New Lower Bounds for Element Distinctness on a One-tape Turing Machine

Alejandro López-Ortiz
Department of Computer Science
University of Waterloo
Waterloo, Ont. N2L 3G1 Canada
e-mail: alopez-o@maytag.UWaterloo.ca

Abstract

It is shown that the Element Distinctness Problem (n numbers of $k \log n$ bits each, $k \geq 2$) on a one-tape Turing machine takes time proportional to almost the square of the size of the input. The proof holds for both deterministic and nondeterministic Turing machines. This proof improves the best known lower bound of $\Omega(n^2/\log n)$ for deterministic Turing machines and of $\Omega(n \log^2 n)$ [4] for nondeterministic Turing machines to $\Omega(n^2 \log n)$. The lower bound is generalized to the n-Element Distinctness problem; on inputs of size $N = nm$, with $1 \leq n < N/\log N$, it is shown to take time $\Omega(max\{Nn, Nm\})$. The proof makes use of Kolmogorov Complexity.

Keywords: Theory of Computation, Kolmogorov Complexity

1 Introduction

The study of lower bounds for any computational model is one of the most difficult problems in Computational Complexity. Recent developments in Kolmogorov Complexity have proven to be a useful tool for obtaining new bounds in Turing machines [8, 1, 7].

Among the problems for which no tight upper and lower bounds are known, we have Element Distinctness (ED) on a single tape Turing machine. The importance of this problem is partially derived from its close relationship to sorting. Given that in most other computational models the complexity of sorting and ED match, it is then a natural question whether this holds in more restricted models of computation.

This paper presents a $\Omega(n^2 \log n)$ bound for the Element Distinctness problem on n numbers of $O(\log n)$ bits each on a one-tape Turing machine. This proof improves on the previously known lower bounds for ED, which is inherited from a slightly more powerful computational model [4]. The proof of this lower bound uses crossing sequences and Kolmogorov Complexity [7].

A natural generalization of ED is the n-Element Distinctness problem, in which n-numbers of m-bits each are to be distinguished from each other. As a consequence of the newly proven lower bound, we can prove that the n-Element Distinctness problem takes time $\Omega(\max\{Nn, Nm\})$ where $N = nm$ is the input size.

These proofs hold for deterministic and nondeterministic Turing machines. Since the complement of Element Distinctness is in NTIME($n \log^2 n$) time, this results separates NTIME($N^2/\log N$) from its complement for single-tape machines.

2 Preliminaries

Definition 1 *A single-tape Turing machine is a standard TM (as defined, say in [3]), consisting of a finite control and a single two-way infinite tape with one read/write head, where input, intermediate computations and output are to be written. Whatever is left in the tape after the TM halts is said to be the output. Furthermore, the machine may halt in an accepting or rejecting state [3].*

Definition 2 *A* crossing sequence *associated with a cell boundary is the sequence of states at which the finite control was when the head crossed the boundary between the cells.*

Notice that it is possible to modify any TM that computes a decision problem as to make it leave a unique identification of its decision (accept/reject) on all nonnull crossing sequences with at most a constant factor penalty in computing time. We assume our TM's are so modified.

Moreover, the sum of the length of all (or some of) the crossing sequences corresponding to the computation on a particular input give a lower bound on the time taken by the Turing machine to accept or reject such an input string.

As it is usual for decision problems on nondeterministic TMs, the time taken on input of length n is the longest computation over all inputs of this size.

Definition 3 *[7] Given an enumeration of Turing machines, the Kolmogorov Complexity $K(x)$ of a binary string x is the shortest pair (M, u) such that M represents a Turing machine which on input u computes x as output.*

Definition 4 *A string x is called incompressible if $K(x) \geq |x| + c$, where c is a fix constant.*

Lemma 1 *For every n there exist $\Omega(2^n)$ incompressible strings of length n.*

3 Element Distinctness

Definition 5 *The input for an instance of* Element Distinctness *on a single tape Turing machine is a list of n numbers encoded in binary notation, with each number being of length $k \log n$ for fixed $k \geq 2$. The TM halts in an accepting state if all the numbers are distinct (corresponding to a* YES *answer) and halts in a rejecting state (or* NO *answer) otherwise.*

Notice that the size of the input is $N = kn \log n$. In this notation, quadratic time on the input size means $O(N^2) = O(n^2 \log^2 n)$.

It is not hard to show that sorting on the single tape Turing machine takes $\Theta(N^2)$ steps on the Turing machine for n numbers of $k \log n$ bits each ($k > 2$). This brings up the question of whether the comparison-model-equivalent decision problem Element Distinctness has quadratic complexity on the single-tape Turing machine.

