# Integer Complexity: Breaking the $\Theta(n^2)$ barrier

Srinivas Vivek V., and Shankar B. R.

***Abstract*—**The *integer complexity* of a positive integer $n$, denoted $f(n)$, is defined as the least number of 1's required to represent $n$, using only 1's, the addition and multiplication operators, and the parentheses. The running time of the algorithm currently used to compute $f(n)$ is $\Theta(n^2)$. In this paper we present an algorithm with $\Theta(n^{\log_2 3})$ as its running time. We also present a proof of the theorem: the largest solutions of $f(m) = 3k,\ 3k\pm1$ are, respectively, $m = 3^k,\ 3^k \pm 3^{k-1}$.

***Keywords*—**Integer complexity, Number theory, Running time

## I. Introduction

The *integer complexity* of a positive integer $n$, denoted $f(n)$, is defined as the least number of 1's required to represent $n$, using only 1's, the addition and multiplication operators, and the parentheses [3]. $f(n)$ can be computed as follows [2]:

$$f(n) = \min\left\{f(e) + f(n-e),\ f(d) + f(n/d)\right\}, \quad (1)$$

where $d|n,\ 2 \le d \le \sqrt{n}$, and $1 \le e \le n/2$.

This is the currently used algorithm to compute $f(n)$. The above algorithm, when implemented in the bottom-up manner i.e. computing the smaller terms of the sequence first and then reusing them later, runs in time $\Theta(n^2)$. We must note that here the running time of the algorithm is measured in terms of the number of comparisons required to determine the value of $f(n)$. The above running time can be arrived at as follows: let the total number of comparisons needed to determine $f(n)$ be $C_n$. Then $C_n = O\left(\sum_{j=2}^{n}\left(\frac{j}{2} + \sqrt{j}\right)\right)$ and $C_n = \Omega\left(\sum_{j=2}^{n}\frac{j}{2}\right)$, where $j \in \mathbb{N}$. Therefore, $C_n = \Theta(n^2)$. We now present an algorithm whose running time is $\Theta(n^{\log_2 3})$.

## II. Proposed Algorithm

We propose that

$$f(n) = \min\left\{f(e') + f(n-e'),\ f(d) + f(n/d)\right\}, \quad (2)$$

where $d|n,\ 2 \le d \le \sqrt{n},\ 1 \le e' \le n(1-r_n)/2,\ r_n = \sqrt{1 - \frac{4(3)^{1/3}(n-1)^{\log_2 3}}{n^2}}$, and $n \ge 65$.

The inequality $n \ge 65$ follows from the fact that $4(3)^{1/3}\,(n-1)^{\log_2 3} \le n^2$. It is interesting to observe that $\lim_{n\to\infty} r_n = 1$.

### *Proof of correctness*

To prove the correctness of the algorithm (2), we show that

$$f(e') + f(n-e') \ \ge\ f(n-1) + f(1) \quad \forall\, e' \in \mathbb{N} \cap [n(1-r_n)/2,\ n/2]\,. (3)$$

We arrive at the above result by trying to find a range of values of $e$ for which the inequality (3) is true. In [2] it is shown that $3\,\log_3 n \le f(n) \le 3\,\log_2 n = 3\,(\log_2 3)\,\log_3 n$. By using this fact and assigning the individually smallest possible values to $f(e')$ and $f(n-e')$, and the largest possible value to $f(n-1)$ in the inequality (3), we get

$$3\,\log_3 e' + 3\,\log_3(n-e') \ge 3\,\log_2(n-1) + 1$$

$$\Rightarrow\ \log_3 \frac{e'(n-e')}{(n-1)^{\log_2 3}} \ge \frac{1}{3}$$

$$\Rightarrow\ -(e')^2 + n\,e' - 3^{1/3}(n-1)^{\log_2 3} \ \ge\ 0. \quad (4)$$

The integer values of $e'\ (\le n/2)$ which satisfy (4) are $\mathbb{N} \cap \left[\frac{n}{2}\left(1 - \sqrt{1 - \frac{4(3)^{1/3}(n-1)^{\log_2 3}}{n^2}}\right),\ \frac{n}{2}\right]$, which can be equivalently written as $\mathbb{N} \cap [n(1-r_n)/2,\ n/2]$. Clearly, the values of $e'$ which satisfy (4) also satisfy (3). Hence, the proof of correctness of the algorithm.

