Rose-Hulman Undergraduate Mathematics Journal

Volume 22
Issue 1
Article 9

The Cost of a Positive integer

Maxwell Norfolk
Bloomsburg University, mn66942@huskies.bloomu.edu

The Cost of a Positive integer

Cover Page Footnote

This manuscript would not have been made possible without Dr. William Calhoun. He has been exceptionally supportive and helpful while working on this project. He also created the original computer programming contest problem (Cheap Integers) where the cost function was originally defined.

$\mathbb{R}$ose-Hulman Undergraduate Mathematics Journal
VOLUME 22, ISSUE 1, 2021

The Cost of a Positive Integer

By *Maxwell Norfolk*

Abstract. The *cost* C_S of a positive integer m relative to a set S of binary operations is defined to be the lesser of m and the minimum of $C_S(a) + C_S(b)$ where a and b are positive integers and $m = a \circ b$ for some binary operation $\circ \in S$. The cost of a positive integer measures the complexity of expressing m using the operations in S, and is intended to be a simplification of Kolmogrov compelexity. We show that, unlike Kolmolgorov complexity, C_S is computable for any finite set S of computable binary operations. We then study C_S for various choices of S, in particular the sets: $\{*\}, \{+,*\}, \{+,\wedge\}, \{+,*,-\}$. Several interesting open questions are also discussed.

1 Introduction

There is an extensive literature on Kolmogorov Complexity. (See [2], for example.) The Kolmogorov complexity of a string (relative to a fixed universal machine) is the length of the shortest program that produces the string as output. For reasons related to Turing's Unsolvability of the Halting Problem, the Kolmogorov complexity of a string is not computable in general. In a 2005 computer programming contest problem called *Cheap Integers*, Dr. William Calhoun defined the *cost* of a positive integer. The objective was to define a notion similar to Komogorov complexity, but simpler and computable. The original recursive definition of the cost function is $C(1) = 1$ and, for $m > 1$, $C(m)$ is the minimum of $C(a) + C(b)$ where $m = a + b$ or $m = ab$. Thus the cost of a positive integer measures the complexity of expressing the integer as a sum or product of smaller positive integers. Since only two operations on the finitely many smaller positive integers need to be checked, $C(m)$ can easily be computed recursively. Still, there is enough complexity to the sequence of values of C to raise many interesting questions. Research on the cost function was begun in 2013 by Dr. William Calhoun and D. Golomb [1]. The current work answers some questions left open previously and generalizes the definition of cost so the cost may be defined relative to a set of binary operations.

Mathematics Subject Classification. 11A41
Keywords. Algorithmic information theory, Kolmogorov complexity, Number theory

1.1 Definition

Definition 1.1 (The Cost of a Positive Integer)**.** The cost function relative to a set of binary operations S on Z^+ is defined recursively by $C_S(1) = 1$ and, for $m > 1$,

$$C_S(m) = \min(\{m\} \cup \{C_S(a) + C_S(b) : m = a \circ b \text{ where } a, b \in \mathbb{Z}^+ \text{ and } \circ \in S\})$$

The following lemma follows directly from Definition 1.1.

Lemma 1.2. *For any set of binary operations* S *and any positive integer* m, $C_S(m) \leq m$.

First we show that the cost is well defined.

Theorem 1.3. *For any set* S *of binary operations on* Z^+ *there is a unique function* C_S *that satisfies Definition 1.1.*

Proof. We define a sequence of sets $A_i \subseteq Z^+$ for $i = 1, 2, \ldots$ so that $A_i = \{m : C_S(m) \leq i\}$. Note that for any $a, b \in Z^+$, $C_S(a) + C_S(b) \geq 2$. Thus $A_1 = \{1\}$. We will see that if i is least such that $m \in A_i$ then $C_S(m) = i$. Since $m \in A_m$ for all m, this means C_S is defined on all of Z^+. Now, we will see how to find A_{i+1} from A_i. If $m \in A_{i+1}$ then either $m \in A_i$, or $m = i + 1$ or $m = a \circ b$ for some $a, b \in Z^+$ and $\circ \in S$ where $C_S(a) + C_S(b) = i + 1$. In the last case, $a, b \in A_i$ since otherwise $C_S(a) + C_S(b) > i + 1$. Therefore, the members of A_{i+1} are determined by A_i and S. Furthermore, to be consistent with Definition 1.1, we must set $C_S(m) = i + 1$ for all $m \in A_{i+1} \setminus A_i$ since it is not possible for $C(a) + C(b) < i + 1$ unless $a, b \in A_i$. The function C_S constructed in this way is consistent with Definition 1.1 since, for each m, $C_S(m)$ is assigned the least value possible according to the definition. Since there is no choice in the construction, the cost function is unique. □

See section 4 for a short table showing this process with $S = \{+, *, -\}$.

