Dokl. Akad. Nauk SSSR
Tom 226 (1976), No. 1

Soviet Math. Dokl.
Vol. 17 (1976), No. 1

THE ADDITIVE COMPLEXITY OF A NATURAL NUMBER

UDC 519.14

È. G. BELAGA

1. The study of optimum multiplicative schemes for calculating the simplest polynomials of the form x^n, where $n \geq 1$ (cf. [1] for the definition of schemes) reduces to the examination of a certain number-theoretic combinatorial function $l(n)$, the least possible number of multiplications in such schemes. We will call this function the additive complexity of the natural number n, or briefly the length of n. The height $l(n)$ is also the least number of additions necessary to obtain n from 1.

The function $l(n)$ was first defined by A. Scholz [2], and its simplest properties as well as an upper bound and asymptotics were found and proved by A. Brauer [3]. P. Erdös [4] proved a theorem refining the asymptotic law for $l(n)$ for "almost all" n.

The interesting problem of a nontrivial lower bound for $l(n)$ was treated in [5], where this bound was found for numbers n of a very special form (the total number of one's in the binary decomposition of n is at most 3); D. E. Knut [6] somewhat strengthened this result (4 one's). A number of open problems related to the "local" (passage from $l(n)$ to $l(2n)$ and so on) and "global" ("ramified" numbers and so on) behavior of $l(n)$ was discussed in [6]; the latter article also contains a complete survey of results through 1967.

In the present note a refinement is given of an upper bound for $l(n)$ (Theorem 1), a nontrivial general lower bound (Theorem 2) is proved, and it is proved that this bound cannot be improved with a slightly larger constant (Theorem 3). We will also discuss the problem of calculating $l(n)$ on a modern computer.

2. We set $\lambda(n) = -[-\log_2 n]$ (the number of digits in the binary representation of n if n is not a power of 2, and the number of zeros in this representation if n is a power of 2).([1]) The following bounds and "recursion inequalities" for $l(n)$ are entirely elementary:

$$\lambda(n) \leqslant l(n) \leqslant 2\lambda(n), \tag{1}$$

$$l(n_1 \cdot n_2) \leqslant l(n_1) + l(n_2), \tag{2}$$

$$l(n_1+n_2) \leqslant l(n_1) + l(n_2), \quad \text{if} \quad n_1 \geqslant 2, \quad n_2 \geqslant 2, \tag{3}$$

$$l(n+1) \leqslant l(n) + 1. \tag{4}$$

The upper bound for $l(n)$ can be improved:

Theorem 1. *For any two natural numbers n and p we have the inequality*

$$l(n) \leqslant (1+1/p)\lambda(n) + 2^{p-1} - p - 1/p. \tag{5}$$

Remark. Theorem 1 was first proved, though in a somewhat weaker form, by A.

AMS (MOS) subject classifications (1970). Primary 10A30, 10–04.

([1]) Unlike [6], where $\lambda(n) = [\log_2 n]$ is always one less than the number of digits ($[x]$ is the integral part of a real x).

Brauer [3] and then proved again [7] in the same weaker form

$$l(n)\leqslant(1+1/p)\lambda(n)+2^p.$$

Corollary 1. *For all* $n\geq 1$ *we have the inequalities*

$$l(n)\leqslant{}^3/_2\lambda(n)-{}^1/_2,\quad l(n)\leqslant{}^4/_3\lambda(n)+{}^2/_3,\quad l(n)\leqslant{}^5/_4\lambda(n)+{}^{13}/_4. \tag{6}$$

We may conclude from the table of values(2) for $l(n)$ that the inequalities (6) obtained from (5) when $p = 2$, 3 and 4 cannot be strengthened for numbers n and height $l(n)\leq 12$ in this notation.

Conjecture. The bound (5) is sharp in the sense that for any $N > 0$ there exist $n > N$ and $p > 0$ such that equality holds in (5).

