Inserting plus signs and adding

Steve Butler **Ron Graham** **Richard Stong**

Abstract. For a fixed base b, we consider the following operation called "insert and add" on a natural number: Write the number as a string of digits and between some of the digits insert a plus sign; finally, carry out the indicated sum. If at least one plus sign is inserted, this results in a smaller natural number and so repeated application can always reduce the number to a single digit. We show that for any base b, surprisingly few applications of this operation are needed to get down to a single digit.

1. INTRODUCTION For a fixed base b consider the following operation that can be performed on a natural number.

Insert and Add. *First write the number down as a string of digits in base-b. Then insert plus signs, "+", between some of the digits (as many or as few as desired). Finally carry out the indicated addition in base-b to produce a new number.*

If during **insert and add** we use at least one plus sign, then the resulting number will be smaller. So for *any* given number, enough applications will allow us to get down to a single digit. One natural question becomes how many applications will be needed to get down to a single digit.

An obvious strategy is to insert every possible plus sign, so that **insert and add** takes a number and produces the sum of its digits. For example, in base-10 if we start with $31415926535897932384626433832795$, then adding the digits gives 155, which adding those digits gives 11, which adding those digits gives 2. So we only needed three applications to get to a single digit. We can do better though. For instance, if in the first step we replace "$+5+9+$" with "$+59+$" then we will get 200, which summing those digits gives 2. There are many other possible options that only need two steps, such as the following:

$$3141+592+65358+9793+23846+2643+3832+795 = 110000$$
$$1+1+0000 = 2$$

Each time we ended up with a final answer of 2. In fact, no matter how we apply the **insert and add** to get to the final digit, the final digit will always be the same. This is because the value is invariant modulo $(b-1)$, where b is the base. (This is the same principle which states a number is divisible by 9 if and only if the sum of its digits is divisible by 9.)

For small numbers it is not surprising that the summing the digits strategy only needs a few steps. For a typical number n, the sum of the digits is of order $\log n$, and it takes very few applications of the logarithm function to get down to a single digit. Nevertheless, one might expect that as the numbers grow very large that we would need to take more and more steps to get down to a single digit. This turns out to not be true!

Theorem 1. *In base-2 we can take any natural number to a single digit in at most* two *applications of* ***insert and add****. In base-b with $b \geq 3$ we can take any natural number to a single digit in at most* three *applications of* ***insert and add****.*

The original problem for base-2 comes from Gregory Galperin and was featured in the Puzzles Column of *Emissary* [**1**]. In what follows we show how to establish Theorem 1. Throughout this note, we will use a subscript notation to indicate a number is written in a particular base, i.e., $111_{(2)}$ is 7 written in base-2.

2. BASE-2 In base-2 the final number at the end of the process will always be 1, and in particular, the penultimate number must be at a power of 2 (i.e., $100\ldots0_{(2)}$). So to show that two applications suffice we must show how in one application we can get from any number to a power of 2.

Our approach will be to start by inserting all possible plus signs and then start removing some of the plus signs strategically to increase the current sum up to a power of 2. We will make use of the following observation where we use a "?" to indicate an unknown binary digit.

Observation. *Replacing "$+1+?+$" with "$+1?_{(2)}+$" increases the current sum by 1; replacing "$+1+0+?+$" with "$+10?_{(2)}+$ increases the current sum by 3; replacing "$+1+1+?+$" with "$+11?_{(2)}+$" increases the current sum by 4.*

When we combine three digits together, as done in the preceding observation, then the value of that three digit number is at least $7/3$ the value of the sum of the digits (i.e., $+1+1+1+$ to $+111_{(2)}+$ changed that portion of the sum from 3 to 7; the other possibilities are better). If we can then form many such "triples", we can more than double what we would get if we summed the digits. And between any number and its double is a power of 2 (which we want!). This suggests the following strategy.

Triples Strategy. *Given a number whose binary expansion has m total 1's where $2^k < m \le 2^{k+1}$, insert all possible plus signs in the expansion and perform the following as long as the resulting sum is $\le 2^{k+1}$: Find the left-most "$+1+$" and combine it with its two succeeding digits as in the observation.*

When triples can no longer be formed, then perform the following as long as the resulting sum is $\le 2^{k+1}$: Find the left-most "$+1+$" and combine it with its succeeding digit as in the observation.