### *Running time*

Let $C_n$ be as previously defined . Then $C_n = O\left(\sum_{j=65}^{n}\left(\frac{j(1-r_j)}{2} + \sqrt{j}\right)\right)$ and $C_n = \Omega\left(\sum_{j=65}^{n}\frac{j(1-r_j)}{2}\right)$. Let $h_j = \frac{4(3)^{1/3}(j-1)^{\log_2 3}}{j^2}$. By binomial expansion,

$$j(1-r_j) \ = \ j\left(1 - \sqrt{1-h_j}\right) \le \frac{j\,h_j}{1-h_j}$$

$$\Rightarrow\ j(1-r_j) = \Theta\left(j^{(\log_2 3)-1}\right).$$

$$\Rightarrow\ C_n = \Theta\left(n^{\log_2 3}\right) = O\left(n^{1.59}\right).$$

Hence, the running time of the proposed algorithm is $\Theta\left(n^{\log_2 3}\right)$.

This paper is based on a thesis submitted by Srinivas Vivek V., under the guidance of Dr. Shankar B. R., in partial fulfillment of the requirements for the degree of Bachelor of Technology in the Department of Information Technology at the National Institute of Technology Karnataka, Surathkal, India in April 2008. This paper was previously available at www.waset.org since 2008. As on October 2014, it is no longer available there due to unknown reasons.

Srinivas Vivek V. was with the Department of Information Technology, National Institute of Technology Karnataka, Surathkal, India. As on October 2014, the author is at the University of Luxembourg. Email: svivekv@gmail.com

Dr. Shankar B. R. is with the Department of Mathematical and Computational Sciences, National Institute of Technology Karnataka, Surathkal, India. Email: shankarbr@gmail.com

## III. Proof of a Theorem

The following theorem has been mentioned in [2]. It is also mentioned there that the theorem has been proved using induction but the proof was not given. We now give a proof of the theorem. This proof also is based on induction.

*Theorem:* The largest solutions of $f(m) = 3k,\, 3k \pm 1$ are, respectively, $m = 3^k,\, 3^k \pm 3^{k-1}$.

*Proof:* Define $g(s)$ as the largest number which can be formed using $s$ number of 1's, the addition and multiplication operators, and the parentheses. Clearly, the largest solution of $f(m) = s$ is $m = g(s)$. $g(s)$ can be recursively obtained by the following formula:

$$g(s) = \max\left\{g(e) + g(s-e),\, g(e) \times g(s-e)\right\},$$

where $1 \le e \le s/2$.

Let the above theorem be true for $k = 1, 2 \ldots n-1$. First, we prove the case of $f(m) = 3n-1$, and in a similar manner we subsequently prove the cases of $f(m) = 3n$ and $f(m) = 3n+1$. It can be easily verified that when $k = 1$, the theorem is true since $g(2) = 2$, $g(3) = 3$, and $g(4) = 4$. Define $g'(s,\, e) = \max\left\{g(e) + g(s-e),\, g(e) \times g(s-e)\right\}$. It is easy to see that $g'(s,\, e) = g(e) \times g(s-e)\ \forall\, e \ge 2,\ s \ge 4$.

First, consider $f(m) = 3n-1$.
$g'(3n-1,\, 1) = 1 + g(3(n-1)+1) = 1 + 3^{n-1} + 3^{n-2}$.
Let $2 \le e = 3k'-1 \le (3n-1)/2$, then $g'(3n-1,\, e) = g(3k'-1) \times g(3(n-k')) = \left(3^{k'} - 3^{k'-1}\right) \times 3^{n-k'} = 3^n - 3^{n-1}$.
Let $2 \le e = 3k' \le (3n-1)/2$, then $g'(3n-1,\, e) = g(3k') \times g(3(n-k')-1) = 3^{k'} \times \left(3^{n-k'} - 3^{n-k'-1}\right) = 3^n - 3^{n-1}$.
Let $2 \le e = 3k'+1 \le (3n-1)/2$, then $g'(3n-1,\, e) = g(3k'+1) \times g(3(n-k'-1)+1) = \left(3^{k'} + 3^{k'-1}\right) \times \left(3^{n-k'-1} + 3^{n-k'-2}\right) = 16 \times 3^{n-3}$.
Therefore, the maximum possible value of $g'(3n-1,\, e)$ is $3^n - 3^{n-1}$. Hence, the theorem is proved for the case of $f(m) = 3n-1$.

