslash \{0, 1\}$ and all integers e such that $n(n-1)/2 \geq e > \epsilon$. Let $a \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$ be an execution of bubble sort such that $E(B) = \epsilon$ and $\mathrm{mrem}(n, \epsilon) = RM(a, B)$. Since $\epsilon \neq n(n-1)/2$, there is at least one comparison in B that is not in error. Call the last pass where such a comparison exists k (where $1 \leq k \leq n-1$). In passes $k+1, \ldots, n-1$ (if there are any left), the execution B is always in error. Define the execution $B' \in \mathcal{B}_n$ to be identical to B, except that in all the comparisons of pass k, B' is in error.

In the output lists of the operations of B and B' on a, respectively, the first $k-1$ elements, say $c_1, c_2, \ldots, c_{k-1}$, are the same. (If $k = 1$, then there are no such c in the output lists.) Assume that after B operates on a, the remaining $n - (k-1)$

elements of a are $d_1, d_2, \ldots, d_{n-(k-1)}$ (in that order). In other words, the output list is $b = (c_1, c_2, \ldots, c_{k-1}, d_1, \ldots, d_{n-(k-1)})$. Also, if $k < n-1$, $d_2 > \ldots > d_{n-(k-1)}$. Let S be a set of elements of a such that, when S is removed from list b, the resulting list is sorted. We may choose S such that $\#S = RM(a, B)$. Note that at most one of the elements $d_2, \ldots, d_{n-(k-1)}$ can be out of S.

Denote the output list of the operation of B' on a by b'. We consider two cases: (i) $b = b'$ and (ii) $b \neq b'$. Obviously, in case (i), $RM(a, B') = RM(a, B)$.

In case (ii), d_2 occupies position k in output list b', and $d_2 > d_1$. In general, however, we do not know which position (after k) d_1 will occupy.

If $d_1, d_2 \in S$, then removing S from b' gives exactly the same (sorted) list as removing S from b. It follows then by the definition of $RM(a, B')$ that $RM(a, B') \leq RM(a, B)$.

If $d_1 \notin S$ and $d_2 \in S$, then removing both S and d_1 from b' gives a sorted list that is the sorted list obtained after removing S from b minus d_1. In such a case, $RM(a, B') \leq RM(a, B) + 1$.

If $d_1 \in S$ and $d_2 \notin S$, then by removing S and d_2 from b' we get a sorted list, and so $RM(a, B') \leq RM(a, B) + 1$.

Finally, if $d_1, d_2 \notin S$, then $\{d_3, \ldots, d_{n-(k-1)}\} \subseteq S$. Also, from the definition of S, if $k > 1$, then $(c_i \notin S \Rightarrow d_1 > c_i)$ for $i = 1, \ldots, k-1$. Hence, by removing S and d_2 from b', we get a sorted list. Thus, $RM(a, B') \leq RM(a, B) + 1$.

Combining all cases, we conclude that $RM(a, B') \leq RM(a, B) + 1$. Since $E(B') > E(B) = \epsilon$, it follows that $E(B') \geq \epsilon + 1$. Also, it follows from the induction hypothesis that

$$\begin{aligned} &\epsilon + 1 - (n-1)(n-2)/2 \ \leq\ E(B') - (n-1)(n-2)/2 \ \leq\ \mathrm{mrem}(n, E(B')) \\ &\leq\ RM(a, B') \ \leq\ RM(a, B) + 1 \ =\ \mathrm{mrem}(n, \epsilon) + 1. \end{aligned}$$

Therefore, $\epsilon - (n-1)(n-2)/2 \leq \mathrm{mrem}(n, \epsilon)$, and the induction is complete.

(d) By definition, there is a list $a \in \mathcal{A}_n$ and an execution $B \in \mathcal{B}_n$ of bubble sort such that $E(B) = e$ and $RM(a, B) = \mathrm{mrem}(n, e)$. The list a may be chosen so that it is a permutation of the numbers $1, 2, \ldots, n$. Define $a' \in \mathcal{A}_{n+1}$ and $B' \in \mathcal{B}_{n+1}$ as in the proof of part (b)(i) of Theorem 5.1. Then $\mathrm{mrem}(n+1, e) \leq RM(a', B') = RM(a, B) = \mathrm{mrem}(n, e)$.