Corollary 2. *There exists a constant* α, $1 < \alpha < 1 + (2\log_2 e)/e < 2.1$, such that for all $n\geq 2$ we have

$$l(n)\leqslant\lambda(n)+\alpha\cdot\lambda(n)/\log_2\lambda(n). \tag{7}$$

Theorem 1 is proved below (§3). Inequality (7) is obtained from (5) if $p = \log_2\lambda(n) - 2\log_2\log_2\lambda(n)$. In fact the minimum of the expression in the right side of (5) is reached when

$$p=\log_2\left(\frac{2}{\ln 2}\lambda(n)\right)-2\log_2\log_2\lambda(n)+o(1);$$

after simple calculations we obtain the inequality

$$l(n)\leqslant\lambda(n)+\frac{\lambda(n)}{\log_2\lambda(n)}+2\frac{\lambda(n)}{\log_2\lambda(n)}\frac{\log_2\log_2\lambda(n)}{\log_2\lambda(n)},$$

which also implies (7) in view of the obvious inequality $x\geq(\log_2 e/e)\cdot\log_2 x$ $(x > 0)$.

Remark. Erdös proved [4] that for "almost all" n the length $l(n)$ is asymptotically equal to $\lambda(n) + \lambda(n)/\lambda(\lambda(n))$; thus the lower bound for the constant α in (7) cannot be decreased.

3. Proof of Theorem 1. For $p\geq 1+\log_2\lambda(n)$ the inequality (5) follows from the right-most inequality in (1). Suppose $p < 1 + \log_2\lambda(n)$ and thereby $p < \lambda(n)$. Let us indicate a scheme for calculating n, the number of additions in which is less than the right side of inequality (5). The idea of this scheme, found by the author in 1973, was used by Brauer [3] and R. È. Val′skiĭ [7], but in a somewhat cruder form.

Thus, suppose $1\leq p < \lambda(n)$ and let

$$n=n_1+n_2\cdot 2^p+n_3\cdot 2^{2p}+\ldots+n_{q-1}2^{(q-2)p}+n_q2^{(q-2)p+\Delta}, \tag{8}$$

where

$$\lambda(n_j)\leqslant p,\quad 1\leqslant j\leqslant q-1,\quad \lambda(n_q)\leqslant\Delta,$$
$$q=-\left[-\frac{\lambda(n)}{p}\right]\leqslant\frac{\lambda(n)-1}{p}+1,\quad 1\leqslant\Delta=\lambda(n)-p(q-1)\leqslant p. \tag{9}$$

Assuming that the numbers $n_1, \ldots, n_q$ have already been constructed, we may consider (8) as a compact formulation of the scheme for calculating the number n in terms of the known numbers $n_1, \ldots, n_q$, where we first carry out the Δ-fold doubling of n_q

(2) Calculations were performed at the Institute of Control Problems of the Academy of Sciences of the USSR by A. Futer and the author. A detailed description of the algorithm and a table of values for $l(n)$ will be published.

(doubling is realized as the addition of a number to itself), followed by the addition of n_{q-1} to the resulting number, then the p-fold doubling of the result, and so on. The number of additions in this scheme, in view of (9), is given by

$$p(q-2)+\Delta+q=\lambda(n)+q-1-p\leqslant(1+1/p)\lambda(n)-p-1/p.$$

We now note that, instead of the p-digit numbers $n_1, \ldots, n_q$, it suffices to know all the odd p-digit numbers from 1 to $2^p - 1$ inclusively. In fact if m is such a number and $n_k = 2^l m$, the fragment of the scheme consisting in the p-fold doubling and successive addition of n_k should be replaced by the $(p-1)$-fold doubling of this intermediate result and the addition of the m- and l-fold doublings of the resulting number. 2^{p-1} additions are necessary (and sufficient) to calculate all odd numbers less than 2^p. The theorem is proved.

4. Not much is yet known concerning lower bounds for $l(n)$; the bound

$$l(n)\leqslant\lambda(n) \tag{10}$$

is trivial, though it cannot be improved using solely the classical monotonically increasing functions. For example, for any $k > 0$, $l(2^k) = \lambda(2^k) = k$. It is immediately clear that the size of the difference $l(n) - \lambda(n)$ depends both on the number $\nu(n)$ of one's in the binary decomposition of n and on the order of the one's in this decomposition. A formalization of the first of these dependences is possible, and we will prove the corresponding bound below. The more delicate problem concerning the dependence of the deviation of $l(n) - \lambda(n)$ on the measure of irregularity of occurrence of one's in the binary decomposition of n remains as yet open. These two approaches, we should mention, have been synthesized in very special cases in theorems derived in [5], [6].