Corollary 1.4. *If* S *is a finite set of computable binary operations, then* C_S *is computable.*

Proof. Under the given hypothesis, the sets A_i in the preceding proof are finite and each is computable from the previous one using the construction in the proof. The function C_S is computable from the sequence of sets. □

See the last section for a brief table showing the cost functions considered here for $m \leq 50$.

2 Cost with $S = \{*\}$

In this section we consider the cost function when the only allowed operation is multiplication. We will use the abbreviation C_* for $C_{\{*\}}$. Here are the first few integers and their cost using $S = \{*\}$.

m	$C_*(a \circ b)$	$C_*(m)$
1	$C_*(1)$	1
2	$C_*(2)$	2
3	$C_*(3)$	3
4	$C_*(2*2)$	4
5	$C_*(5)$	5
6	$C_*(2*3)$	5
7	$C_*(7)$	7
8	$C_*(2*4)$	6
9	$C_*(3*3)$	6
10	$C_*(2*5)$	7

$C_*(m)$ can easily be calculated as it is equal to the sum of the prime factors of m (with repetition). This is shown by the following theorem and proof. First we prove some easy lemmas. The first one shows that the cost of a prime is itself.

Lemma 2.1. *If p is a prime number, then $C_*(p) = p$.*

Proof. If p is a prime, then it cannot be factored into $x * y$ where $x, y \in \mathbb{Z}^+$ and $x, y < p$ therefore $C_*(p) = \min\{p\} = p$. □

Lemma 2.2. *For integers $x, y \geq 2$, $x * y \geq x + y$*

Proof. Without loss of generality, we may assume $x \leq y$. Since $x \geq 2$, $x * y \geq 2y = y + y \geq x + y$. □

Theorem 2.3. *If $m > 1$ then $C_*(m)$ is equal to the sum of the prime factors of m (with repetition).*

Proof. Base Step, $m = 2 : C_*(2) = 2$ by Lemma 2.1.

Induction Step: Let m be an integer greater than 2. We suppose the statement is true for all positive integers n where $1 < n < m$ (Induction hypothesis).

If m is prime, by Lemma 2.1, $C_*(m) = m$, and the only prime factor is m, which proves the theorem in this case. If m is not prime, then

$$C_*(m) = \min(\{m\} \cup \{C_*(x) + C_*(y) : m = xy \text{ where } x, y \in \mathbb{Z}^+\}).$$

Since m is not prime, $m = xy$ for some $x, y \in \mathbb{Z}^+$ where $1 < x, y < m$. By Lemma 1.1, $C_*(x) + C_*(y) \leq x + y$, and by Lemma 2.2 $x + y \leq x * y$. So, $C_*(x) + C_*(y) \leq m$.

Let a and b be chosen so that

$$C_*(a) + C_*(b) = \min\{C_*(x) + C_*(y) : m = xy \text{ where } x, y \in \mathbb{Z}^+\}.$$

Then, $C_*(m) = C_*(a) + C_*(b)$.

By the induction hypothesis, the cost of a is equal to the sum of its prime factors, and the cost of b is also the sum of its prime factors. By the fundamental theorem of arithmetic, the multiset of the prime factors of m is the union of the multiset of the prime factors of a and the multiset of the prime factors of b.

