$\Omega(\log\log(1/\epsilon))$ LOWER BOUND FOR APPROXIMATING THE SQUARE ROOT

Nader H. Bshouty

Department of Computer Science, Technion-I.I.T, Haifa, Israel
Department of Computer Science, University of Calgary
2500 University Drive N.W.
Calgary, Alberta, Canada T2N 1N4
e-mail: bshouty@cpsc.ucalgary.ca

ABSTRACT:

In [FOCS 89], Mansour-Schieber-Tiwari proved that any computation tree with the operations $\{+,-,\times,/,\lfloor\ \rfloor,<\}$ and constants $\{0,1\}$ that computes $\sqrt{x}$ to accuracy ϵ, for all $x\in[1,2]$, must have depth $\Omega(\sqrt{\log\log(1/\epsilon)})$.

In this paper we prove that any computation tree with operations $\{+,-,\times,/,\lfloor\ \rfloor,<$, NOT, AND, OR, XOR$\}$, indirect addressing, unlimited power of answering YES/NO questions and constants $\{0,1\}$ that computes $\sqrt{x}$ to accuracy ϵ for all $x\in[1,2]$ must have depth $\Omega(\log\log(1/\epsilon))$. By Newton iteration our bound is tight.

1. THE MODEL

Let $\mathbf{Q}$ and $\mathbf{Z}$ be the rational field and the integer ring, respectively. A random access machine RAM has an unbounded number of registers $\{M[i]\}_{i\in\mathbf{Z}}$, each of which can store an element in the rational field $\mathbf{Q}$. The register $M[0]$ is called the accumulator A. The computation is directed by a finite program that consists of instructions of the following type: direct and indirect storage accesses, conditional branching (IF-THEN-ELSE), arithmetic operations $\{+,-,\times,/\}$, integer floor $\{\lfloor\ \rfloor\}$, integer boolean operations $\{NOT, AND, OR, XOR\}$. To formalize this we define the following: A program is a finite sequence $P=(1:I_1),\cdots,(q:I_q)$ of instructions from the following set:($e\in\mathbf{Z},\quad d_1,d_2\in\{1,\cdots,q\}$)

(1) $A\leftarrow 1,\ A\leftarrow M[e],\ A\leftarrow M[M[e]]$.

(2) $M[e]\leftarrow A,\ M[M[e]]\leftarrow A$.

(3) $A\leftarrow A\circ M[e]$ where $\circ\in\{+,-,\times,/,AND,OR,XOR\}$.

(4) $A\leftarrow \circ A$ where $\circ\in\{\lfloor\ \rfloor, NOT\}$.

(5) IF <any condition> THEN GOTO d_1 ELSE GOTO d_2.

Here $\lfloor\ \rfloor x=\lfloor x\rfloor$ is the greatest integer that is not greater than x, $M[M[e]]$ is $M[\lfloor M[e]\rfloor]$, $NOT(x)$ is $(2^{\lfloor\log_2|x|\rfloor+2}-1)-\lfloor|x|\rfloor$ where $|x|$ is the absolute value of x and $x\ AND\ y$, $x\ OR\ y$ and $x\ XOR\ y$ are bitwise AND, OR and XOR, respectively, of the binary representations of $\lfloor|x|\rfloor$ and $\lfloor|y|\rfloor$.

In the program P the first instruction I_1 is the input step of $x\in\mathbf{Q}$, which is stored in A. The content of the accumulator A after the execution of the program is the output.

We say that the program P computes $\sqrt{x}$ with error ϵ if the execution of the program for the inputs $x\in[1,2]$ stops with $A=y$, where $|y-\sqrt{x}|\le\epsilon$. The *complexity* $Comp(P)$ of the program P is the maximal number of steps of the forms (3) and (4) over all the possible inputs $x\in[1,2]$ needed to execute the program. The complexity $Comp(\sqrt{x},\epsilon)$ is the minimum of $Comp(P)$ over all programs P that computes $\sqrt{x}$ with error ϵ.

2. OLD AND NEW RESULTS

In [FOCS89] Mansour-Shieber-Tiwari proved the following

Theorem 1. *Any computation tree with operations* $\{+,-,\times,/,\lfloor\ \rfloor,<\}$ *and constants* $\{0,1\}$ *that computes* $\sqrt{x}$ *to accuracy* ϵ, *for all* $x\in[1,2]$, *must have depth* $\Omega(\sqrt{\log\log(1/\epsilon)})$.