Theorem (A. A. Gioia et al.). $l(2^A + 2^B + 2^C) = A + 2$ *if* $A > B > C$.

Theorem (Knut). *If* $\nu(n) \geq 4$, *then* $l(n) \geq \lambda(n) + 3$ *except in the following cases when* $A > B > C > D$ *and* $l(2^A + 2^B + 2^C + 2^D) = A + 2$:

Case 1. $A - B = C - D$ (*example*: $n = 15$).

Case 2. $A - B = C - D + 1$ (*example*: $n = 23$).

Case 3. $A - B = 3$, $C - D = 1$ (*example*: $n = 39$).

Case 4. $A - B = 5$, $B - C = C - D = 1$ (*example*: $n = 135$).

Let us prove the following general result.

Theorem 2. *There exists a constant* β, $1/5 < \beta \leq 1$, *such that for all* $n \geq 1$, *we have*

$$\lambda(n)+\beta\cdot\lambda(\nu(n))\leqslant l(n). \tag{11}$$

Remark. Inequality (11) is apparently true for all $n \geq 1$ with unit constant $\beta = 1$. Furthermore, the very plausible equality

$$l(2^{2^k} - 1) = 2^k + k,$$

which is implied by the unproved conjecture $\lambda(n) + \lambda(\nu(n)) \leq l(n)$ and the obvious inequality $l(2^{2^k} - 1) \leq 2^k + k$ has yet to be either proved or disproved.

We will also prove that the (unique) upper bound for the constant β cannot be increased.

Theorem 3. *For any* $\epsilon > 0$ *the inequality*

$$\lambda(n)+(1+\varepsilon)\lambda(\nu(n))>l(n)$$

holds for an infinite number of natural n. *Moreover, the inequality*

$$\lambda(n)+l(\nu(n))>l(n) \tag{12}$$

also possesses this property.

Remark. Inequality (12) is stronger than the preceding inequality in view of (7) and (10).

5. In proving Theorem 2 we will use the following assertions, which are of some independent interest.

Let us consider an arbitrary scheme with k additions (we will refer to any indexed sequence of additions in which either one's or the results of additions with a lower index enter as a scheme; the result of a scheme may be nonunique) and denote by n_q the result of an addition with index q $(0 \le q \le k$ and we agree that $n_0 = 1)$. An addition with index q is said to be a doubling if $n_q = n_r + n_r = 2n_r$, $0 \le r \le q-1$. We denote by j_q the number of doublings in operations with indices from 1 to q inclusive. (In any scheme $n_1 = n_0 + n_0 = 2n_0 = 2$, so that $1 \le j_q \le q$.) We note that

$$j_{q-1}\leqslant j_q, \quad (q-1)-j_{q-1}\leqslant q-j_q \tag{13}$$

for all q.

Lemma 1. *For any scheme with* k *additions and any* q, $1 \le q \le k$, *we have*

$$n_q\leqslant 2^{j_q-1}\cdot F_{q-j_q+3}, \tag{14}$$

where F_i $(i = 0, 1, 2, \ldots)$ *is a Fibonacci number.*

The proof is carried out by induction on $q \ge 1$, using the inequalities (13) and the definition of Fibonacci numbers.

Lemma 2. *For any scheme with* k *additions and any* q, $1 \le q \le k$, *we have*

$$\lambda(\nu(n_q))\leqslant q-j_q. \tag{15}$$

Proof. Induction on $q \ge 1$, using the obvious inequalities $\nu(2m) = \nu(m)$ and $\nu(m+n) \le \nu(m) + \nu(n)$.

