On the Optimality of Tape Merge of Two Lists with Similar Size[*]

Qian Li[1], Xiaoming Sun[2], and Jialin Zhang[3]

1 CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China; and University of Chinese Academy of Sciences, Beijing, China
liqian@ict.ac.cn

2 CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China; and University of Chinese Academy of Sciences, Beijing, China
sunxiaoming@ict.ac.cn

3 CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China; and University of Chinese Academy of Sciences, Beijing, China
zhangjialin@ict.ac.cn

Abstract

The problem of merging sorted lists in the least number of pairwise comparisons has been solved completely only for a few special cases. Graham and Karp [18] independently discovered that the tape merge algorithm is optimal in the worst case when the two lists have the same size. Stockmeyer and Yao[28], Murphy and Paull[24], and Christen[6] independently showed when the lists to be merged are of size m and n satisfying $m \leq n \leq \lfloor\frac{3}{2}m\rfloor + 1$, the tape merge algorithm is optimal in the worst case. This paper extends this result by showing that the tape merge algorithm is optimal in the worst case whenever the size of one list is no larger than 1.52 times the size of the other. The main tool we used to prove lower bounds is Knuth's adversary methods [18]. In addition, we show that the lower bound cannot be improved to 1.8 via Knuth's adversary methods. We also develop a new inequality about Knuth's adversary methods, which might be interesting in its own right. Moreover, we design a simple procedure to achieve constant improvement of the upper bounds for $2m - 2 \leq n \leq 3m$.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases comparison-based sorting, tape merge, optimal sort, adversary method

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2016.51

1 Introduction

Suppose there are two disjoint linearly ordered lists A and B: $a_1 < a_2 < \cdots < a_m$ and $b_1 < b_2 < \cdots < b_n$ respectively, where the $m + n$ elements are distinct. The problem of merging them into one ordered list is one of the most fundamental algorithmic problems which has many practical applications as well as important theoretical significance. This problem has been extensively studied under different models, such as comparison-based model [18], parallel model [9], in-place merging model [11], etc. In this paper we focus on the classical

[*] This work was supported in part by the National Natural Science Foundation of China Grant 61222202, 61433014, 61502449, 61602440 and the China National Program for support of Top-notch Young Professionals.

27th International Symposium on Algorithms and Computation (ISAAC 2016).
Editor: Seok-Hee Hong; Article No. 51; pp. 51:1–51:17
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

comparison-based model, where the algorithm is a sequence of pairwise comparisons. People are interested in this model due to two reasons. Firstly, it is independent to the underlying order relation used, no matter it is "<" in **R** or another abstract order relation. Secondly, it is unnecessary to access the value of elements in this model. Such a restriction could come from security or privacy concerns where the only operation available is a zero-knowledge pairwise comparison which reveals only the order relation between elements.

The main theoretical question in this merge problem is to determine $M(m,n)$, the minimum number of comparisons which is always sufficient to merge the lists[18]. Given any algorithm g[1] to solve the (m,n) merging problem (i.e. where $|A| = m$ and $|B| = n$), let $M_g(m,n)$ be the number of comparisons required by algorithm g in the worst case, then

$$M(m,n) = \min_g M_g(m,n).$$

An algorithm g is said to be optimal on (m,n) if $M_g(m,n) = M(m,n)$. By symmetry, it is clear that $M(m,n) = M(n,m)$. To much surprise, this problem seems quite difficult in general, and exact values are known for only a few special cases. Knuth determined the value of $M(m,n)$ for the case $m,n \leq 10$ in his book [18]. Graham [18] and Hwang and Lin [16] completely solved the case $m = 2$ independently. The case $m = 3$ is quite a bit harder and was solved by Hwang [15] and Murphy [25]. Mönting solved the case $m = 4$ and also obtained strong results about $m = 5$ [26]. In addition, Smith and Lang [27] devised a computer program based on game solver techniques such as alpha beta search to compute $M(m,n)$. They uncovered many interesting facts including $M(7,12) = 17$, while people used to believe $M(7,12) = 18$.

Several different algorithms have been developed for the merge problem, among them *tape merge* or *linear merge* might be the simplest and the most commonly used one. In this algorithm, two smallest elements (initially a_1 and b_1) are compared, and the smaller one will be deleted from its list and placed on the end of the output list. Then repeat the process until one list is exhausted. It's easy to see that this algorithm requires $m+n-1$ comparisons in the worst case, hence $M(m,n) \leq m+n-1$. However, when m is much smaller than n, it is obvious that this algorithm becomes quite inefficient. For example, when $m = 1$, the merging problem is equivalent to an insertion problem and the rather different *binary insertion* procedure is optimal, i.e. $M(1,n) = \lceil \lg(n+1) \rceil$.

One nature question is "*when is tape merge optimal?*". By symmetry, we can assume $n \geq m$ and define $\alpha(m)$ be the maximum integer $n (\geq m)$ such that tape merge is optimal, i.e.

$$\alpha(m) = \max\{n \in \mathbb{N} \mid M(m,n) = m+n-1, n \geq m\}.$$

Assume a conjecture proposed by Knuth [18], which asserts that $M(m,n+1) \leq M(m,n)+1 \leq M(m+1,n)$, for $m \leq n$, is correct, it's easy to see tape merge is optimal if and only if $n \leq \alpha(m)$, and $\alpha(m)$ is monotone increasing.

Graham and Karp [18] independently discovered that $M(m,m) = 2m-1$ for $m \geq 1$. Then Knuth [18] proved $\alpha(m) \geq 4$ for $m \leq 6$. Stockmeyer and Yao[28], Murphy and Paull[24], and Christen[6] independently significantly improved the lower bounds by showing $\alpha(m) \geq \lfloor \frac{3}{2}m \rfloor + 1$, that is $M(m,n) = m+n-1$, for $m \leq n \leq \lfloor \frac{3}{2}m \rfloor + 1$. On the other hand, Hwang[14] showed that $M(m,2m) \leq 3m-2$, which implies $\alpha(m) \leq 2m-1$. For

[1] We only consider deterministic algorithms in this paper.

$m \leq n \leq 2m-1$, the best known merge algorithm is tape merge algorithm. It is conjectured by Fernandez et al. [8] that $\alpha(m) = \frac{1+\sqrt{5}}{2}m \pm o(m)$.

For general $n \geq m$, Hwang and Lin[14] proposed an in-between algorithm called *binary merge*, which excellently compromised between binary insertion and tape merge in such a way that the best features of both are retained. It reduces to tape merge when $n \leq 2m$, and reduces to binary insertion when $m = 1$. Let $M_{bm}(m,n)$ be the worst-case complexity of this algorithm. They showed that

$$M_{bm}(m,n) = m(1 + \lfloor \lg \frac{n}{m} \rfloor) + \lfloor \frac{n}{2^{\lfloor \lg \frac{n}{m} \rfloor}} \rfloor - 1.$$

Hwang and Deutsch[13] designed an algorithm which is optimal over all *insertive algorithms* including binary merge, where for each element of the smaller list, the comparisons involving it are made consecutively. However, the improvement for fixed n/m over binary merge increases more slowly than linearly in m[23]. Here we say that algorithm A_1 with complexity $M_{A_1}(m,n)$ is significantly faster for some fixed ratio n/m than algorithm A_2 with complexity $M_{A_2}(m,n)$, if $M_{A_2}(m,n) - M_{A_1}(m,n) = \Omega(m)$. The first significant improvement over binary merge was proposed by Manacher[23], which can decrease the number of comparisons by $\frac{31}{336}m$ for $n/m \geq 8$, and Thanh and Bui[31] further improved this number to $\frac{13}{84}m$. In 1978, Christen[5] proposed an elegant algorithm, called *forward testing and backward insertion*, which is better than binary merge when $n/m \geq 3$ and saves at least $\sum_{j=1}^{k} \lfloor \frac{m-1}{4^j} \rfloor$ comparisons over binary merging, for $n \geq 4^k m$. Thus it saves about $m/3$ comparisons when $n/m \to \infty$. Moreover, Christen's procedure is optimal for $5m - 3 \leq n \leq 7m$, i.e. $M(m,n) = \lfloor (11m + n - 3)/4 \rfloor$.