In this paper we prove the following stronger result

Theorem 2. *Any program with direct and indirect addressing, unlimited power of answering YES/NO questions (also questions that use any constant as* ϵ*), arithmetic operations* $\{+,-,\times,/,\lfloor\ \rfloor\}$ *and integer bitwise boolean operations* $\{NOT,AND,OR,XOR\}$ *that computes* $\sqrt{x}$ *to accuracy* ϵ, *for all* $x\in[1,2]$ *must have depth*

$$\Omega\left(\log\log\left(\frac{1}{\epsilon}\right)\right).$$

It is also well known that by Newton iteration we have

Theorem 3. *There exist a computation tree with operations* $\{+,-,\times,/,>\}$ *and constants* $\{0,1\}$ *that computes* $\sqrt{x}$ *to accuracy* ϵ *with depth* $O(\log\log(1/\epsilon))$.

3. PRELIMINARY RESULTS

In this section we give the lemma needed for the proof of ours result. We shall present the proof of W. Eberly because it is shorter than our

Lemma 1. *Any straight line algorithm with the operations* $\{+,-,\times,/,$ *NOT, AND, OR, XOR*$\}$ *and constants* $\{0,1\}$ *that computes a constant* N *must have depth* $\Omega(\log\log(N))$.

Proof . Since the algorithm begins with constants 0,1 all the constants in the algorithm are rational numbers. Let $x_i = s_i\frac{x_{i,1}}{x_{i,2}}$ be the constant that is computed in step i, where $s_i = sign(x_i)$ and $x_{i,1}, x_{i,2}$ are positive integers, $gcd(x_{i,1}, x_{i,2}) = 1$. Let $T_i = \max_{j\le i}\max\{x_{j,1}, x_{j,2}\}$, then $T_1 \le 1$ and obviously, $T_i \le 2T_i^2$. This implies that $T_i \le 2^{2^{i+1}-1}$. If N is computed at depth t then $N = x_t \le T_t \le 2^{2^{t+1}-1}$ which implies the result. ○

4. PROOF OF THEOREM 2

Let T be a program with depth t that computes an ϵ-approximation of $x \in [1,2]$. We look at the path that computes $\sqrt{2}$. This path is a straight line algorithm $I_1,\cdots,I_{t'}$ that computes a number $\sqrt{2}+\delta$ in $[\sqrt{2}-\epsilon, \sqrt{2}+\epsilon]$ with the constants $\{0,1,2\}$. Since this straight line algorithm is a path in the tree we have

$$t' \le t, \qquad |\delta| \le \epsilon. \tag{1}$$

Since we begin with rational numbers, the constant $\sqrt{2}+\delta$ is rational and therefore

$$\delta \neq 0 \tag{2}$$

Now we add to $I_1,\cdots,I_{t'}$ the following steps:

I_{-1} :	Compute $1+1=2$.
I_0 :	Compute $2+2=4$.
$I_{t'+1}$:	Compute $(\sqrt{2}+\delta)^2 = 2+2\sqrt{2}\delta+\delta^2$.
$I_{t'+2}$:	Compute $(2+2\sqrt{2}\delta+\delta^2)-2 = 2\sqrt{2}\delta+\delta^2$.
$I_{t'+3}$:	Compute $(2\sqrt{2}\delta+\delta^2)/4 = \frac{\sqrt{2}}{2}\delta+\frac{1}{4}\delta^2$.
If the constant in $I_{t'+3}$ is positive then we add	
$I_{t'+4}$:	Compute $N = \frac{1}{(\sqrt{2}/2)\delta+(1/4)\delta^2}$.
Otherwise we add	
$I_{t'+4}$:	Compute $0-(\frac{\sqrt{2}}{2}\delta+\frac{1}{4}\delta^2) = -\frac{\sqrt{2}}{2}\delta-\frac{1}{4}\delta^2$.
$I_{t'+5}$:	Compute $N = \frac{1}{-(\sqrt{2}/2)\delta-(1/4)\delta^2}$.

In both cases (for sufficiently small ϵ) we compute a constant

$$N \geq \frac{1}{\epsilon}.$$

By lemma 1 we have $t' + 7 \geq \Omega(\log\log(1/\epsilon))$ and by (1) the result follows. ○

Acknowlegement. I would like to thank Wayne Eberly and Lisa Higham for a number of helpful comments.

REFERENCES

[FOCS89] Y. Mansour, B. Schieber, P. Tiwari, The complexity of approximating the square root. *Proc. 30th IEEE Symp. on Foundations of Computer Science*, 1989.

[Renegar] J. Renegar, On the worst-case arithmetic complexity of approximating the square root, *J. of Complexity*, 3, (1987), 90-113.