As an example, the strategy does the following for $5495 = 1010101110111_{(2)}$.

$$\begin{array}{c} 1{+}0{+}1{+}0{+}1{+}0{+}1{+}1{+}1{+}0{+}1{+}1{+}1{=}\ 1001_{(2)}{=}\ 9 \\ 101\ \ {+}0{+}1{+}0{+}1{+}1{+}1{+}0{+}1{+}1{+}1{=}\ 1100_{(2)}{=}12 \\ 101\ \ {+}0{+}\ \ 101\ \ {+}1{+}1{+}0{+}1{+}1{+}1{=}\ 1111_{(2)}{=}15 \\ 101\ \ {+}0{+}\ \ 101\ \ {+}\ 11\ {+}0{+}1{+}1{+}1{=}10000_{(2)}{=}16 \end{array}$$

This strategy will *not* always get us to a power of 2. There are situations when this fails, i.e., we don't have enough digits left to combine. An example of this is $31 = 11111_{(2)}$, which must be handled separately.

Proposition 1. *If there are exactly five 1's in the binary expansion of a number, then in one application of **insert and add** we can get to a power of 2.*

Proof. If anywhere in the binary expansion there is $10?$, then insert plus signs between the digits leaving one occurrence of $+10?_{(2)}+$. By the observation this $+10?_{(2)}+$ increases the sum by 3, which takes 5 to 8.

This leaves us with the two possibilities, $111110_{(2)}$ and $11111_{(2)}$, which can be handled by $11_{(2)}{+}11_{(2)}{+}10_{(2)} = 1000_{(2)}$ and $1_{(2)} + 1111_{(2)} = 10000_{(2)}$. ∎

Proposition 2. *If the number of 1's in the binary expansion of a number is not 5, then the insertion of the plus signs given by the **triples strategy** will result in a power of 2.*

Proof. We will use "triple" to indicate the digits that the strategy will combine into three digit blocks, and "double" to indicate the digits that the strategy will combine into two digit blocks.

If the sum of the digits, m, is 1, 2 or 4, then the strategy does nothing as we are already at a power of 2. If $m = 3$, then we cannot form a triple (i.e., it would take the resulting sum over 4) so the strategy starts forming doubles. Since there are at least three 1's then we can form at least one double and by the observation that takes the sum to 4.

We may now assume that $m \geq 6$ and $2^k < m \leq 2^{k+1}$.

Claim. If $\frac{7}{3}(m-5)+5 > 2^{k+1}$, then this strategy gives a way to get to a power of 2.

To see this, suppose that q is the number of 1's used in our triples at some point in the strategy. Then by the remark following the observation, the current sum is at least $\frac{7}{3}q + (m-q)$ (i.e., $\frac{7}{3}q$ is a lower bound for the contribution from the triples, and then there are at least $m-q$ remaining 1's which each contribute at least 1 to the sum).

The strategy dictates that we stop forming triples while the current sum is $\leq 2^{k+1}$. Therefore, we must have stopped when $q < m-5$ (otherwise we get a contradiction). In particular, when we have stopped forming triples we will have at least *six* 1's which have not yet been combined.

There are only two reasons to stop forming triples; either we have run out of enough digits to form the next triple (which is impossible since we still have six 1's), or the next triple would put the resulting sum above 2^{k+1}. By the observation, forming a triple increases our sum by at most 4. So if forming the next triple puts us over 2^{k+1} then we are at most 3 away from 2^{k+1}. With six 1's remaining we can now form doubles to make up any remaining difference. In particular we can get to the next power of 2, establishing the claim. (Note we are using that the strategy always work with the leftmost available $+1+$ to ensure that there are no "isolated" 1's that can't be combined.)

Rearranging $\frac{7}{3}(m-5)+5 > 2^{k+1}$, we can conclude that the strategy works for $m > \frac{6}{7}2^k + \frac{20}{7}$. When we combine this with $m \geq 2^k + 1$, we can conclude that this holds for $m \geq 10$. For $m = 6, 7, 8$ the strategy would not be able to form a triple and so starts forming doubles, based on the number of 1's available we can always succeed in these cases.

That leaves us with $m = 9$, and to finish this off we consider the possible triples that the strategy gives us before we have to move to forming doubles.