On the lower bound side, there are two main techniques in proving lower bounds. The first one is the information theoretic lower bound $I(m,n) = \lceil \lg \binom{m+n}{m} \rceil$. Hwang and Lin[14] have proved that

$$I(m,n) \leq M(m,n) \leq M_{bm}(m,n) \leq I(m,n) + m.$$

The second one is called *Knuth's adversary methods* [18]. The idea is that the optimal merge problem can be viewed as a two-player game with perfect information, in which the algorithm chooses the comparisons, while the adversary chooses (consistently) the results of these comparisons. It is easy to observe that $M(m,n)$ is actually the min-max value of this game. Thus a given strategy of the adversary provides a lower bound for $M(m,n)$. Mainly because of the consistency condition on the answers, general strategies are rather tedious to work with. Knuth proposed the idea of using of "disjunctive" strategies, in which a splitting of the remaining problem into two disjoint problems is provided, in addition to the result of the comparison. With this restricted adversary, he used term $.M.(m,n)$ to represent the minimum number of comparisons required in the algorithm, which is also a lower bound of $M(m,n)$. The detail will be specified in Section 2.

1.1 Our Results

In this paper, we first improve the lower bounds of $\alpha(m)$ from $\lfloor \frac{3}{2}m \rfloor + 1$ to $\lfloor \frac{38}{25}m \rfloor$ by using Knuth's adversary methods.

▶ **Theorem 1.** *$M(m,n) = m + n - 1$, if $m \leq n \leq \frac{38}{25}m$.*

We then show limitations of Knuth's adversary methods.

▶ **Theorem 2.** *$.M.(m,n) < m + n - 1$, if $n \geq 9\lceil m/5 \rceil$.*

This means that by using Knuth's adversary methods, it's impossible to show $\alpha(m) \geq 9\lceil m/5 \rceil \approx \frac{9}{5}m$ for any m.

When $m \leq n \leq 3m$, binary merge is the best known algorithm, which reduces to tape merge for $n \in [m, 2m]$ and gives $M_{bm}(m, 2m+k) = 3m + \lfloor k/2 \rfloor - 1$ for $k \in [0, m]$. In this paper, we give improved upper bounds for $M(m,n)$ for $2m - 2 \leq n \leq 3m$. In particular, it also improves the upper bounds of $\alpha(m)$, that is, $\alpha(m) \leq 2m - 3$ for $m \geq 7$.

▶ **Theorem 3.**

(a) $M(m, 2m+k) \leq 3m + \lfloor k/2 \rfloor - 2 = M_{bm}(m, 2m+k) - 1$, *if* $m \geq 5$ *and* $k \geq -1$.

(b) $M(m, 2m-2) \leq 3m - 4 = M_{bm}(m, 2m-2) - 1$, *if* $m \geq 7$. *That is* $\alpha(m) \leq 2m - 3$ *for* $m \geq 7$.

(c) $M(m, 2m) \leq 3m - 3 = M_{bm}(m, 2m) - 2$, *if* $m \geq 10$.

1.2 Related work

Besides the worst-case complexity, the average-case complexity has also been investigated for merge problems. Tanner [29] designed an algorithm which uses at most $1.06I(m,n)$ comparisons on average. The average case complexity of insertive merging algorithms as well as binary merge has also been investigated [7, 8].

Bui et al. [30] gave the optimal randomized algorithms for $(2,n)$ and $(3,n)$ merge problems and discovered that the optimal expected value differs from the optimal worst-case value by at most 1. Fernandez et al. [8] designed a randomized merging algorithm which performs well for any ratio n/m and is significantly faster than binary merge for $n/m > (\sqrt{5}+1)/2 \approx 1.618$. More preciously, they showed that

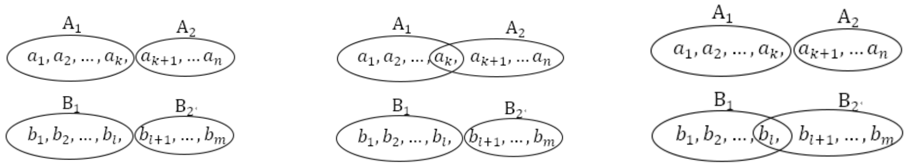
$$M_F(m,n) = \begin{cases} sn + (1+s)m, & if \quad 1+s \leq n/m \leq 2+s, \\ 2\sqrt{mn}, & if \quad 2+s \leq n/m \leq 2r, \end{cases} \tag{1}$$

where $s = (\sqrt{5}-1)/2 \approx 0.618$ and $r = (\sqrt{2} - 1 + \sqrt{2}s)^2 \approx 1.659$.

Nathan Linial [21] studied a more general problem where partial order relations are already known. He showed the information-theoretic lower bound is good, that is, an algorithms exists which merges A and B in no more than $(\lg(\sqrt{5}+1)/2)^{-1} \lg N_0$ comparisons, where N_0 is the number of extensions of the partial order on $A \cup B$. They also pointed out that this bound is tight, and the computation needed for finding the appropriate queries can be done in time polynomial in $m + n$.

Sorting, merging and searching are always closely related to each other. Manacher et al. [22] used efficient merge algorithms to improve Ford-Johnson sorting algorithm, which was conjectured to be optimal for almost twenty years. Linial and Saks [20] observed that $M(m,n)$ is equivalent to the minimum number of pairwise comparisons to determine whether a given element is present in a $m \times n$ matrix in which distinct entries are known to be increasingly ordered along rows and columns. They also studied the generalized problem in monotone multi-dimensional arrays, and their result was further improved by Cheng et al. [4]. Ajtai et al. [1] considered the problem of sorting and selection with imprecise comparisons. The non-uniform cost model has also been investigated [3, 10, 17, 12], for example, Huang et al. [12] studied the sorting problem where only a subset of all possible pairwise comparisons is allowed. For practical use, Brown and Tarjan [2] gave a merging procedure which runs in $O(I(m,n))$ time on a real computer.

Organization. We introduce some notations and explain Knuth's adversary methods in Section 2. In Section 3 we provide some properties of Knuth's adversary methods which will


A1
A2
$a_1, a_2, \ldots, a_k, a_{k+1}, \ldots a_n$
B1
B2
$b_1, b_2, \ldots, b_l, b_{l+1}, \ldots, b_m$

(a) Case 1, simple strategy **(b)** Case 2, complex strategy **(c)** Case 3, complex strategy

Figure 1 The Adversary's Splitting Strategies.

be used. In Section 4 we improve the lower bounds for $\alpha(m)$ via Knuth's adversary methods. Then we show limitations of this method in Section 5. Section 6 improves the upper bounds for $M(m, 2m+k)$. We conclude the paper with some open problems in Section 7.

2 Preliminaries

In this section, we introduce Knuth's adversary methods and present some notations.

We use the notations $\lambda M \rho$ proposed by Knuth in this paper. The detailed definitions can be found in Knuth's comprehensive monograph [18]. Yao et al. [28] gave an example to illustrate the use of that. Here, we briefly introduce the idea.