(e) It follows from part (d) of this theorem. □

Table 4 gives the minimum number of the smallest number of removals for small values of n.

9 Concluding remarks

Algorithms for sorting are of fundamental importance in Computer Science, and existing algorithms have been analyzed quite exhaustively. In this paper, we analyzed the familiar bubble sort algorithm and quantified the deviation of the output from the correct sorted order if the outcomes of one or more comparisons are in error. Thus,

Table 4: Values of $\text{mrem}(n, e)$ when n is small.

		e															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	2	0	1														
	3	0	0	1	2												
n	4	0	0	0	1	1	2	3									
	5	0	0	0	0	0	1	1	2	2	3	4					
	6	0	0	0	0	0	0	0	0	1	1	2	2	3	3	4	5

this paper extends the work of Islam and Lakshmanan [9] who handled the case of a single error. Part of their results (for the worst case scenario) are summarized in Table 5. (For simplicity we assume that n is a multiple of 4.)

Table 5: Sort algorithms with one error: Worst-case scenario.

Algorithm	Number of runs	Smallest number of removals	Number of inversions
Bubble sort	2	1	$n-1$
Straight insertion sort	2	$n/2$	$n^2/4$
Recursive merge sort	2	$n/4$	$n^2/8+n/4$
Heapsort	$\Omega(n)$	$\Omega(n)$	$\Omega(n^2)$

One may intuitively guess that algorithms that sort efficiently, i.e., those that use $O(n\log_2 n)$ comparisons in the worst case, will be more sensitive to errors than those that use $O(n^2)$ comparisons. One of the surprising results of Islam and Lakshmanan [9] is that the above intuitive guess is not true, as presented in the table (e.g., compare straight insertion sort and recursive merge sort). Hence there is a need for a detailed analysis of all sorting algorithms when we have more than one comparison in error.

From Theorems 5.2(e), 7.2(c), and 6.1(c) of the paper, we can easily deduce the following corollary about the asymptotic behaviour of bubble sort when the number of errors is small compared to the length of the input list:

Corollary 9.1 *Let e be a fixed nonnegative integer. Then*
(a) $\lim_{n\to\infty} \text{Mruns}(n,e) = e+1$;
(b) $\lim_{n\to\infty} \text{Mrem}(n,e) = e$;
(c) $\lim_{n\to\infty} \text{Minv}(n,e)/n = e$.

Actually, Theorems 5.2(e), 7.2(c), and 6.1(c) say something more: If $e = \lfloor n/2 \rfloor$, then $\text{Mruns}(n,e) = \lfloor n/2 \rfloor + 1$, $\text{Mrem}(n,e) = \lfloor n/2 \rfloor$, and $n^2/4 - 1 \le \text{Minv}(n,e) \le n(n-2)/2 + 1$. In contrast, even a *single error* in comparison can lead to $O(n)$ minimum removals and $O(n^2)$ inversions for the algorithms presented in the table. In other words, the results of this paper show that the resulting disorder in the output of the bubble sort with $\lfloor n/2 \rfloor$ errors is comparable to that of straight insertion sort, recursive merge sort and heapsort algorithms with just one error.

An obvious direction for future research is to develop results for straight insertion sort, recursive merge sort and heapsort, allowing more than one error in comparisons. Then, we will be able to compare the performance of all the algorithms when multiple errors occur. Moreover, it is not clear if the measures of disorder chosen by us are the most appropriate for this comparison. It may be possible to analyze the algorithms for other measures of disorder proposed in the literature; see [7]. In [10], Knuth concludes the analysis of the bubble sort algorithm with the remark, "the bubble sort seems to have nothing to recommend it, except a catchy name and the fact that it leads to some interesting theoretical problems". Our results in this paper suggest that bubble sort is probably the least sensitive to errors. Even bubble sort has some redeeming features!

Acknowledgements

The authors thank the anonymous referee for many constructive comments that greatly improved the presentation of the paper. In particular, Lemma 4.2 was suggested by the referee.