Thus, $\mathrm{C}_*(m)$ is the sum of prime factors of a (with repetition) plus the sum of the prime factors of b (with repetition), which is equal to the sum of the prime factors of m (with repetition). □

3 Cost with $\mathrm{S} = \{+, *\}$

The cost with $\mathrm{S} = \{+, *\}$ is similar to $\mathrm{S} = \{*\}$, however including the + operator does make a considerable difference. Note that the explicit inclusion of m as a potential value in the definition of $\mathrm{C}_{\{+,*\}}(m)$ is not necessary while $+ \in \mathrm{S}$, since in this case,

$$\mathrm{C_S}(m) \le \mathrm{C_S}(m-1) + \mathrm{C_S}(1) = \mathrm{C_S}(m-1) + 1 \le \mathrm{C_S}(m-2) + 2 \le \ldots \le \mathrm{C_S}(1) + m - 1 = m$$

Listed below are a few examples comparing $\mathrm{S} = \{+, *\}$ to $\mathrm{S} = \{*\}$. A larger table can be viewed at the end of the paper.

m	$\mathrm{C}_*(a \circ b)$	$\mathrm{C}_*(m)$	$\mathrm{C}_{\{+,*\}}(a \circ b)$	$\mathrm{C}_{\{+,*\}}(m)$
1	$\mathrm{C}_*(1)$	1	$\mathrm{C}_{\{+,*\}}(1)$	1
2	$\mathrm{C}_*(2)$	2	$\mathrm{C}_{\{+,*\}}(1+1)$	2
3	$\mathrm{C}_*(3)$	3	$\mathrm{C}_{\{+,*\}}(1+2)$	3
4	$\mathrm{C}_*(2*2)$	4	$\mathrm{C}_{\{+,*\}}(1+3)$	4
5	$\mathrm{C}_*(5)$	5	$\mathrm{C}_{\{+,*\}}(1+4)$	5
6	$\mathrm{C}_*(2*3)$	5	$\mathrm{C}_{\{+,*\}}(2*3)$	5
7	$\mathrm{C}_*(7)$	7	$\mathrm{C}_{\{+,*\}}(1+6)$	6
8	$\mathrm{C}_*(2*4)$	6	$\mathrm{C}_{\{+,*\}}(2*4)$	6
9	$\mathrm{C}_*(3*3)$	6	$\mathrm{C}_{\{+,*\}}(3*3)$	6
10	$\mathrm{C}_*(2*5)$	7	$\mathrm{C}_{\{+,*\}}(2*5)$	7

3.1 Completely Multiplicative Numbers

For some numbers the cost is the same whether multiplication is the only allowed operation or with addition as well.

Definition 3.1. An integer m is called *completely multiplicative* if:

$$\mathrm{C}_{\{+,*\}}(m) = \mathrm{C}_*(m)$$

Corollary 3.1.1 (of Theorem 2.1)**.** *If a positive integer, m, is completely multiplicative, then $\mathrm{C}_{\{+,*\}}(m)$ is equal to the sum of its prime factors.*

Proof. This follows immediately from Definition 3.1 and Theorem 2.1. □

If a positive integer is completely multiplicative, then it can be written in the form $2^a * 3^b * 5^c$, as we will show after proving two easy lemmas. To simplify the notation, $C_{\{+,*\}}(m)$ will be written as $C(m)$ for the rest of this section.

Lemma 3.2. $C(m) \le C(m-1)+1$

Proof. $C(m) \le C(m-1)+C(1) = C(m-1)+1$ □

Lemma 3.3. $C(m) < m$ *for* $m \ge 6$

Proof. $C(6) = C(2)+C(3) = 5$. Repeated application of Lemma 3.1 shows $C(m) \le m-1$ for all $m \ge 6$. □

Theorem 3.4. *If a positive integer, m, is completely multiplicative, then m can be written in the form $2^a * 3^b * 5^c$.*

Proof. By Lemmas 2.1 and 3.2, no prime number greater than 6 is completely multiplicative. Therefore, the set of completely multiplicative prime numbers is $\{2,3,5\}$. If m had a prime factor $p > 5$, then $C(p) < p$ and $C(m)$ would be less than the sum of the prime factors of m, contradicting Corollary 3.1.1. Therefore, all the prime factors of m are in the set $\{2,3,5\}$ and m can be written in the form $2^a * 3^b * 5^c$. □

3.2 Bounds on $C_{\{+,*\}}(m)$

In this section we derive upper and lower bounds on $C(m)$. Since the binary (or decimal) representation of m expresses m in terms of $+$ and $*$, and since the length of the binary representation of m is bounded between $\log_2 m$ and $\log_2 m+1$, it is perhaps not surprising that we can derive logarithmic upper and lower bounds on $C(m)$. We begin with a lower bound.