The basic idea of Knuth's adversary methods is to restrict the possible adversary strategies. In general, the adversary can arbitrarily answer the comparison query from the algorithm as long as there are no contradictions in his answer. But in Knuth's adversary methods, after each comparison query between a_i and b_j, the adversary is required to split each sorted list into two parts $A = A_1 \cup A_2$ and $B = B_1 \cup B_2$ (Figure 1). The adversary guarantees that each element in A_1 or B_1 is smaller than any element in A_2 or B_2. It is also guaranteed that a_i and b_j are not in the same subproblem, i.e. neither $a_i \in A_1, b_j \in B_1$ nor $a_i \in A_2, b_j \in B_2$, thus, the comparison result between a_i and b_j is determined after the splitting. Then the merge problem will be reduced to two subproblems (A_1, B_1) and (A_2, B_2) with different left or right constraints. For example, in case 2, the constraint for subproblem (A_1, B_1) is a right constraint $b_l < a_k$ since $a_k \in A_2$ while $b_l \in B_1$.

Knuth introduced notation $\lambda M \rho$ to represent nine kinds of restrict adversaries, where $\lambda, \rho \in \{., \backslash, /\}$ are the left and right constraint. In general, the constraint notation $'.'$ means no left (or right) constraint. Left constraint $\lambda = \backslash$ or $/$ means that outcomes must be consistent with $a_1 < b_1$ or $a_1 > b_1$ respectively. Similarly, right constraint $\rho = \backslash(/)$ means the outcomes must be consistent with $a_m < b_n$ (or $a_m > b_n$ respectively). Thus, merge problem $\lambda M \rho(A, B)$ will reduce to subproblem $\lambda M.(A_1, B_1)$ and $.M\rho(A_2, B_2)$ in case 1, to subproblem $\lambda M/(A_1, B_1)$ and $\backslash M \rho(A_2, B_2)$ in case 2, and to subproblem $\lambda M \backslash(A_1, B_1)$ and $/M\rho(A_2, B_2)$ in case 3. For convenience, we say the adversary adopts a *simple* strategy if he splits the lists in the way of case 1, otherwise the adversary adopts a *complex* strategy (case 2 or 3).

There are obvious symmetries, such as $/M.(m,n) = .M\backslash(m,n) = \backslash M.(n,m) = .M/(n,m)$, $/M/(m,n) = \backslash M \backslash(m,n)$, and $/M\backslash(m,n) = \backslash M/(n,m)$, which means we can deduce the nine functions to four functions: $.M.$, $/M.$, $/M\backslash$, and $/M/$. These functions can be calculated by a computer rather quickly, and the values for all $m, n \leq 150$ and the program are available in [19].

Note $M(m,n) \geq .M.(m,n)$, but $M(m,n)$ is not equal to $.M(m,n)$. in general, since we restrict the power of adversary in the decision tree model by assuming there is a (unknown) division of the lists after each comparison. But this restrict model still covers many interesting cases. For example, when $m \leq n \leq \lfloor \frac{3}{2}m \rfloor + 1$ [6, 24, 28] or $5m - 3 \leq n \leq 7m$ [5], $.M(m,n). = M(m,n)$.

Let $\lambda M_{i,j}\rho(m,n)$ denote the number of comparisons resulted from adversary's best strategy if the first comparison is a_i and b_j, thus $\lambda M\rho(m,n) = min_{i,j}\lambda M_{i,j}\rho(m,n)$.

The following notation is also used in our paper.

▶ **Definition 4.** Let $\overline{.M.}(m,n)$ be the difference of the number of comparisons required by tape merge in the worst case and $.M.(m,n)$, i.e. $\overline{.M.}(m,n) \triangleq m + n - 1 - .M.(m,n)$.

3 Inequalities about Knuth's adversary methods

In this section, we list several inequalities about $\lambda M\rho$, which will be used in Section 4 and Section 5.

▶ **Lemma 5.** *For any $\lambda, \rho \in \{., /, \backslash\}$, we have*
(a) $.M\rho(m,n) \geq \lambda M\rho(m,n)$*;*
(b) $/M\rho(m,n) \leq .M\rho(m,n-1) + 1$.

Proof. *Part (a)* is obvious, the adversary can perform at least as well on less restrictions. In *Part (b)*, if the first comparison is a_1 and b_1 for $/M\rho(m,n)$, the adversary has to claim $a_1 > b_1$, thus it reduces to $.M\rho(m,n-1)$. Therefore we have $/M\rho(m,n) \leq /M_{1,1}\rho(m,n) = 1 + .M\rho(m,n-1)$. ◀

The following lemma shows that if $.M.(m,n) = m + n - 1$, then tape merge is optimal for any (m', n') satisfying $m' \geq m$, $n' \leq n$ and $m' \leq n'$. The proof is in Appendix A.

▶ **Lemma 6.** *For any $m, n \geq 0$, $m + n \geq 1$ and $m \leq n$, we have $.M.(m+1,n) \geq .M.(m,n) + 1 \geq .M.(m,n+1)$ or $\overline{.M.}(m+1,n) \leq \overline{.M.}(m,n) \leq \overline{.M.}(m,n+1)$.*

We can show a similar statement about $/M.$ function as well. The proof is very similar, and we omit it here.

▶ **Lemma 7.** *For any $m, n \geq 1$, we have*
(a) $/M.(m+1,n+1) \geq /M.(m,n) + 2$ *[28];*
and for any $m, n \geq 1$ and $m \leq n$, we have
(b) $/M.(m,n+1) \leq /M.(m,n) + 1$*;*
(c) $/M.(m+1,n) \geq /M.(m,n) + 1$*, except $(m,n) = (1,1), (2,2)$ or $(3,3)$.*

4 Lower bounds for $\alpha(m)$

The key step is to show that $.M.(m+25,n+38) \geq .M.(m,n) + 63$, which directly implies Theorem 1. Since it's unavoidable to show similar statements for other restricted adversaries $\lambda M\rho$, we prove them by induction in parallel. In addition, by symmetry, we have $/M.(m,n) = .M\backslash(m,n)$, $\backslash M.(m,n) = .M/(m,n)$, and $/M/(m,n) = \backslash M\backslash(m,n)$, so the following theorem is enough for our goal.

▶ **Theorem 8.** *For $m, n \geq 0$ and $m + n \geq 1$, we have*
(a) $.M.(m+25,n+38) \geq .M.(m,n) + 63$*;*
and for $m, n \geq 1$, we have
(b) $/M.(m+25,n+38) \geq /M.(m,n) + 63$*;*
(c) $\backslash M.(m+25,n+38) \geq \backslash M.(m,n) + 63$*;*

(d) $/M\backslash(m+25,n+38) \geq /M\backslash(m,n)+63$*;*
(e) $\backslash M\backslash(m+25,n+38) \geq \backslash M\backslash(m,n)+63,$ *except* $(m,n)=(1,1)$*;*
(f) $\backslash M/(m+25,n+38) \geq \backslash M/(m,n)+63,$ *except* $(m,n)=(2,1)$.

Proof. The proof is by induction on m and n. The starting values for $m,n \leq 50$ are given in [19]. Now suppose the theorem holds for any m',n' satisfying $m' \leq m$, $n' \leq n$ and $m'+n' < m+n$, we then prove the case (m,n) where $m \geq 51$ or $n \geq 51$. Recall that our task is to design a strategy for the adversary for $(m+25,n+38)$.

Part (a). Suppose an algorithm begins by comparing a_i and b_j, if $i \leq m$ and $j \geq n+1$, then the adversary claims $a_i < b_j$ and follows the simple strategy, yielding

$$.M_{i,j}.(m+25,n+38) \geq 1+.M.(m,n)+.M.(25,38) = .M.(m,n)+63.$$

If $i \geq m+1$ and $j \leq n$, the adversary claims $a_i > b_j$ and uses the simple strategy. This leads to

$$.M_{i,j}.(m+25,n+38) \geq 1+.M.(m,n)+.M.(25,38) = .M.(m,n)+63.$$

If $i \leq m$ and $j \leq n$, assume if we compare a_i and b_j in $.M.(m,n)$, the adversary's best strategy is $1+.M\rho(p,q)+\lambda M.(s,t)$ where $\lambda,\rho \in \{.,/,\backslash\}$, then adversary uses the same strategy here, and we get

$$\begin{aligned}.M_{i,j}.(m+25,n+38) \geq& 1+.M\rho(p,q)+\lambda M.(s+25,t+38)\\ \geq& 1+.M\rho(p,q)+\lambda M.(s,t)+63 \geq .M.(m,n)+63\end{aligned}$$

by using the induction hypothesis.