To prove Theorem 2 we use (14) and (15) and the well-known bound on Fibonacci numbers

$$F_{t+3}<2\cdot\varphi^t, \quad \varphi=(\sqrt{5}+1)/2\approx 1.6180. \tag{16}$$

Taking logarithms in (14) and using (16) and then (15), we obtain the inequality

$$\lambda(n_q)+(1-\gamma)\lambda(\nu(n_q))\leqslant q-1, \quad \text{where} \quad \gamma\leqslant\log_2\varphi,$$

which implies (11) when $n_q = n$, $q = l(n)$, and $\lambda(\nu(n)) > 0$.

To prove Theorem 3 we set

$$n_t=6\cdot(2^{3\cdot 2^t}-1)/7+2, \quad t=0, 1, 2, \ldots;$$

In particular $n_0 = 8$, $n_1 = 56$, and so on, or in binary representation: $\langle n_0 \rangle = 110 + 10 = 1000$, $\langle n_1 \rangle = 110110 + 10 = 111000$, $\langle n_2 \rangle = 110110110110 + 10 = 110110111000$, and so on. Let us prove that for all $t \geq 33$

$$l(v(n_t)) > l(n_t) - \lambda(n_t).$$

Evidently

$$\lambda(n_t) = 3 \cdot 2^t - 1, \quad v(n_t) = 2^{t+1} - 1, \quad (n_{t+1} - 2) = 2^{3 \cdot 2^t}(n_t - 2) + n_t - 2,$$

so that

$$l(n_t) \leq l(n_t - 2) + 1 \leq 3 \cdot 2^t + t + 1,$$

and, in view of (11),

$$l(v(n_t)) = l(2^{t+1} - 1) \geq t + 1 + {}^1/_5 \lambda(t+1),$$

so that

$$l(n_t) - \lambda(n_t) \leq t + 2 \leq t + 1 + {}^1/_5 \lambda(t+1) \leq l(v(n_t))$$

for all $t \geq 2^{1/\beta} + 1 \geq 33$.

6. Searches for algorithms to actually calculate the function $l(n)$ have shown that this function possesses several "bounded general recursive" features (this is meant metaphorically, and not strictly speaking). More precisely, to calculate $l(n)$ for a given n it is necessary to sort all the schemes in order of increasing length (i.e. number of additions) and to sort in arbitrary order (say, in lexicographic order for the selected method of coding the schemes) until the first scheme calculating a given n is found; $l(n)$ is set equal to the length of this scheme. Apparently no other more "rapid" method of calculating $l(n)$ for "almost all" n exists, though we know of no proof of this fact. In this connection we note that if $l(n)$ is itself an arithmetic "projection" of complexity in the sense of A. N. Kolmogorov, the problem of calculating it recalls the constructions of the "universal solver" of L. Levin [8], [9].

A significant increase in efficiency is achieved by using a variant of the "branch and bound" method in realizing this algorithm for calculating $l(n)$ on a computer.

Institute for Control Problems

Received 14/JULY/75

BIBLIOGRAPHY

1. È. G. Belaga, Dokl. Akad. Nauk SSSR 123 (1958), 775. (Russian) MR 21 #3935.

2. A. Scholz, Jber. Deutsch. Math-Verein. 47 (1937), Abt. II, 41.

3. A. Brauer, Bull. Amer. Math. Soc. 45 (1939), 736. MR 1, 40.

4. P. Erdös, Acta Arith. 6 (1960), 77. MR 22 #12085.

5. A. A. Gioia, M. V. Subbarao and M. Subunamma, Duke Math. J. 29 (1962), 481. MR 25 #3898.

6. D. E. Knut, *The art of computer programming*. Vols. 1, 2, Addison-Wesley, Reading, Mass., 1969. MR 44 #3530; #3531.

7. R. È. Val'skiĭ, Problemy Kibernet. 2 (1959), 73. (Russian) MR 23 #A875.

8. A. N. Kolmogorov, IEEE Trans. Information Theory IT-14 (1968), 662. MR 39 #3900.

9. L. A. Levin, Problemy Peredači Informacii 9 (1973), 115 = Problems of Information Transmission 9 (1973), 265.

Translated by R. H. SILVERMAN