Starting triples	Current sum	Remaining 1's
$11?_{(2)}$	13	≥ 6
$11?_{(2)}$ $10?_{(2)}$	16	≥ 4
$10?_{(2)}$ $11?_{(2)}$	16	≥ 4
$10?_{(2)}$ $10?_{(2)}$	15	≥ 5

In each case the number of 1's that remain are sufficient to put into doubles to increase the sum to 16. This finishes the case for $m = 9$ and also the proof. ∎

3. BASE-B WITH $B \geq 4$ The previous section highlights the basic approach for this type of problem. Namely, we look for a way to handle special "small" cases, and then show that for "large" cases there is a simple strategy that works. We will continue this approach by first showing that if our initial number n is small, then there is a simple strategy that works for **insert and add**.

Lemma 1. *Let $b \geq 4$ be our base. Then any number $n < 3b^2 - b - 1$ can be collapsed to a single digit in at most two applications of* ***insert and add****.*

Proof. First we observe that for $n = 1, 2, \ldots, 2b - 2$, we can apply the summing digits strategy to get to a single digit in one step. The first number for which the summing digits strategy fails to reach a single digit in two steps is $1(b-1)(b-1)_{(b)}$. However, for this number we can first do $1_{(b)} + (b-1)(b-1)_{(b)} = 100_{(b)}$ and then sum our digits as before.

The number $2(b-2)(b-1)_{(b)} = 3b^2 - b - 1$ is the next time where the summing digits strategy fails, establishing the lemma. ∎

This lemma is tight in that the number $2(b-2)(b-1)_{(b)}$ takes three applications of **insert and add**. More generally, we have the following.

Observation. *Let $b \geq 4$ be our base. Then $20\ldots0(b-2)(b-1)_{(b)}$ takes three applications of* ***insert and add*** *for* any *number of zeroes. In particular, there are infinitely many numbers that take three steps.*

As noted in the introduction, this operation is invariant modulo $(b-1)$, so at the end this number will produce 1. Now if it could be done in two application, the first application would have to get to a number of the form $10\ldots0_{(b)}$, i.e., a power of b. In particular, the last digit after the first step would have to be zero. However, for a number of this form the last digit in the application would come from $b-1$, $(b-1)+(b-2)$, $(b-1)+2$ or $(b-1)+(b-2)+2$ and none of these are 0 modulo b when $b \geq 4$. (For $b = 3$ we can get a 0 in the last digit and so this number can be handled in two applications.)

Now we see that if the sum of digits is small then we can apply Lemma 1 after first summing the digits. We next show that if the sum of digits is large then we can take one application of **insert and add** to get us to a number whose form can be finished using at most two applications of summing the digits.

Lemma 2. *Let $b \geq 4$ be our base and let n a number with the sum of its digits $m \geq b^2$. Then in one step, n can be collapsed to a number of the form $c0\ldots0de$ where $c \leq 2$ and $de_{(b)} \leq b^2 - 2b$.*

Proof. Let $n = (\ldots a_4a_3a_2a_1a_0)_{(b)}$ and let

$$A = \max\{a_1 + a_3 + a_5 + \cdots, a_2 + a_4 + a_6 + \cdots\}.$$

Since we did not use the last digit to compute A, we have $A \geq (m - (b-1))/2$. We now consider what happens if we combine in pairs so that the leading digits in the pairs sum to A (i.e., we insert plus signs to form two digit numbers so that all the leading digits come from even or odd positions depending on which gave us A). The sum total of this strategy will be

$$(b-1)A + m \geq \frac{(b-1)(m-b+1)}{2} + m = \frac{mb + m - (b-1)^2}{2} > \frac{mb}{2}.$$

(The last step is by our assumption that $m \geq b^2$.) If we now break these pairs one at a time (i.e., reinsert the plus sign), say from left to right, then the difference of the current sum at each step would be at most $(b-1)^2$. Therefore we have a sequence of ways to insert plus signs which go from m to $(b-1)A + m$ where the difference between two consecutive methods is at most $(b-1)^2$.