If $i \geq m+1$ and $j \geq n+1$, then $i \leq 25$ and $j \leq 38$ cannot happen simultaneously, thus there are only three possible cases: $(i \geq 26,\ j \leq 38)$, $(i \leq 25,\ j \geq 39)$, or $(i \geq 26,\ j \geq 39)$. Reversing the order of the elements in A and B maps all these three cases to the above ones, so we can handle these cases as well by symmetry.

Therefore no matter which two elements are chosen to compare at the first step, the adversary can always find a strategy resulting the value not smaller than $.M.(m,n)+63$. This completes the proof of *Part (a)*.

Part (b). The proof for cases where $(i \leq m,\ j \leq n)$, $(i \geq m+1,\ j \leq n)$, and $(i \leq m,\ j \geq n+1)$ is similar with *Part (a)*.

If $i \geq m+1$ and $j \geq n+1$, then $i \leq 25$ and $j \leq 38$ cannot happen simultaneously, so we only need to consider the following cases:

If $i \leq 25$ and $j \geq 39$, or $i \geq 26$ and $j \leq 38$, the adversary uses the simple strategy, yielding

$$/M_{i,j}.(m+25,n+38) \geq 1+/M.(25,38)+.M.(m,n) \geq 1+62+.M.(m,n) \geq /M.(m,n)+63.$$

If $i \geq 26$ and $j \geq 39$: assume if we compare a_{i-25} and b_{j-38} in $/M.(m,n)$, the adversary's best strategy is $1+/M\rho(p,q)+\lambda M.(s,t)$. If $(p,q,\rho) \neq (1,1,/)$, the adversary uses the same strategy, and we get

$$\begin{aligned}/M_{i,j}.(m+25,n+38) \geq& 1+/M\rho(p+25,q+38)+\lambda M.(s,t)\\ \geq& 1+/M\rho(p,q)+63+\lambda M.(s,t) \geq /M.(m,n)+63\end{aligned}$$

by the induction hypothesis. If $(p, q, \rho) = (1, 1, /)$, then we know that $i \geq 27$ and $j = 39$, and the adversary can use the simple strategy, yielding

$$/M_{i,j}.(m{+}25, n{+}38) \geq 1{+}/M.(25, 39){+}.M.(m, n{-}1) = 1{+}63{+}.M.(m, n{-}1) \geq 63{+}/M.(m, n).$$

The last inequality is due to Lemma 5.

Therefore the adversary can always find a strategy resulting the value not smaller than $/M.(m, n) + 63$, no matter what the first comparison is. This completes the proof of *Part (b).*

Part (c). The proof for cases where $(i \leq m,\ j \leq n)$, $(i \geq m + 1,\ j \leq n)$, and $(i \leq m,\ j \geq n + 1)$ is similar with *Part (a).*

Similar with the above argument, if $i \geq m + 1$ and $j \geq n + 1$, we only need to investigate the following cases:

If $i \leq 25$ and $j \geq 39$, or $i \geq 27$ and $j \leq 38$, the adversary uses the complex strategy with a_{26} in both subproblems. This leads to

$$\backslash M_{i,j}.(m{+}25, n{+}38) \geq 1{+}\backslash M/(26, 38){+}\backslash M.(m, n) = 1{+}62{+}\backslash M.(m, n) = \backslash M.(m, n){+}63.$$

If $i = 26$ and $j \leq 38$, since $i \geq m + 1$ and $j \geq n + 1$, then $m \leq 25$ and $n \leq 37$ and these cases have been checked as starting values.

If $i \geq 26$ and $j \geq 39$: assume if we compare a_{i-25} and b_{j-38} in $\backslash M.(m, n)$, the adversary's best strategy is $1 + \backslash M\rho(p, q) + \lambda M.(s, t)$. If $(p, q, \rho) \neq (1, 1, \backslash)$ or $(2, 1, /)$, the adversary uses the same strategy, yielding

$$\begin{aligned} \backslash M_{i,j}.(m + 25, n + 38) \geq & 1 + \backslash M\rho(p + 25, q + 38) + \lambda M.(s, t) \\ \geq & 1 + \backslash M\rho(p, q) + 63 + \lambda M.(s, t) \geq \backslash M.(m, n) + 63 \end{aligned}$$

by the induction hypothesis. If $(p, q, \rho) = (1, 1, \backslash)$, we have $i = 26$ and $j \geq 40$. The adversary claims $a_i < b_j$ and follows the simple strategy, yielding

$$\backslash M_{i,j}.(m{+}25, n{+}38) \geq 1{+}\backslash M.(26, 38){+}.M.(m{-}1, n) = 1{+}63{+}.M.(m{-}1, n) \geq 63{+}\backslash M.(m, n)$$

by using Lemma 5. If $(p, q, \rho) = (2, 1, /)$, we have $i \geq 28$ and $j = 39$ or $i = 26$ and $j > 39$, since the case where $i = 26$ and $j > 39$ has already been considered, we only need to investigate the case where $i \geq 28$ and $j = 39$. Notice that $j \geq n + 1$, i.e. $n \leq 38$, hence $m \geq 50$. If $i > 28$, the adversary claims $a_i > b_j$ and follows the complex strategy with a_{28} in both subproblems. This leads to

$$\begin{aligned} \backslash M_{i,j}.(m + 25, n + 38) \geq & \, 1 + \backslash M/(28, 39) + \backslash M.(m - 2, n - 1) \\ = & \, 1 + 66 + \backslash M.(m - 2, n - 1) \\ \geq & \, 66 + \backslash M.(m - 1, n - 1) \\ \geq & \, 63 + 1 + \backslash M/(2, 1) + \backslash M.(m - 1, n - 1) \\ = & \, 63 + \backslash M_{i-25,1}.(m, n) \\ \geq & \, \backslash M.(m, n) + 63. \end{aligned}$$

The second inequality is according to Lemma 7 and the second equality is the assumption of the best strategy for the adversary. If $i = 28$ and $j = 39$, we have $m \leq 27$ and $n \leq 38$, which have been checked as starting values.

Therefore the adversary can always find a strategy resulting the value not smaller than $\backslash M.(m, n) + 63$. This completes the proof of *Part (c).*

Part (d), (e), (f). Due to the space constraint, we put the proofs of Part (d), Part (e) and Part (f) in Appendix B. ◀

Now, we are ready to prove Theorem 1.

Proof. The small cases $1 \le m \le 25$ and $1 \le n \le 38$ are given in [19]. Given any pair (m, n) satisfying $m \le n \le \frac{38}{25}m$, let $m = 25p + s$, $n = 38q + t$ where $0 < s \le 25$, $0 < t \le 38$, and observe that $m \ge 25q + \lceil \frac{25}{38}t \rceil$, thus

$$\overline{.M.}(m,n) = \overline{.M.}(25p+s, 38q+t) \le \overline{.M.}(25q + \lceil \tfrac{25}{38}t \rceil, 38q+t) \le \overline{.M.}(\lceil \tfrac{25}{38}t \rceil, t) = 0.$$

The first inequality is due to Lemma 6 and the second one is due to Theorem 8. ◀

5 Limitations of Knuth's adversary methods

In this section, we prove Theorem 2, which shows Knuth's adversary methods can not provide lower bounds beyond $\alpha(m) \ge 9\lceil m/5 \rceil$. Actually, we prove a stronger result:

▶ **Theorem 9.** *$.M.(5k, 9k+12t) \le 14k + 11t - 2$, for $k, t \ge 0$ and $t + k \ge 1$.*

With this theorem, Theorem 2 is obvious, since if $n \ge 9\lceil m/5 \rceil$, $\overline{.M.}(m,n) \ge \overline{.M.}(m, 9\lceil m/5 \rceil) \ge 1$.