Next, consider $f(m) = 3n$.
$g'(3n,\, 1) = 1 + g(3n-1) = 1 + 3^n - 3^{n-1}$.
Let $2 \le e = 3k'-1 \le 3n/2$, then $g'(3n,\, e) = g(3k'-1) \times g(3(n-k')+1) = \left(3^{k'} - 3^{k'-1}\right) \times \left(3^{n-k'} + 3^{n-k'-1}\right) = 8 \times 3^{n-2}$.
Let $2 \le e = 3k' \le 3n/2$, then $g'(3n,\, e) = g(3k') \times g(3(n-k')) = 3^{k'} \times 3^{n-k'} = 3^n$.
Let $2 \le e = 3k'+1 \le 3n/2$, then $g'(3n,\, e) = g(3k'+1) \times g(3(n-k')-1) = \left(3^{k'} + 3^{k'-1}\right) \times \left(3^{n-k'} - 3^{n-k'-1}\right) = 8 \times 3^{n-2}$.
Therefore, the maximum possible value of $g'(3n,\, e)$ is $3^n$. Hence, the theorem is proved for the case of $f(m) = 3n$.

Finally, consider $f(m) = 3n+1$.
$g'(3n+1,\, 1) = 1 + g(3n) = 1 + 3^n$.
Let $2 \le e = 3k'-1 \le (3n+1)/2$, then $g'(3n+1,\, e) = g(3k'-1) \times g(3(n-k'+1)-1) = \left(3^{k'} - 3^{k'-1}\right) \times \left(3^{n-k'+1} - 3^{n-k'}\right) = 3^n + 3^{n-1}$.
Let $2 \le e = 3k' \le (3n+1)/2$, then $g'(3n+1,\, e) = g(3k') \times g(3(n-k')+1) = 3^{k'} \times \left(3^{n-k'} + 3^{n-k'-1}\right) = 3^n + 3^{n-1}$.
Let $2 \le e = 3k'+1 \le (3n+1)/2$, then $g'(3n+1,\, e) = g(3k'+1) \times g(3(n-k')) = \left(3^{k'} + 3^{k'-1}\right) \times 3^{n-k'} = 3^n + 3^{n-1}$.
Therefore, the maximum possible value of $g'(3n+1,\, e)$ is $3^n + 3^{n-1}$. Hence, the theorem is proved for the case of $f(m) = 3n$+1, and this completes the proof of the theorem. ■

## IV. Conclusion

In this paper we have presented a more efficient algorithm to compute the *integer complexity* that runs in $\Theta(n^{\log_2 3})$ time when compared with the existing algorithm that runs in $\Theta(n^2)$ time. Working on the same lines, a better algorithm can be provided if one can give a tighter upper bound on the value of $f(n)$ than the existing $3\log_2 n$ bound. Also, we have given a proof of the theorem: the largest solutions of $f(m) = 3k,\, 3k \pm 1$ are, respectively, $m = 3^k,\, 3^k \pm 3^{k-1}$, using induction.

## References


[1] R. K. Guy, Unsolved Problems in Number Theory. Second edition, Springer-Verlag, 1994, Problem F26.
[2] R. K. Guy, Some Suspiciously Simple Sequences. American. Math. Monthly 93, 186-190, 1986.
[3] Sequence A005245, The On-Line Encyclopedia of Integer Sequences. http://www.research.att.com/~njas/sequences/A005245
[4] Integer Complexity, Math Games, MAA Online. http://www.maa.org/editorial/mathgames/mathgames_04_12_04.html