Theorem 3.5 (Lower Bound)**.** $C(m) \ge 3\log_3 m$

Proof. Base Step: The inequality is true for $m = 1,2,3$. Now suppose $m \ge 4$

Induction Step: We suppose the inequality is true for all positive integers n where $1 \le n < m$ (Induction hypothesis)

Case 1. $C(m) = C(a)+C(b)$ *where* $m = ab$.

$$C(m) \ge 3\log_3 a + 3\log_3 b = 3(\log_3 a + \log_3 b) = 3\log_3(ab) = 3\log_3 m$$

Case 2. $C(m) = C(a)+C(b)$ *where* $m = a+b$.

Case 2.1. $a, b > 1$.

By Lemma 2.3, $ab \geq a + b$.

$$\mathrm{C}(m) \geq 3\log_3 a + 3\log_3 b = 3\log_3(ab) \geq 3\log_3(a+b) = 3\log_3 m.$$

Case 2.2. *Either $a = 1$ or $b = 1$. Without loss of generality suppose $b = 1$.*

$m = a+b$, so $m = a+1 = (m-1)+1$. Since $m \geq 4$ observe that $m > \frac{1}{3^{1/3}-1} + 1 \approx 3.26116$. Using basic algebra we simply this to $3^{1/3} > \frac{m}{m-1} = 3^{\log_3(\frac{m}{m-1})}$, then to $1/3 > \log_3\left(\frac{m}{m-1}\right) = \log_3 m - \log_3(m-1)$, and then finally,

$$3\log_3(m-1) + 1 > 3\log_3 m$$

Using this, observe that $\mathrm{C}(m) = \mathrm{C}(m-1) + 1 \geq 3\log_3(m-1) + 1 \geq 3\log_3 m$. □

Corollary 3.6. *Any integer of the form 3^a is completely multiplicative and $\mathrm{C}(3^a) = 3a$.*

Proof.

$$\mathrm{C}(3^a) \geq 3\log_3(3^a) = 3a$$

$$3^a = 3 * 3 * \ldots * 3 \ (a \text{ times}) \text{ so } \mathrm{C}(3^a) \leq a * \mathrm{C}(3) = 3a.$$

Therefore $\mathrm{C}(3^a) = 3a$ and 3^a is completely multiplicative. □

Now we prove upper bounds on $\mathrm{C}_{\{+,*\}}(m)$.

Theorem 3.7 (Upper bound)**.** *Let $n \in \mathbb{Z}^+$ and $a \in \mathbb{Z}$. If $\mathrm{C}(n) \leq 3\log_2 n + a$ for $n = k, k+1, \ldots, 2k-1$, then $\mathrm{C}(n) \leq 3\log_2 n + a$ for all $n \geq k$.*

Proof. The base cases $n = k, k+1, \ldots, 2k-1$ are true by the hypothesis.

Inductive Step, $n \geq 2k$:

If n is odd,

$$\mathrm{C}(n) \leq \mathrm{C}(n-1) + 1 = \mathrm{C}(2 * (\frac{n-1}{2})) + 1 \leq \mathrm{C}(2) + \mathrm{C}(\frac{n-1}{2}) + 1 = \mathrm{C}(\frac{n-1}{2}) + 3.$$

Since n is odd, $n \geq 2k+1$. Therefore, $n > \frac{n-1}{2} \geq k$, so we may apply the induction hypothesis to get

$$\mathrm{C}(\frac{n-1}{2}) + 3 \leq 3\log_2(\frac{n-1}{2}) + a + 3 = 3(\log_2(n-1) - 1) + a + 3 = 3\log_2(n-1) + a \leq 3\log_2 n + a.$$

If n is even,

$$\mathrm{C}(n) = \mathrm{C}(2 * \frac{n}{2}) \leq \mathrm{C}(\frac{n}{2}) + \mathrm{C}(2) = \mathrm{C}(\frac{n}{2}) + 2.$$

Since $n > \frac{n}{2} \geq k$ we may apply the induction hypothesis to get

$$\mathrm{C}(\frac{n}{2}) + 2 \leq 3\log_2(\frac{n}{2}) + a + 2 = 3(\log_2 n - 1) + a + 2 = 3\log_2 n + a - 1 \leq 3\log_2 n + a.$$

□

We then for $n \geq 1$ can observe $C(n) \leq 3\log_2 n + 1$ by checking the hypothesis to the previous statement. Using this method we can also derive more examples of this form such as $C(n) \leq 3\log_2 n - 1$ for all $n \geq 2$, and $C(n) \leq 3\log_2 n - 2$ for all $n \geq 6$.