Proof. The proof is by induction on k and t. We verify the case $k \le 10$ first: when $t \ge k/10 + 2/5$, we have

$$.M.(5k, 9k+12t) \le M(5k, 10k+12t-k) \le M_{bm}(5k, 10k+12t-k) \le 14k + 11t - 2.$$

When $t < k/10 + 2/5$, these finite cases can be checked in [19].

Now suppose $k \ge 11$ and we have already proven this theorem for any (k', t') satisfying $k' < k$, or $k' = k$ and $t' < t$. Since $.M.(m,n) = min_{i,j}.M_{i,j}.(m,n) \le .M_{50,79}.(m,n)$, thus it's enough to show $.M_{50,79}.(5k, 9k+12t) \le 14k + 11t - 2$. In other word, an algorithm which begins by comparing a_{50} with b_{79} can "beat" the adversary. We'll prove it by enumerating the adversary's best strategy.

Case(a). The adversary claims $a_{50} < b_{79}$ and follows three possible strategies.

(i) The adversary uses the simple strategy, then

$$.M_{50,79}.(5k, 9k+12t) = 1 + .M.(50+x, 78-y) + .M.(5k-50-x, 9k+12t-78+y),$$

where $x, y \ge 0$. Thus it's sufficient to show

$$\overline{.M.}(50+x, 78-y) + \overline{.M.}(5k-50-x, 9k+12t-78+y) \ge \overline{.M.}(5k-50, 9k-78+12t) \ge t+2.$$

The first inequality is according to Lemma 6 and the second one is by the induction hypothesis.

(ii) The adversary uses the complex strategy, with a_{51+x} in both subproblems.

$$\begin{aligned} &.M_{50,79}.(5k, 9k+12t) \\ &\qquad = 1 + .M/(51+x, 78-y) + \backslash M.(5k-50-x, 9k+12t-78+y) \\ &\qquad \le 1 + .M.(51+x, 78-y) + .M.(5k-50-x, 9k+12t-78+y), \end{aligned}$$

where $x, y \ge 0$. Thus it's equivalent to show

$$\begin{aligned} &\overline{.M.}(51+x, 78-y) + \overline{.M.}(5k-50, 9k+12t-78+y) - 1 \\ &\qquad\qquad \ge \overline{.M.}(5k-50, 9k-78+12t) - 1 \ge t+1. \end{aligned}$$

(iii) The adversary uses the complex strategy with b_{78-y} in both subproblems.

$$\begin{aligned}&.M_{50,79}.(5k{,}9k+12t)\\ &\qquad\qquad = 1 + .M\backslash(50+x, 78-y) + /M.(5k-50-x, 9k+12t-77+y)\\ &\qquad\qquad \leq 1 + .M.(50+x, 78-y) + .M.(5k-50-x, 9k+12t-77+y),\end{aligned}$$

where $x, y \geq 0$. Thus it's equivalent to show

$$\begin{aligned}&\overline{.M.}(50+x, 78-y) + \overline{.M.}(5k-50-x, 9k+12t-77+y) - 1\\ &\qquad\qquad\qquad \geq \overline{.M.}(5k-50, 9k-78+12t) - 1 \geq t+1.\end{aligned}$$

Case(b). The adversary claims $a_{50} > b_{79}$ and follows three possible strategies.

(i) The adversary uses the simple strategy, then

$$.M_{50,79}.(5k, 9k+12t) = 1 + .M.(49-x, 79+y) + .M.(5k-50+1+x, 9k+12t-79-y),$$

where $x, y \geq 0$. Thus it's sufficient to show

$$\overline{.M.}(49-x, 79+y) + \overline{.M.}(5k-50+1+x, 9k+12t-79-y) \geq t+2.$$

Let $5p \leq x \leq 5p+4$ and $12q-10 \leq y \leq 12q+1$, then we claim that $\overline{.M.}(49-x, 79+y) \geq p+q+1$. If $q \leq 2$, these finite cases can be checked in [19]. Otherwise ($q \geq 3$), then $79+y > 2 \times (49-5p)$, and $.M.(49-5p, 81+12(q-1)) \leq M_{bm}(49-5p, 81+12(q-1)) \leq 127+11(q-1)-6p$. Therefore $\overline{.M.}(49-x, 79+y) \geq \overline{.M.}(49-5p, 81+12(q-1)) \geq 1+p+q$ due to Lemma 6.
Since $\overline{.M.}(49-x, 79+y) \geq p+q+1$, if $p+q \geq t+1$, we've done. If $p+q \leq t$, according to Lemma 6 and the induction hypothesis, we have

$$\begin{aligned}&\overline{.M.}(5k-50+1+x, 9k+12t-79-y)\\ &\qquad\qquad \geq \overline{.M.}(5k-50+5p+5, 9k-90+9p+9+12(t-q-p))\\ &\qquad\qquad \geq t-p-q+1.\end{aligned}$$

Thus $\overline{.M.}(49-x, 79+y) + \overline{.M.}(5k-50+1+x, 9k+12t-79-y) \geq t+2$.

(ii) The adversary uses the complex strategy with a_{49-x} in both subproblems, then

$$\begin{aligned}&.M_{50,79}.(5k, 9k+12t)\\ &\qquad = 1 + .M/(49-x, 79+y) + \backslash M.(5k-50+2+x, 9k+12t-79-y)\\ &\qquad \leq 2 + .M.(49-x, 79+y) + 1 + .M.(5k-50+1+x, 9k+12t-79-y)\\ &\qquad \leq 14k+11t-2,\end{aligned}$$

where $x, y \geq 0$.

(iii) The adversary uses the complex strategy with b_{80+y} in both subproblems, then

$$\begin{aligned}&.M_{50,79}.(5k, 9k+12t)\\ &\qquad = 1 + .M\backslash(49-x, 80+y) + /M.(5k-50+1+x, 9k+12t-79-y)\\ &\qquad \leq 2 + .M.(49-x, 79+y) + .M.(5k-50+1+x, 9k+12t-79-y)\\ &\qquad \leq 14k+11t-2,\end{aligned}$$

where $x, y \geq 0$.

◀

6 Upper bounds for $\alpha(m)$

In this section, we give better upper bounds for $\alpha(m)$ by proposing a simple procedure. This procedure only involves the first two elements in each A and B and can be viewed as a modification of binary merge.

Algorithm 1 Modified Binary Merge

```
Compare $a_1$ and $b_2$
if $a_1 > b_2$ then
  merge $(m, n-2)$.
else
  compare $a_2$ and $b_2$
  if $a_2 > b_2$ then
    compare $a_1$ and $b_1$, then merge $(m-1, n-2)$.
  else
    compare $a_2$ and $b_1$
    if $a_2 > b_1$ then
      compare $a_1$ and $b_1$, then merge $(m-2, n-1)$ .
    else
      merge $(m-2, n)$.
    end if
  end if
end if
```

It is easy to see that this procedure induces the following recurrence relation:

$$M(m,n) \leq \max\{M(m,n-2)+1, M(m-1,n-2)+3, M(m-2,n)+3, M(m-2,n-1)+4\}.$$

In the following, we'll use the induction to give better upper bounds for $n \in [2m-2, 3m]$. The following proofs are very similar, but we give all the details for sake of completeness.