While it has been shown that on the comparison model ED and sorting have the same time complexity, this argument does not carry over to the single tape TM since it may be possible that sorting requires extra data movements to write its output which may not be needed for the Element Distinctness decision problem. The following two theorems shed light on this question.

Theorem 1 *Let N be the size of the input. Then* Element Distinctness *takes time $O(N^2)$ on a single tape deterministic TM.*

Proof (sketch). An all-pairs comparison gives the desired upper bound. □

Theorem 2 *Let N be the size of the input. Then* Element Distinctness *takes time $\Omega(n^2 \log n) = \Omega(N^2/\log N)$ in the worst case on any single-tape Turing machine.*

Proof. In this proof we assume $k = 2$. A simple padding argument extends this proof for any constant k. We also assume that n is a multiple of the form $6k$.

We construct a class of input strings $\mathcal{I}$ for which any Turing Machine accepting ED takes time $\Omega(N^2/\log N)$.

Each input I in $\mathcal{I}$ is such that $I = XYZ$ and $|X| = |Y| = |Z|$. That is, each of X, Y, and Z contains $n/3$ numbers.

Let $X = x_0 \ldots x_{(n-1)/3}$, $Y = y_0 \ldots y_{(n-1)/3}$ and $Z = z_0 \ldots z_{(n-1)/3}$ be the n numbers forming the input sequence. The first (leftmost) $\log n$ bits of each y_i are set to 0 and the remaining $\log n$ bits of y_i are set to the binary representation of i. The first bit of x_i and z_i is set to 1, for $i = 0 \ldots n-1$. This ensures that no x_i or z_i is equal to y_j ($\forall i,j$) and that all the y_i's are pairwise distinct.

Given this arrangement, the answer to a particular instance of the Element Distinctness problem is NO if and only if $x_i = z_j$ for some i, j or $x_i = x_j$ or $z_i = z_j$ for $i \neq j$.

The remaining bits of the x_i's and z_j's are chosen so that when concatenated they form an incompressible string in the sense of Definition 4. For the purposes of this proof, the selection of the incompressible bits of X and Z are such that no two x_i's and/or z_j's are equal. That is, the answer to the decision problem on input XYZ is YES.

The class $\mathcal{I}$ contains exactly those strings that satisfy the above criteria.

Let $\mathcal{C}$ denote the crossing sequence for I at some cell boundary in the Y section. Also consider all the strings $T_1, T_2 \ldots$ of the same length as X such that $T_i YZ \in \mathcal{I}$ but T_i differs from X in at least one bit.

Now, given any such string T_i, it is possible to replace X by T_i and verify if the computation of the TM on input T_iYZ has the same crossing sequence $\mathcal{C}$ as I. Let $\mathcal{L}$ be the set of strings T_i that do have the same crossing sequence as X. Analogously, let $\mathcal{R}$ be the set of those T_i such that the input XYT_i has $\mathcal{C}$ as crossing sequence as well.

The sets $\mathcal{L}$ and $\mathcal{R}$ are recursively enumerable, which implies that X (and Z) can be reconstructed from the crossing sequence $\mathcal{C}$ and its index in the lexicographical numbering of $\mathcal{L}$ ($\mathcal{R}$).

Since X is an incompressible string, the length of any description of it has to be of length $(n/3)(2\log n - 1) - c$, in particular, $|\mathcal{C}| + |\text{index}(X, \mathcal{L})| \geq (n/3)(2\log n - 1)$.

Lemma 2 *Given an accepting crossing sequence $\mathcal{C}$, the smallest of the sets $\mathcal{L}$ and $\mathcal{R}$ has at most $(n^2/4)! \,/\, (n^2/4 - n/3)!$ elements.*

Corollary 1 *The index of X or Z in $\mathcal{L}$ or $\mathcal{R}$ is* $\log(\,(n^2/4)! \,/\, (n^2/4 - n/3)!\,)$ *bits long.*

Observation 1 *Note that* $\prod_{i=0}^{n/3-1}(n^2/4 - i) \leq (n^2/4)^{n/3}$ *and thus* $\log(\,(n^2/4)! \,/\, (n^2/4 - n/3)!\,) \leq (2/3)n(\log n - 1)$.

Clearly, once Lemma 2 has been proved, Corollary 1 and Observation 1 imply that $\log|\mathcal{C}| \geq n/3$ in order for X or Z to be incompressible. And since $\mathcal{C}$ is any crossing sequence in one of the $2n\log n/3$ cell boundaries of Y, the total time taken by the TM is $\Omega(n^2 \log n)$ as required.

PROOF (Lemma 2). From the definition of $\mathcal{L}$ and $\mathcal{R}$ it follows that for any $U \in \mathcal{L}$ and $V \in \mathcal{R}$ the string UYV also has $\mathcal{C}$ as crossing sequence. Since the TM accepts the string I, the crossing sequence uniquely identifies such decision. That is, for all U and V as above, the string UYV is also accepted. But UYV can only be accepted if no two numbers in U and V are the same for all U and V.