Any of these examples implies the asymptotic result in the following corollary.

Corollary 3.8. $C(n) \in O(\log n)$.

3.3 Conjectures Regarding $C_{\{+,*\}}(m)$

The following conjectures are consistent with the values of $C(m)$ for $m \leq 1000$.

Conjecture 3.1. If m is of the form $2^a * 3^b * 5^c$ where $a, b, c \in \mathbb{Z}$, then $C(m) = 2a + 3b + 5c$ and m is completely multiplicative.

This conjecture is the converse of theorem 3.1, and a generalization of Corollary 3.2.1.

Corollary 3.2.1 was proved using the lower bound on $C(m)$. This method cannot be applied to 2 and 5. Using the change of base formula, we can write:

$$\begin{aligned} 3\log_3 m &= 3\frac{\log_2 m}{\log_2 3} \\ &= \frac{3}{\log_2 3}\log_2 m \approx 1.89\log_2 m \end{aligned}$$

Therefore, $C(m) \geq 1.89\log_2 m$. When $m = 2^a$ we can simplify this so $C(m) \geq 1.89a$. We also know $C(2^a) \leq 2a$, as 2^a can be written as $2 * 2 * \ldots * 2$ a times. This means

$$1.89a \leq C(2^a) \leq 2a$$

The small gap between $1.89a$ and $2a$ causes issues when one attempts to prove $C(2^a) = 2a$. Similar issues arise when one attempts to prove $C(5^a) = 5a$. Since Conjecture 3.1 implies both of these equations, it appears difficult to prove Conjecture 3.1.

To state the following conjecture, we use the notation $+1$ to denote the binary operation $a + b$ restricted to the cases where $b = 1$.

Conjecture 3.2. $C_{\{+,*\}} = C_{\{+1,*\}}$

In the positive integers less than 1000, the addition operation is only used in cases where $C(m) = C(m-1) + 1$. If this is always true, as stated in this conjecture, then it would be easier to compute $C(m)$, since other ways to use addition would not need to be checked.

This conjecture would be refuted if the following scenario occurs. Suppose there is a positive integer m and $m = x + y$, where $1 < x, y < m - 1$ and the costs of x and y are small enough to provide the least expensive way to describe m. More precisely, $C(x) + C(y) < C(m-1) + 1$ and $C(x) + C(y) < C(a) + C(b)$ for any factorization $ab = m$. This scenario does not occurs for $m \leq 1000$, but if it occurs for some larger m, then Conjecture 3.2 is false.

4 Cost with $S = \{+, *, -\}$

Now consider C_S, where $S = \{+, *, -\}$. Because the subtraction operation is allowed, we cannot compute $C_S(m)$ from the values of $C_S(n)$ for $n < m$. However, by Theorem 1.1 we know there is a single solution for the cost for every $m \in \mathbb{Z}^+$, and we may use the method of the proof of Theorem 1.1 to compute C_S. The first example that illustrates the difference is $m = 23$. While $C_{\{+,*,-\}}(n) = C_{\{+,*\}}(n)$ for $n < 23$, the pattern is broken at 23. Using only addition and multiplication we find $C_{\{+,*\}}(23) = C_{\{+,*\}}(22) + 1 = 10 + 1 = 11$. But we cannot stop there. Going on to 24, it turns out that $C_S(24) = C_{\{+,*\}}(24)$, and since 24 is completely multiplicative $C_S(24) = 2 * 3 + 3 = 9$.

$C_S(23)$ can be obtained using.

$$C_S(23) = C_S(24 - 1) \leq C_S(24) + C_S(1) = 9 + 1 = 10$$

In fact, we can show that $C_S(23) = 10$. Therefore, $C_S(23) = 10 < 11 = C_{\{+,*\}}(23)$

This example illustrates that $C_S(m)$ cannot be computed from the the values of $C_S(n)$ for $n < m$. However, the values of C_S can be calculated using the method described in the proof of Theorem 1.1, where the recursion is on the output of C_S rather than on m. The following table illustrates this algorithm.