▶ **Theorem 10.** *$M(m, 2m+2k) \leq 3m+k-2$, for $m \geq 3$ and $k \geq -1$.*

Proof. We induce on k and m. The case for $k = -1$ just follows tape merge algorithm. The case for $m = 3$ are given by Hwang [15] and Murphy [25].

Now suppose that $m \geq 4$ and $k \geq 0$, and the claim has already been proven for any (m', k') satisfying $m' + k' \leq m + k - 1$. According to the procedure, we have

$$\begin{aligned} &M(m,2m+2k) \\ &\quad\leq \max\{M(m, 2m+2(k-1))+1, M(m-1, 2m+2k-2)+3, \\ &\qquad\qquad M(m-2, 2m+2k)+3, M(m-2, 2m+2k-1)+4\} \end{aligned}$$

$\leq \max\{3m+k-2$ (the induction hypothesis), $3(m-1)+k-2+3$ (the induction hypothesis), $3(m-2)+1+k+3$ (binary merge), $3(m-2)+k+4$ (binary merge)$\} \leq 3m+k-2$. ◀

▶ **Theorem 11.** *$M(m, 2m+2k-1) \leq 3m+k-3$, for $m \geq 5$ and $k \geq -1$.*

Proof. We induce on k and m. The case for $k = -1$ just follows tape merge algorithm. The case for $m = 5$ are given by Mönting [26].

Now suppose that $m \geq 6$ and $k \geq 0$, and the claim has already been proven for any (m', k') satisfying $m' + k' \leq m + k - 1$. According to the procedure, we have

$$\begin{aligned} &M(m,2m + 2k - 1) \\ &\quad\leq \max\{M(m, 2m + 2(k - 1) - 1) + 1, M(m - 1, 2m + 2k - 1 - 2) + 3, \\ &\qquad\qquad M(m - 2, 2m + 2k - 1) + 3, M(m - 2, 2m + 2k - 2) + 4\} \end{aligned}$$

$\leq \max\{3m+k-3$ (the induction hypothesis), $3(m-1)+k-3+3$ (the induction hypothesis), $3(m - 2) + k + 3$ (binary merge), $3(m - 2) + k - 1 + 4$ (Theorem 10)$\} \leq 3m + k - 3$. ◀

▶ **Theorem 12.** *$M(m, 2m - 2) \leq 3m - 4$, for $m \geq 7$.*

Proof. We induce on m. The case for $m = 7$ has been verified by Smith and Lang[27]. Now suppose that $m \geq 8$ and the claim has already been proven for $m - 1$. According to S, we have

$$\begin{aligned} &M(m,2m - 2) \\ &\quad\leq \max\{M(m, 2m - 4) + 1, M(m - 1, 2m - 4) + 3, \\ &\qquad\qquad M(m - 2, 2m - 2) + 3, M(m - 2, 2m - 3) + 4\} \end{aligned}$$

$\leq \max\{3m - 4$(tape merge), $3(m - 1) - 4 + 3$ (the induction hypothesis), $3(m - 2) - 1 + 3$ (Theorem 10), $3(m - 2) - 2 + 4$ (Theorem 11)$\} \leq 3m - 4$. ◀

▶ **Theorem 13.** *$M(m, 2m) \leq 3m - 3$, for $m \geq 10$.*

Proof. We do the induction on m. Smith and Lang[27] have verified the case for $m = 10$. Now suppose that $m \geq 11$ and the claim has already been proven for $m - 1$. According to the procedure, we have

$$\begin{aligned} &M(m,2m) \\ &\quad\leq \max\{M(m, 2m - 2) + 1, M(m - 1, 2m - 2) + 3, \\ &\qquad\qquad M(m - 2, 2m) + 3, M(m - 2, 2m - 1) + 4\} \end{aligned}$$

$\leq \max\{3m - 3$ (Theorem 12), $3(m - 1) - 3 + 3$ (the induction hypothesis), $3(m - 2) + 3$ (Theorem 10), $3(m - 2) - 1 + 4$ (Theorem 11)$\} \leq 3m - 3$. ◀

Finally, we put together the above theorems to get Theorem 3.

As we can see in the proofs, if better basic cases can be provided, we can get better upper bounds by using this procedure. However, there is a barrier of this approach: if we want to show $\alpha(m) \leq 2m - k$ or $M(m, 2m - k + 1) < 3m - k$, it's necessary to obtain $\alpha(m - 1) \leq 2(m - 1) - k$ or $M(m - 1, 2m - 2 - k + 1) < 3m - 3 - k$ at first, thus it is impossible to show $\alpha(m) \leq 2m - \omega(1)$ via this approach.

7 Conclusion

In this paper we improve the lower bounds for $\alpha(m)$ from $\lfloor \frac{3}{2}m \rfloor + 1$ to $\lfloor \frac{38}{25}m \rfloor$ via Knuth's adversary methods. We also show that it is impossible to get $\alpha(m) \geq 9\lceil m/5 \rceil \approx \frac{9}{5}m$ for any m by using this methods. We then design an algorithm which saves at least one comparison compared to binary merge for $2m - 2 \leq n \leq 3m$. Specially, for the case $M(m, 2m - 2)$, our algorithm uses one comparison less than tape merge or binary merge, which means we can improve the upper bounds of $\alpha(m)$ by 1. We wonder whether there exists a universal efficient algorithm to give significantly better upper bounds for $M(m, n)$ in the case $n \leq 2m$, or maybe it's intrinsically hard to compute M functions since there doesn't exist general patterns or underlying structures in the corresponding decision trees.

Besides that, we are also curious about the following conjectures proposed by Knuth [18]:

▶ **Conjecture 14.** $M(m+1, n+1) \geq 2 + M(m, n)$.

Via a similar proof with Lemma 7, the above conjecture implies the following conjecture which has been mentioned in Section 1.

▶ **Conjecture 15.** $M(m+1, n) \geq 1 + M(m, n) \geq M(m, n+1)$*, for* $m \leq n$.

In the attempt to prove these two conjectures, we introduced the notation $.M^{(k)}.(m, n)$. Roughly speaking, $.M^k.(m, n)$ is the adversary which can delay k steps to give the splitting strategy, and $.M.(m, n) = .M^0.(m, n) \leq .M^1.(m, n) \cdots \leq .M^{m+n-2}.(m, n) = M(m, n)$. In the case $k = 0$, it is exactly Lemma 16 in appendix A, but it becomes much harder even for $k = 1$.

References

1 Miklós Ajtai, Vitaly Feldman, Avinatan Hassidim, and Jelani Nelson. Sorting and selection with imprecise comparisons. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part I*, ICALP'09, pages 37–48, 2009.

2 Mark R. Brown and Robert E. Tarjan. A fast merging algorithm. *J. ACM*, 26(2):211–226, 1979.

3 Jean Cardinal and Samuel Fiorini. *On Generalized Comparison-Based Sorting Problems*, pages 164–175. Springer Berlin Heidelberg, 2013.

4 Yongxi Cheng, Xiaoming Sun, and Yiqun Lisa Yin. Searching monotone multi-dimensional arrays. *Discrete Mathematics*, 308(11):2213–2221, 2008.

5 C. Christen. Improving the bounds on optimal merging. In *Proc 19th IEEE conf on the foundations of computer science*, pages 259–266, 1978.

6 C. Christen. On the optimality of the straight merging algorithm, 1978.

7 W. Fernandez de la Vega, M. A. Frieze, and M. Santha. Average-case analysis of the merging algorithm of hwang and lin. *Algorithmica*, 22(4):483–489, 1998.

8 Wenceslas Fernandez de la Vega, Sampath Kannan, and Miklos Santha. Two probabilistic results on merging. *SIAM Journal on Computing*, 22(2):261–271, 1993.