Now m is in an interval of the form $[b^t, 2b^t)$ or $[2b^t, b^{t+1})$ for some $t \geq 2$. However we have that $(b-1)A + m > bm/2$ cannot be in the same interval (here we use that $b \geq 4$). Therefore there will be some configuration of plus signs for which the sum, M, will exceed the top of the given range containing m by at most $(b-1)^2$. We now either have $2b^t \leq M < 2b^t + (b-1)^2$ or $b^{t+1} \leq M < b^{t+1} + (b-1)^2$, depending on which case we are in. This is exactly the sort of base-b representation as given in the statement of the lemma. ■

We now put everything together to show that at most three applications are needed. If the sum of digits $m < b^2$ then sum the singletons and apply Lemma 1 to the result, using at most three applications. On the other hand, if the sum of digits $m \geq b^2$ then by Lemma 2 in one step we will collapse to a number of the form $c0\ldots0de$ with $c \leq 2$ and $db + e \leq b(b-2)$. We have that $d + e \leq (b-3) + (b-1) = 2b - 4$, and so adding all the singletons gives a number of size at most $2b - 2$, which can be finished in one more application.

4. BASE-3 For base-3 we can again break the situation down into two cases, namely we directly establish what happens for cases when the sum of digits is small; otherwise, the sum of digits is large allowing for flexibility in **insert and add** so that we can readily finish in three applications. The specific details of this are not additionally enlightening and so we omit them here and refer interested readers to [**2**]. One thing that separates base-3 from larger bases is that while there are numbers that require three applications, there are only eleven of them!

Theorem 2 (Butler-Graham-Stong [2]). *In base-3 any natural number can be collapsed to a single digit in at most two applications of* ***insert and add*** except *for the following eleven:*

$$\begin{aligned}
1781 &= 2102222_{(3)} & 41065 &= 2002022221_{(3)}\\
3239 &= 11102222_{(3)} & 43981 &= 2020022221_{(3)}\\
3887 &= 12022222_{(3)} & 98657 &= 12000022222_{(3)}\\
11177 &= 120022222_{(3)} & 131461 &= 20200022221_{(3)}\\
14821 &= 202022221_{(3)} & 393901 &= 202000022221_{(3)}\\
33047 &= 1200022222_{(3)}
\end{aligned}$$

This was proved using a computer to exhaustively handle the cases when the sum of digits is small and using theory to handle the case when the sum of digits is large. Finding a simpler, non-computer proof, for the base-3 case is an open problem.

5. CONCLUSION The speed of **insert and add** is unexpected and can be used to surprise and amaze your friends and colleagues. The proof given in base-2 is algorithmic and gives a simple way to demonstrate the result.

One open problem is what happens if we want to reduce several numbers *simultaneously* to a single digit. Say we have numbers $n_1, n_2, \ldots, n_k$ and at each stage we insert plus signs in the same location in each number (we prefix numbers with 0's to make them the same length if needed). One easy observation is that the last application we always need is summing the digits which can be done simultaneously. So we can first work with n_1 and ignore the effect on the other numbers until we get to the penultimate step, then do the same with n_2' (what is now the second number after working with n_1), ..., and finally at the end we simultaneously sum the digits. In particular, in base-2 we never need more than $k+1$ steps while in base ≥ 3 we never need more

than $2k + 1$ steps. Can this be improved for large k? What if we add the assumption that the sum of digits is sufficiently large?

Another avenue for exploration is changing the perspective on the "cost" of **insert and add**. We looked at the cost in terms of the number of applications needed to get to the end. An alternative is to look at the number of times a plus sign was inserted (ignoring the number of applications). For a warmup on this problem you can show that the number given in the introduction of the paper can be reduced to a single digit using only *five* plus signs. We note that 10, 19, 118, 3187, and 3014173 are the smallest numbers in base-10 that require 1, 2, 3, 4, and 5 plus signs, respectively. For base-2 we have $2 = 10_{(2)}$, $3 = 11_{(2)}$, $6 = 110_{(2)}$, and $216 = 1101100_{(2)}$ are the smallest numbers that require 1, 2, 3, and 4 plus signs, respectively, while the smallest number which requires 5 plus signs is

$$268468497 = 10000000000001000000100010001_{(2)}.$$

A simple strategy of repeatedly inserting a plus sign in the middle and adding gives a bound of $\log \log n$, is this the truth?

REFERENCES

1. Elwyn Berlekamp and Joe Buhler, Puzzles Column, *Emissary*, Fall 2011, 9.
2. Steve Butler, Ron Graham and Richard Stong, *Collapsing numbers in bases 2, 3, and beyond*, in the Proceedings of The Gathering for Gardner 10. Available online at `http://www.math.ucsd.edu/~ronspubs/`.