$C(m)$	m
1	1
2	2
3	3
4	4
5	5, 6
6	7, 8, 9
7	10, 12
8	11, 14, 16, 18, 15, 13
9	17, 19, 20, 24, 21, 27
10	25, 22, 28, **23**, 26, 32, 30, 36

Each line of the table can be computed from the previous lines. For instance, in line 6, we are looking for numbers m that can be obtained by performing the allowed operations on a pair of numbers a and b where C(a)+C(b)=6. Without loss of generality, we can assume $a \leq b$, So either $C(a) = 1, C(b) = 5$ or $C(a) = 2, C(b) = 4$ or $C(a) = C(b) = 3$. Using addition, we get $1 + 5 = 6$, $1 + 6 = 7$, $2 + 4 = 6$ and $3 + 3 = 6$. The new number 7 is added to the list on line 6 since we have determined that $C(7) = 6$. Using multiplication we get $1 * 5 = 5$, $1 * 6 = 6$, $2 * 4 = 8$, and $3 * 3 = 9$. We have now determined that $C(8) = C(9) = 6$. Using subtraction we get $5 - 1 = 4$, $6 - 1 = 5$, $4 - 2 = 2$, $3 - 3 = 0$ but 0 is not positive and we have already determined the costs of 4, 5 and 2. The number 23 is highlighted in the

table since is obtained from line 9 and line 1 as 24-1=23. As discussed above, this is the first time a new number is generated by subtraction.

An interesting thing to note with this set of operations is that the absolute value of the difference between $C_S(m)$ and $C_S(m+1)$ is always at most 1. This can be written mathematically as:

Lemma 4.1. *For any* $m \in \mathbb{Z}^+$,

$$|C_{\{+,*,-\}}(m) - C_{\{+,*,-\}}(m+1)| \leq 1$$

Proof. By the proof of Lemma 3.1, $C_{\{+,*,-\}}(m+1) \leq C_{\{+,*,-\}}(m) + 1$. Similarly, since $m = (m+1) - 1$, $C_{\{+,*,-\}}(m) \leq C_{\{+,*,-\}}(m+1) + 1$. □

The next conjecture strengthens Conjecture 3.2 by including subtraction as well as addition. The conjecture is based on the observation that the subtraction operation is only used in cases where $C(m) = C(m+1) + 1$, for $m \leq 1000$. In section 3.3, the notation $+1$ was used to denote the binary operation $a + b$ restricted to the cases where $b = 1$. Here we also use the notation -1 to denote the binary operation $a - b$ restricted to the cases where $b = 1$.

Conjecture 4.1. $C_{\{+,*,-\}} = C_{\{+1,*,-1\}}$

5 Cost with $S = \{+, \wedge\}$

Definition 5.1. The $\wedge$ operator is used to represent exponentiation. Therefore $a \wedge b = a^b$.

Since exponentiation of a pair of positive integers can generate much larger values than multiplication, $C_{\{+,\wedge\}}(m)$ is often smaller than $C_{\{+,*\}}(m)$. However, the reduction in cost only occurs when m can be expressed as a nontrivial power or sum of powers.

It is interesting to note that the analog to Conjecture 3.2 (+1 Conjecture) does not apply to this set of operations. This can easily be shown by considering a number that is the sum of two nontrivial powers (and is not a nontrivial power itself). For instance, $24 = 8 + 16$.

$$\begin{aligned} C_{\{+,\wedge\}}(8) &= C_{\{+,\wedge\}}(2^3) \\ &= C_{\{+,\wedge\}}(2) + C_{\{+,\wedge\}}(3) \\ &= 5 \end{aligned}$$

$$\begin{aligned} C_{\{+,\wedge\}}(16) &= C_{\{+,\wedge\}}(2^4) \\ &= C_{\{+,\wedge\}}(2) + C_{\{+,\wedge\}}(4) \\ &= 6 \end{aligned}$$

With the cost of 8 and 16 defined, we can now see the cost of 24 is

$$\begin{aligned} C_{\{+,\wedge\}}(24) &= C_{\{+,\wedge\}}(8+16) \\ &= C_{\{+,\wedge\}}(8) + C_{\{+,\wedge\}}(16) \\ &= 5+6 \\ &= 11 \end{aligned}$$

On the other hand, $C_{\{+,\wedge\}}(23)+1 = 13+1 = 14 \neq C_{\{+,\wedge\}}(24)$ This example proves the following theorem.