References

[1] M. Ajtai, J. Komlos and E. Szemeredi, *An* $O(n \log n)$ *sorting network.* In: Proc. 15th ACM Symposium on Theory of Computing, Boston, MA, (1983), 1–9.

[2] S. Baase, *Computer algorithms: Introduction to design and analysis*, 2nd edn., Addison-Wesley, Reading, MA (1988).

[3] A. Bagchi, On sorting in the presence of erroneous information. *Inform. Process. Lett.* **43** (1992), 213–215.

[4] R.S. Borgstrom and S.R. Kosaraju, Comparison based search in the presence of errors, *Proc. 25th Ann. ACM Symposium on Theory of Computing* (1993), 130–136.

[5] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, *Introduction to Algorithms*, 2nd edn., McGraw-Hill, New York, NY, (2001).

[6] J. Czyzowicz, K.B. Lakshmanan and A. Pelc, Searching with a forbidden lie pattern in responses. *Inform. Process. Lett.* **37** (1991), 127–132.

[7] V. Estivill-Castro and D. Wood, A survey of adaptive sorting algorithms, *ACM Computing Surveys* **24** (1992), 441–476.

[8] U. Feige, P. Raghavan, D. Peleg and E. Upfal, Computing with noisy information, *SIAM J. Comput.* **23** (1994), 1001–1018.

[9] T. Islam and K.B. Lakshmanan, On the error-sensitivity of sort algorithms. In: *Advances in Computing and Information, 1990* (S.G. Akl, F. Fiala, W.W. Koczkodaj, eds.), Canadian Scholars' Press, Toronto, Canada, 81–85.

[10] D.E. Knuth, *The art of computer programming, Vol. 3: Sorting and Searching*, 2nd edn., Addison-Wesley, Boston, MA, (1998).

[11] K.B. Lakshmanan, B. Ravikumar and K. Ganesan, Coping with erroneous information while sorting, *IEEE Trans. Comp.* **40** (1991), 1081–1084.

[12] P.K. Lala, *Self-checking and fault-tolerant digital design*, Morgan Kaufmann, San Francisco, CA, (2001).

[13] T. Leighton and Y. Ma, Breaking the $\Theta(n \log^2 n)$ barrier for sorting with faults. *J. Computer and System Sci.* **54** (1997), 265–304.

[14] T. Leighton and Y. Ma, Tight bounds on the size of the fault-tolerant merging and sorting networks with destructive faults, *SIAM J. Comput.* **29** (1999), 258–273.

[15] H. Mannila, Measures of presortedness and optimal sorting algorithms, *IEEE Trans. Comp.* **34** (1985), 318–325.

[16] S.M. Merritt, An inverted taxonomy of sorting algorithms, *Communications of the ACM* **28** (1985), 96–99.

[17] J. Pearl, Theoretical bounds on the complexity of inexact computations, *IEEE Trans. Information Theory* **IT-22**(5) (1976), 580–586.

[18] A. Pelc, Searching games with errors — Fifty years of coping with liars, *Theoret. Comput. Sci.* **270** (2002), 71–109.

[19] D.K. Pradhan, *Fault-tolerant computer system design*, Prentice-Hall, Upper Saddle River, N.J. (1996).

[20] B. Ravikumar, K. Ganesan and K.B. Lakshmanan, On finding the largest element in spite of erroneous information. In: *Lecture Notes in Computer Science* (STACS 87; F.J. Brandenburg, G. Vidal-Naquet and M. Wirsing, eds.), **247**, Springer Verlag, Berlin, (1987), 88–99.

[21] G. Rawlins, *Compared to what? An introduction to the analysis of algorithms*, Computer Science Press, New York, NY, (1992).

[22] R.L. Rivest, A.R. Meyer, D.J. Kleitman, K. Winklmann and J. Spencer, Coping with errors in binary search procedures, *J. Comput. System Sci.* **20** (1980), 396–404.

[23] A.C. Yao and F.F. Yao, On fault-tolerant networks for sorting, *SIAM J. Computing* **14** (1985), 120–128.

(Received 30 Mar 2003; revised 15 Dec 2003)