9 Fănică Gavril. Merging with parallel processors. *Commun. ACM*, 18(10):588–591, 1975.

10 A. Gupta and A. Kumar. Sorting and selection with structured costs. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, page 416, 2001.

11 Bing-Chao Huang and Michael A. Langston. Practical in-place merging. *Commun. ACM*, 31(3):348–352, 1988.

12 Z. Huang, S. Kannan, and S. Khanna. Algorithms for the generalized sorting problem. In *Proceedings of the 52nd IEEE Symposium on Foundations of Computer Science*, pages 738–747, 2011.

13 F. K. Hwang and D. N. Deutsch. A class of merging algorithms. *J. ACM*, 20(1):148–159, 1973.

14 Frank Hwang and Shen Lin. A simple algorithm for merging two disjoint linearly ordered sets. *SIAM Journal on Computing*, 1(1):31–39, 1972.

15 Frank K. Hwang. Optimal merging of 3 elements with n elements. *SIAM Journal on Computing*, 9(2):298–320, 1980.

16 Frank K. Hwang and Shen Lin. Optimal merging of 2 elements with n elements. *Acta Informatica*, 1(2):145–158, 1971.

17 Sampath Kannan and Sanjeev Khanna. Selection with monotone comparison costs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'03, pages 10–17, 2003.

18 Donald Knuth. The art of computer programming. *Sorting and searching*, 3:197–207, 1999.

19 Qian Li, Xiaoming Sun, and Jialin Zhang. Tables of $\lambda m \rho$ and the program, 2016. URL: `http://theory.ict.ac.cn/liqian`.

20 N. Linial and M. Saks. Searching ordered structures. *Journal of Algorithms*, 6(1):86–103, 1985.

21 Nathan Linial. The information-theoretic bound is good for merging. *SIAM Journal on Computing*, 13(4):795–801, 1984.

22 G. K. Manacher, T. D. Bui, and T. Mai. Optimum combinations of sorting and merging. *J. ACM*, 36(2):290–334, 1989.

23 Glenn K. Manacher. Significant improvements to the hwang-lin merging algorithm. *J. ACM*, 26(3):434–440, 1979.

24 Paul E. Murphy and Marvin C. Paull. Minimum comparison merging of sets of approximately equal size. *Information and Control*, 42(1):87–96, 1979.

25 P. E. Murphy. A problem in optimal meging. Technical Report DCS-TR-69, Rutgers University, Dept. of Computer Sciences, 1978.

26 Jürgen Schulte Mönting. Merging of 4 or 5 elements with n elements. *Theoretical Computer Science*, 14(1):19–37, 1981.

27 Warren D. Smith and Kevin J. Lang. Values of the merging function and algorithm design as a game, 1994.

28 Paul K. Stockmeyer and F. Frances Yao. On the optimality of linear merge. *SIAM Journal on Computing*, 9(1):85–90, 1980.

29 R. Michael. Tanner. Minimean merging and sorting: An algorithm. *SIAM Journal on Computing*, 7(1):18–38, 1978.

30 Mai Thanh, V. S. Alagar, and T. D. Bui. Optimal expected-time algorithms for merging. *Journal of Algorithms*, 7(3):341–357, 1986.

31 Mai Thanh and T. D. Bui. An improvement of the binary merge algorithm. *BIT Numerical Mathematics*, 22(4):454–462, 1982.

A Proof of Lemma 6

We show the following lemma first:

▶ **Lemma 16.** $.M.(m+1, n+1) \geq .M.(m, n) + 2$, *for any* $m, n \geq 0$ *and* $m + n \geq 1$.

Proof. The proof is by induction on m and n. The starting values for $m, n \leq 3$ can be easily checked in [18]. Now suppose the theorem holds for any m', n' satisfying $m' \leq m$, $n' \leq n$ and $m' + n' < m + n$, we then prove the case (m, n). Note that our task is to design a strategy for the adversary for $(m+1, n+1)$.

Suppose an algorithm begins by comparing a_i and b_j, where $i \leq m$, $j = n + 1$. The adversary claims that $a_i < b_j$, and follows the simple strategy, yielding

$$.M_{i,j}.(m+1, n+1) \geq 1 + .M.(m, n) + .M.(1, 1) \geq .M.(m, n) + 2\,.$$

If $i = m + 1$ and $j \leq n$, the adversary claims that $a_i > b_j$, and uses the simple strategy. This leads to

$$.M_{i,j}.(m+1, n+1) \geq 1 + .M.(m, n) + .M.(1, 1) \geq .M.(m, n) + 2\,.$$

If $i \leq m$ and $j \leq n$, assume that if we compare a_i and b_j in $.M.(m,n)$, the adversary's best strategy is $1 + .M\rho(p,q) + \lambda M.(s,t)$, where $\rho, \lambda \in \{., /, \backslash\}$ and $s+t \geq 1$ if $\lambda = .$ and $s,t \geq 1$ if $\lambda = \{/, \backslash\}$. Then the adversary uses the same strategy here, and we get

$$\begin{aligned} .M_{i,j}.(m+1,n+1) &\geq 1 + .M\rho(p,q) + \lambda M.(s+1,t+1) \\ &\geq 1 + .M\rho(p,q) + \lambda M.(s,t) + 2 = .M.(m,n) + 2 \end{aligned}$$

by applying the induction hypothesis and Lemma 7.

If $i = m+1$ and $j = n+1$, we can handle this case as well by reversing the order of the elements in A and B.

Therefore adversary can always find a strategy which is not smaller than $.M.(m,n)+2$, no matter what the first comparison is. This completes our proof. ◀

Now we can give the proof of Lemma 6

Proof. We induce on m and n. The case for $1 \leq m+n \leq 10$ are given in [18]. Now suppose that $m+n \geq 11$ and the lemma is already established for any (m', n') satisfying $m'+n' < m+n$.

Part (a).

If $m = 1$, we have $.M.(1, n+1) \leq .M_{1,n+1}.(1, n+1) = \max\{1, 1+.M.(1,n)\} = 1+.M.(1,n)$. If $m \geq 2$, then $.M_{m,n+1}.(m, n+1) = \max\{.M.(m-1,n+1), .M.(m,n)\} + 1$, and by the induction we know $.M.(m,n) \geq .M.(m-1,n) + 1 \geq .M.(m-1,n+1)$, thus

$$.M.(m,n+1) \leq .M_{m,n+1}.(m,n+1) \leq \max\{.M.(m-1,n+1), .M.(m,n)\}+1 = .M.(m,n)+1.$$

Part (b).

When $m = n$, $.M.(m+1,m) \geq .M.(m,m-1) + 2 \geq .M.(m-1,m-2) + 4 \geq \cdots \geq .M.(2,1) + 2m - 2 = 2m$ according to Lemma 16, thus $.M.(m+1,m) \geq .M.(m,m)+1$, since $.M.(m,m) \leq 2m-1$. When $m < n$, we have

$$.M.(m+1,n) \geq .M.(m,n-1) + 2 \geq .M.(m,n) + 1.$$

The first inequality is due to Lemma 16, and the second one is by the induction hypothesis. ◀

B Proof of Theorem 8

Proof.

Part (d). The proof for cases where $i \leq 25$ and $j > 38$, $i > 25$ and $j \leq 38$, and $i > 25$ and $j > 38$ is similar with *Part (b)*.

If $i \leq 25$ and $j \leq 38$, reserving the order of the elements maps this case to the above cases, thus we can handle this case as well by symmetry.

Therefore the adversary can always find a strategy which is not smaller than $/M\backslash(m,n)+63$, no matter which the first comparison is. So *Part (d)* is true.

Part (e). The adversary's strategies for cases where $(i \leq m,\ j \leq n)$, $(i > m,\ j \leq n)$, and $(i \leq m,\ j > n)$ are similar with *Part (d)*.