Theorem 5.2. $C_{\{+,\wedge\}} \neq C_{\{+1,\wedge\}}$

If m is expressible as a nontrivial power a^b then it seems reasonable that the cost will make use of that exponentiation. This makes sense since exponentiation can be used to get a large integer from much smaller integers. The following Conjecture is consistent with the values of $C_{\{+,\wedge\}}(m)$ for $m \leq 1000$.

Conjecture 5.1. For any $a, b \geq 2$, $C_{\{+,\wedge\}}(a^b) = \min\{C_{\{+,\wedge\}}(x) + C_{\{+,\wedge\}}(y) : x^y = a^b\}$

6 Table of Costs

The table below shows values of the cost functions we have discussed. The last column shows the calculation of m used in computing $C_{\{+,*,-\}}(m)$. In the interest of space, the table is limited to $m \leq 50$. Larger tables and the Python code used to compute the values can be found at the website: https://www.github.com/mnorfolk03/cost.

m	C_{*}(m)	C_{+,*}(m)	C_{+,^}(m)	C_{+,* -}(m)	Calculation for C_{+,* -}(m)
1	1	1	1	1	1
2 (prime)	2	2	2	2	1 + 1
3 (prime)	3	3	3	3	1 + 2
4	4	4	4	4	2 * 2
5 (prime)	5	5	5	5	1 + 4
6	5	5	6	5	2 * 3
7 (prime)	7	6	7	6	1 + 6
8	6	6	5	6	2 * 4
9	6	6	5	6	3 * 3
10	7	7	6	7	2 * 5
11 (prime)	11	8	7	8	1 + 10
12	7	7	8	7	2 * 6
13 (prime)	13	8	9	8	1 + 12

m	C_{*}(m)	C_{+,*}(m)	C_{+,^}(m)	C_{+,* -}(m)	Calculation for C_{+,* -}(m)
14	9	8	10	8	2 * 7
15	8	8	11	8	3 * 5
16	8	8	6	8	2 * 8
17 (prime)	17	9	7	9	1 + 16
18	8	8	8	8	2 * 9
19 (prime)	19	9	9	9	1 + 18
20	9	9	10	9	2 * 10
21	10	9	11	9	3 * 7
22	13	10	12	10	2 * 11
23 (prime)	23	11	13	10	24 - 1
24	9	9	11	9	2 * 12
25	10	10	7	10	5 * 5
26	15	10	8	10	2 * 13
27	9	9	6	9	3 * 9
28	11	10	7	10	2 * 14
29 (prime)	29	11	8	11	1 + 28
30	10	10	9	10	2 * 15
31 (prime)	31	11	10	11	1 + 30
32	10	10	7	10	2 * 16
33	14	11	8	11	3 * 11
34	19	11	9	11	2 * 17
35	12	11	10	11	5 * 7
36	10	10	8	10	2 * 18
37 (prime)	37	11	9	11	1 + 36
38	21	11	10	11	2 * 19
39	16	11	11	11	3 * 13
40	11	11	12	11	2 * 20
41 (prime)	41	12	12	12	1 + 40
42	12	11	13	11	2 * 21
43 (prime)	43	12	12	12	1 + 42
44	15	12	13	12	2 * 22
45	11	11	13	11	3 * 15
46	25	12	14	12	2 * 23
47 (prime)	47	13	15	12	48 - 1
48	11	11	13	11	2 * 24
49	14	12	9	12	7 * 7
50	12	12	10	12	2 * 25

References

[1] Calhoun, W., Golomb, D.: The Cost of a Number (unpublished notes), 2013.

[2] Li, M., Vitányi, P.: An Introduction to Kolmogorov Complexity and Its Applications (4th ed.). Springer-Verlag, New York (2019).

Maxwell Norfolk
Bloomsburg University
mnorfolk03@outlook.com