If no two numbers are the same in U and V this implies that the numbers (elements) appearing in the strings of U and those appearing in V form two disjoint subsets of $\{0, \ldots, n^2/2 - 1\}$. The smaller of the two subsets formed by U and V has then at most $n^2/4$ elements.

Out of a set of at most $n^2/4$ numbers we can construct at most $(n^2/4)!/(n^2/4 - n/3)!$ different strings U or V, which implies that the smallest of $\mathcal{L}$ or $\mathcal{R}$ is at most this large, as stated in Lemma 2.

This concludes the proof of Lemma 2 and of Theorem 2. □

Notice that to prove this lower bound we construct a set of strings which are hard cases for any Element Distinctness algorithm. I.e. the set of strings constructed requires $\Omega(n^2 \log n)$ time. We conjecture that the set proposed requires indeed time $\Theta(n^2 \log^2 n)$. There are sets of simpler construction which require time $\Theta(n^2 \log n)$. The proof for such set then uses a suitably adapted version of Lemma 2. For the sake of generality and with potential extensions in mind we have chosen to present the proof based on the slightly more complex set of strings.

Definition 6 *For a given N, the input for the n-*ELEMENT DISTINCTNESS *problem in a single tape Turing machine is a list of n numbers encoded in binary notation, with each number being of length m where $m = N/n$. The TM halts in an accepting state if all the numbers are distinct (corresponding to a* YES *answer) and halts in a rejecting state (or* NO *answer) otherwise.*

Notice that the 2-Element Distinctness problem is computationally equivalent to the complement of PALINDROMES. The following theorem, which was proven by Hennie [2], gave the first quadratic lower bound for a TM.

Theorem 3 PALINDROMES *recognition takes time $\Theta(N^2)$ in a single-tape Turing machine.*

It is easy to see that the proof techniques used by Hennie and those of Theorem 2 in this work carry over to any n in the n-Element Distinctness problem.

Theorem 4 *The n-Element Distinctness problem takes time $\Omega(\max\{Nn, Nm\})$ for $1 \leq n < N/logN$ and $N = nm$.*

PROOF. First, let $I = xYz$ be an input string to the n-Element Distinctness problem such that x and z are numbers (elements) m bits long with the first bit set to 1 and Y is a string of $n-2$ distinct numbers (elements) each prefixed by 0.

Note that I is then in ED if and only if xy is not in PALINDROMES. Then by Hennie's theorem the length of the crossing sequences in the Y part is $O(m)$. Since the string Y is of length $(n-2)m = O(N)$, we have that this instance takes time $\Omega(Nm)$.

Now consider an input string $I = XYZ$ as in theorem 2, where $|X| = |Y| = |Z|$, and X and Z are incompressible strings of size $mn/3$. It then follows that for $m > logN$ the smallest of the corresponding sets $\mathcal{L}$ and $\mathcal{R}$ has $(2^{m-1})!/(2^{m-1} - n/3)!$ elements. Thus the length of the crossing sequence is $\mathcal{C} = mn/3 - nm/3 - n/3 = O(n)$, and since Y is of size $nm/3 = \Theta(N)$ this implies a lower bound of $\Omega(Nn)$.

Finally it follows trivially that the largest of the two bounds is also a lower bound, which concludes the proof of Theorem 4. □

4 Acknowledgments

I would like to thank Prabhakar Ragde, Ming Li, and Paul Vitanyi for helpful discussions on the content and comments on the presentation of this paper.

References

[1] M. Dietzfelbinger, W. Maass, and G. Schnitger, The Complexity of Matrix Transposition on One-Tape Off-line Turing Machines, Theoretical Computer Science, 1991, 113-129.

[2] F.C. Hennie, One-Tape, Off-Line Turing Machine Computations, Information and Control, 8 (1965), 553-578.

[3] J. Hopcroft, J. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, 1979.

[4] B. Kalyanasundaram, G. Schnitger, Communication Complexity and Lower Bounds for Sequential Machines Buchmann, Ganzinger, Paul (eds.): Informatik, Festschrift zum 60. Geburtstag von Professor Hotz, Teubner-Texte zur Informatik Band 1, Teubner, Stuttgart, 1992, 253–268

[7] M. Li, P. Vitanyi, Kolmogorov Complexity and its Applications, editor, Handbook of Theoretical Computer Science, J. van Leeuwen, MIT Press, 1990.

[8] W.J. Paul, Kolmogorov Complexity and Lower Bounds, Proc. 2nd International Conference on Foundations of Computation Theory, Akademie-Verlag, Berlin, 1979, 325-334.