If $i \geq m+1$ and $j \geq n+1$, then $i \leq 25$ and $j \leq 38$ cannot happen simultaneously, so we only need to investigate the following cases:

If $i \leq 25$ and $j > 38$, or $i \geq 27$ and $j \leq 38$, the adversary uses the complex strategy with a_{26} in both subproblems. This leads to

$$\backslash M_{i,j}\backslash(m+25, n+38) \geq 1 + \backslash M/(26,38) + \backslash M\backslash(m,n) = \backslash M\backslash(m,n) + 63.$$

If $i = 26$ and $j \le 38$, then $m \le 25$ and $n \le 38$, which have been checked as starting values.

If $i \ge 26$ and $j \ge 39$: assume if we compare a_{i-25} and b_{j-38} in $\backslash M\backslash(m,n)$, the adversary's best strategy is $1 + \backslash M\rho(p,q) + \lambda M\backslash(s,t)$. If $(p,q,\rho) \neq (1,1,\backslash)$ or $(2,1,/)$, the adversary uses the same strategy, yielding

$$\backslash M_{i,j}\backslash(m+25,n+38) \ge 1 + \backslash M\rho(p+25,q+38) + \lambda M\backslash(s,t) \ge \backslash M\backslash(m,n) + 63$$

by the induction hypothesis. If $(p,q,\rho) = (1,1,\backslash)$, we have $i = 26$ and $j > 39$. The adversary claims $a_i < b_j$ and uses the simple strategy, leading to

$$\backslash M_{i,j}\backslash(m{+}25,n{+}38) \ge 1{+}\backslash M.(26,38){+}.M\backslash(m{-}1,n) = 64{+}.M\backslash(m{-}1,n) \ge 63{+}\backslash M\backslash(m,n)$$

by using Lemma 5.

If $(p,q,\rho) = (2,1,/)$, the case where $i \ge 28$ and $j = 39$ is the only unconsidered case. If we compare a_2 with b_1 in $\backslash M\backslash(m,n)$, and the adversary claims $a_2 > b_1$, then the best strategy must be $1 + \backslash M.(1,1) + .M\backslash(m-1,n-1)$, otherwise the adversary claims $a_2 < b_1$, and the best strategy must be $1 + .M\backslash(m-2,n)$. So we get $\backslash M\backslash(m,n) \le \backslash M_{2,1}\backslash(m,n) = \max\{1{+}.M\backslash(m{-}2,n), 2{+}.M\backslash(m{-}1,n{-}1)\}$. If $.M\backslash(m{-}2,n) \ge 1{+}.M\backslash(m{-}1,n{-}1)$, then $.M\backslash(m-2,n)+1 \ge \backslash M\backslash(m,n)$, and the adversary splits the problem $\backslash M\backslash(m+25,n+38)$ into two independent subproblems $\backslash M.(27,38)$ and $.M\backslash(m-2,n)$ before the algorithm begins, and this leads to

$$\backslash M\backslash(m+25,n+38) \ge \backslash M.(27,38) + .M\backslash(m-2,n) \ge 63 + \backslash M\backslash(m,n)$$

Otherwise $(.M\backslash(m-2,n) < 1+.M\backslash(m-1,n-1))$, then $2+.M\backslash(m-1,n-1) \ge \backslash M\backslash(m,n)$ and the adversary claims $a_i > b_j$ and uses the simple strategy, yielding

$$\backslash M\backslash(m+25,n+38) \ge \backslash M.(26,39) + .M\backslash(m-1,n-1) + 1 \ge \backslash M\backslash(m,n) + 63.$$

Therefore the adversary can always find a strategy resulting the value not smaller than $\backslash M\backslash(m,n) + 63$. This completes the proof of Part (e).

Part (f). If $n > 50$, we can assume $j \ge \lfloor \frac{38+n}{2} \rfloor \ge 44$ by symmetry. If $i \le 25$, the adversary claims $a_i < b_j$ and uses the complex strategy with a_{26} in both subproblems. This leads to

$$\backslash M/(m+25,n+38) \ge 1 + \backslash M/(26,38) + \backslash M/(m,n) = 63 + \backslash M/(m,n).$$

If $i \ge 26$: assume that if we compare a_{i-25} and b_{j-38} in $\backslash M\backslash(m,n)$, the adversary's best strategy is $1 + \backslash M\rho(p,q) + \lambda M/(s,t)$. If $(p,q,\rho) \neq (1,1,\backslash),(2,1,/)$, adversary uses the same strategy, yielding

$$\backslash M_{i,j}/(m+25,n+38) \ge 1 + \backslash M\rho(p+25,q+38) + \lambda M/(s,t) \ge \backslash M/(m,n) + 63$$

by using the induction hypothesis. If $(p,q,\rho) = (1,1,\backslash)$, we have $i = 26$ and $j \ge 40$. The adversary claims $a_i < b_j$ and use the simple strategy, yielding

$$\backslash M_{i,j}/(m{+}25,n{+}38) \ge 1{+}\backslash M.(26,28){+}.M/(m{-}1,n) = 64{+}.M/(m{-}1,n) \ge 63{+}\backslash M/(m,n)$$

by using Lemma 5. If $(p,q,\rho) = (2,1,/)$, we have $i \ge 28$ and $j = 39$, violating the assumption $j \ge 44$.

If $n \le 50$, then $m > 50$ and we can assume $i \ge \lfloor \frac{25+m}{2} \rfloor \ge 38$ by symmetry. If $j \le 38$, the adversary claims $a_i > b_j$, and uses the complex strategy with a_{26} in both subproblems, yielding

$$\backslash M/(m+25, n+38) \ge 1 + \backslash M/(26, 38) + \backslash M/(m, n) \ge 63 + \backslash M/(m, n).$$

If $j \ge 39$: assume if we compare a_{i-25} and b_{j-38} in $\backslash M\backslash(m, n)$, the adversary's best strategy is $1 + \backslash M\rho(p, q) + \lambda M/(s, t)$. If $(p, q, \rho) \ne (1, 1, \backslash)$ or $(2, 1, /)$, then adversary uses the same strategy, thus

$$\backslash M_{i,j}/(m+25, n+38) \ge 1 + \backslash M\rho(p+25, q+38) + \lambda M/(s, t) \ge \backslash M/(m, n) + 63$$

by using the induction hypothesis. If $(p, q, \rho) = (2, 1, /)$, we have $j = 39$. Similar with the argument in Part (e), we get $\backslash M/(m, n) \le \backslash M_{2,1}/(m, n) = \max\{1 + .M/(m-2, n), 2 + .M/(m-1, n-1)\}$. If $.M/(m-2, n) \ge 1 + .M/(m-1, n-1)$, the adversary splits the problem $\backslash M/(m+25, n+38)$ into two independent subproblems $\backslash M.(27, 38)$ and $.M/(m-2, n)$ before the first comparison begins, and this leads to

$$\backslash M/(m+25, n+38) \ge \backslash M.(27, 38) + .M/(m-2, n) \ge 63 + \backslash M/(m, n).$$

Otherwise ($.M/(m-2, n) < 1 + .M/(m-1, n-1)$), then the adversary claims $a_i > b_j$, and uses the simple strategy, yielding

$$\begin{aligned} \backslash M/(m+25, n+38) &\ge \backslash M.(26, 39) + .M/(m-1, n-1) + 1 \ge 65 + .M/(m-1, n-1) \\ &\ge \backslash M/(m, n) + 63 \end{aligned}$$

If $(p, q, \rho) = (1, 1, \backslash)$, then $i = 26$, violating the assumption $i \ge 38$.

Therefore the adversary can always find a strategy resulting the value not smaller than $\backslash M/(m, n) + 63$. This completes the proof of Part (f). ◀