Technische Universität Ilmenau
Institut für Mathematik

Preprint No. M 09/16

An Ω (n log n) lower bound for computing the sum of even-ranked elements

Mörig, Marc; Rautenbach, Dieter; Smid, Michiel; Tusch, Jan

April 2009

Impressum:
Hrsg.: Leiter des Instituts für Mathematik
Weimarer Straße 25
98693 Ilmenau
Tel.: +49 3677 69 3621
Fax: +49 3677 69 3270
http://www.tu-ilmenau.de/ifm/

ISSN xxxx-xxxx

ilmedia

An $\Omega(n \log n)$ lower bound for computing the sum of even-ranked elements

Marc Mörig* Dieter Rautenbach† Michiel Smid‡
Jan Tusch*

April 17, 2009

Abstract

Given a sequence A of $2n$ real numbers, the EVENRANKSUM problem asks for the sum of the n values that are at the even positions in the sorted order of the elements in A. We prove that, in the algebraic computation-tree model, this problem has time complexity $\Theta(n \log n)$. This solves an open problem posed by Michael Shamos at the Canadian Conference on Computational Geometry in 2008.

1 Introduction

Let $A = (a_1, a_2, \ldots, a_{2n})$ be a sequence of $2n$ real numbers. We define the *even-rank-sum* of A to be the sum of the n values that are at the even positions in the sorted order of the elements in A. Formally, let π be a permutation of $\{1, 2, \ldots, 2n\}$ that sorts the sequence A in non-decreasing order; thus, $a_{\pi(1)} \leq a_{\pi(2)} \leq \ldots \leq a_{\pi(2n)}$. Then the even-rank-sum of the sequence A is the real number

$$a_{\pi(2)} + a_{\pi(4)} + a_{\pi(6)} + \ldots + a_{\pi(2n)}.$$

*Faculty of Computer Science, University of Magdeburg, Magdeburg, Germany.
†Faculty of Mathematics and Natural Sciences, Ilmenau University of Technology, Ilmenau, Germany.
‡School of Computer Science, Carleton University, Ottawa, Ontario, Canada. Research supported by NSERC.

Observe that any permutation π that sorts the sequence A in non-decreasing order gives rise to the same even-rank-sum. We consider the following problem:

EVENRANKSUM: Given a sequence A of $2n$ real numbers, compute the even-rank-sum of A.

By using an $O(n \log n)$–time sorting algorithm, this problem can be solved in $O(n \log n)$ time. In the Open Problem Session at the Canadian Conference on Computational Geometry in 2008, Michael Shamos posed the problem of proving an $\Omega(n \log n)$ lower bound on the time complexity of EVENRANKSUM in the algebraic computation-tree model. (See [1, 2] for a description of this model.) In this paper, we present such a proof:

Theorem 1 *In the algebraic computation-tree model, the time complexity of* EVENRANKSUM *is* $\Theta(n \log n)$.

We prove Theorem 1 by presenting an $O(n)$–time reduction of MINGAP to EVENRANKSUM. The former problem is defined as follows. Let $X = (x_1, x_2, \ldots, x_n)$ be a sequence of n real numbers, and let π be a permutation of $\{1, 2, \ldots, n\}$ such that $x_{\pi(1)} \leq x_{\pi(2)} \leq \ldots \leq x_{\pi(n)}$. For each $1 \leq i < n$, we define the difference $x_{\pi(i+1)} - x_{\pi(i)}$ to be a *gap* in the sequence X.

MINGAP: Given a sequence $X = (x_1, x_2, \ldots, x_n)$ of n real numbers and a real number $g > 0$, decide if each of the $n - 1$ gaps in X is at least g.

Since in the algebraic computation-tree model, MINGAP has an $\Omega(n \log n)$ lower bound (see [2, Section 8.4]), our reduction will prove Theorem 1.

2 The proof of Theorem 1

We now show how to reduce, in $O(n)$ time, MINGAP to EVENRANKSUM.

Let $\mathcal{A}$ be an arbitrary algorithm that solves EVENRANKSUM. We show how to use algorithm $\mathcal{A}$ to solve MINGAP. Let $n \geq 2$ be an integer and consider a sequence $X = (x_1, x_2, \ldots, x_n)$ of n real numbers and a real number $g > 0$. The algorithm for solving MINGAP makes the following three steps:

Step 1: Compute $S = \sum_{i=1}^{n} x_i$ and, for $i = 1, 2, \ldots, n$, compute $a_{2i-1} = x_i$ and $a_{2i} = x_i + g$.

Step 2: Run algorithm $\mathcal{A}$ on the sequence $(a_1, a_2, \ldots, a_{2n})$, and let R be the output, i.e., R is the even-rank-sum of this sequence.

Step 3: If $R = S + ng$, then return YES. Otherwise, return NO.

It is clear that the running time of this algorithm is $O(n)$ plus the running time of $\mathcal{A}$. Thus, it remains to show that the algorithm correctly solves MinGap. That is, we have to show that the minimum gap G of X is at least g if and only if $R = S + ng$. This is an immediate consequence of the following lemma:

Lemma 1 *Let $x_1, x_2, \ldots, x_n$ and g be real numbers such that $x_1 \le x_2 \le \ldots \le x_n$ and $g > 0$. Let $(a_1, a_2, \ldots, a_{2n}) = (x_1, x_1 + g, x_2, x_2 + g, \ldots, x_n, x_n + g)$ and let π be a permutation of $\{1, \ldots, 2n\}$ such that $b_1 \le b_2 \le \ldots \le b_{2n}$ with $b_i = a_{\pi(i)}$ for $1 \le i \le 2n$.*

If $R = \sum_{i=1}^{n} b_{2i}$, $U = \sum_{i=1}^{n} b_{2i-1}$, and $G = \min\{x_{i+1} - x_i \mid 1 \le i \le n-1\}$, then $R - U \le ng$ with equality if and only if $G \ge g$.

Proof. Since $x_1, x_1+g, x_2, x_2+g, \ldots, x_i, x_i+g \le x_i+g$, we have $x_i+g \ge b_{2i}$ for $1 \le i \le n$. Since $x_i, x_i + g, x_{i+1}, x_{i+1} + g, \ldots, x_n, x_n + g \ge x_i$, we have $x_i \le b_{2i-1}$ for $1 \le i \le n$. Hence $b_{2i} - b_{2i-1} \le (x_i + g) - x_i = g$ for $1 \le i \le n$ which implies $R - U \le ng$.

If $G \ge g$, then clearly $R - U = ng$. Conversely, if $R - U = ng$, then $b_{2i} - b_{2i-1} = g$ for $1 \le i \le n$. In view of the above, this implies that $x_i + g = b_{2i}$ and $x_i = b_{2i-1}$ for $1 \le i \le n$. Since $x_{i+1} = b_{2i+1} \ge b_{2i} = x_i + g$ for $1 \le i \le n-1$, we obtain $G \ge g$. ∎

We complete the proof of Theorem 1 by observing that $R + U = 2S + ng$ and by Lemma 1 we have $G \ge g$ if and only if $R = U + ng = S + ng$.

References

[1] M. Ben-Or. Lower bounds for algebraic computation trees. In *Proceedings of the 15th ACM Symposium on the Theory of Computing*, pages 8086, 1983.

[2] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, Berlin, 1988.