Subtraction Without Borrowing

an Alternative Multi-Digit Procedure

Uri Geva, MathVentures, a Division of Ten Ninety

Document Version 1.25 (detailed)

Table of Contents

Editorial Note

For convenience sake, the text below identifies the numbers involved in a subtraction operation using the following notation:

67890	← N_1	(the *minuend*)
–12345	← N_2	(the *subtrahend*)
55545	← N_3	(the *result*)

Background

The standard multi-digit subtraction procedure (algorithm) that is taught to most, if not all, students requires from a person to be capable of subtracting any single-digit number from any number in the range of 1 to 18. For a person must be able to "borrow" a "ten" from the digit to the left of the current one in the minuend. In other words, a person must possess the following computational abilities and perform them mentally:

1. Working in single-digit columns from right to left, compare each digit of the subtrahend with the corresponding digit, the one above it, of the minuend.

2. Recognize whether or not the current digit in the subtrahend is larger than the corresponding digit, the one above it, in the minuend.

3. If it is not, then a person must
   - 3.1. Perform a single-digit subtraction; and
   - 3.2. Proceed to the next single-digit column and repeat Step 2.

4. If it is, then a person must
   - 4.1. Add ***10*** to the current digit of the minuend.
   - 4.2. Subtract ***1*** from the digit to the left of the current of the minuend.
   - 4.3. Perform a single-digit subtraction from the two-digit number in the range of ***10–18***, inclusively;
   - 4.4. Proceed to the next single-digit column and repeat Step 2.

5. The final result is when the person is done with the leftmost column.

For students who struggle with any of these requirements the procedure describe below provides an easier alternative. It required that the strongest subtraction skill is:

- Single-digit subtraction, where the minuend is larger or equal to the subtrahend.
- Multi-digit addition.

Most people, including students before they are taught multi-digit subtraction, possess these skills.

This procedure does not employ borrowing. Instead it replaces the source number with a substitute from which the subtraction can proceed without borrowing. After the subtraction is finished, the result is adjusted by adding the number that was used to generate the substitute source number.

A bonus of this procedure is the fact the actual subtraction can proceed from left to right just as traditionally it is done from right to left. In fact, as long as digits are subtracted in their columns and subtraction is carried out in each and every column, the order is no longer relevant.

Concept

If each and every digit of a minuend is greater than the corresponding digit of the subtrahend than there is no need for borrowing and subtraction proceeds one column at a time, from right to left or from left to right, in the form of single digit subtraction.. In this context the term *corresponding digits* means that the two digits have the same decimal value. In other words, when the two numbers are written one above the other and their digits are aligned, then the two digits form a single column.

Almost every minuend has at least one such a substitute minuend. It is the largest number, which (a) is smaller than the given minuend and (b) terminates with a continuous string of 9's and (c) each and every digit of it is larger or equal to every corresponding digit of the subtrahend.

Demonstration

Let see how it works by following a simple example. Say we want to solve the following subtraction problem:

```
 67693
-49856
```

The largest number that ends with a string of 9's and is smaller than the minuend is 67599. But its 5 (in the middle) is smaller than the corresponding 8 of the subtrahend below. So try for the next largest number 66999. It too fails the test for its second 6 is smaller than the 9 below it. The next largest number is 59999 and it passes the test. This is it.

It is easy to see that the difference between 59999 and 67693 is 7694. Because

```
  67693 –  59999
= 67693 – (60000 – 1)
= 67693 –  60000 + 1
=  7693          + 1
=  7694
```

Now, go ahead and do the simplified subtraction:

```
 59999
-49856
 10143
```

And finally add back the difference between the original minuend and the substitute one:

```
 10143
+ 7694
 17837
```

And, indeed, 17837 is the answer to the original subtraction problem.

You may be tempted to find a number that is **larger** than the original minuend, which satisfy the desired condition that each of its digits is larger than the corresponding digit of the subtrahend. While such a numbers are available they suffer from two problems. Let's look at our example.

```
  67693
 -49856
```

First, one number that is larger than the original minuend is

```
  69999
```

The difference between ***69999*** and the original number ***67693*** is

```
   69999
 - 67693
    2306
```

which is not as simple as 7693 + 1 = 7694, where ***7693*** is part of the original number.

Second, after you are done with the complete subtraction

```
  69999
 -49856
  20143
```

you still have to do **another subtraction** to adjust for the number ***2306*** that you added.

Every time we adjust any of the original numbers by addition, we must reclaim the original value by an inverse subtraction operation and vice versa. As it turns out, our procedure gets around this problem by accomplishing a subtraction operation that is extremely easy to perform. It employs the largest number that terminates with a continuous sequence of ***9*** 's that is **smaller** than the original minuend. This forms a number from the last few digits of the original minuend and adds ***1*** to it.

The Procedure

Let the original minuend be denoted by N_1 and its substitute minuend by N_1'. The digits of N_1' must satisfy the following rules:

1. N_1' ends with a continuous string of ***9*** 's, and

2. the value of the first digit to the left of the ending string of ***9***'s is one less than the value of the original digit of N_1, and

3. If N_1' has any more digits to the left of this digit (R2), then
   3.1. These digits are the original digits of N_1, and
   3.2. Each of these digit is larger or equal to the corresponding digit of the subtrahend, N_2.

The difference between N_1' and N_1 is always so easy to figure out that, practically speaking, it requires no computation. Let us denote this difference by N_{adj}. (For this is *the adjusting number*.) That is, $N_{adj} = N_1' - N_1$. Once the simplified subtraction is completed, the result must be adjusted by adding to it N_{adj}, yielding the correct answer for the original subtraction problem.

Do This

The procedure consists of just 4 simple steps. (For simplicity we do not discuss here the various cases and how to handle the exceptions that may occur and other details. For a more rigorous definition see the appendix Formal Algorithm.) There are two ways to proceed, resulting with slightly two different procedures.

Start at the Right and Proceed Left

Starting with the right-most digit do the following: LR

1. Test it and every digit to its left whether it is larger than the corresponding digit of N_2.
   1.1. If it and all the digits to its left pass the test in S1, then you are done. Move left to the next digit and proceed to Step S1.3.
   1.2. If not, convert this digit to ***9*** (if it is ***9***, leave it as is) and move left to the next digit, then repeating Step S1.
   1.3. This digit, denoted by n, is the first digit of N_1 that you did not have to convert to ***9*** (it may have been 9). Replace n with $n-1$. (So it were $n=9$ then $n-1=8$.)

   Proceed to Step S2.

Start at the Left and Proceed Right

1. Starting with the left-most digit in N_1 that is smaller than or equal to the corresponding digit in N_2, replace it and each and every digit to its right with a ***9***. And if the first digit to its left is n, replace this digit with $n-1$. Let this new number be denoted by in N_1'. For clarity, cross out the original number N_1.

   Proceed to Step S2.

Proceeding to the Finish

2. Above the row of ***9***'s and in square brackets write the number that equals to the number consisting of the original digits plus ***1.***

3. Subtract N_2 from N_1'.

4. Add to the result of the subtraction in Step S3 the number in the square brackets from Step S2. This is your answer. N_1 is larger than the corresponding digit of N_2, the subtraction can proceed from left to right just as traditionally it is done from right to left.

Examples

The following examples illustrate various ways to work out a subtraction using this procedure. We tried to show how to do it in your head but… you'll have to figure this one out yourself.

Example 1

Given	Step 1	Step 2	Step 3	
		[5678+1]=	[5679]	
	39999		39999	
45678	~~45678~~		~~45678~~	
–26789	–26789		–26789	
			13210	
			+ 5679	← Step 4
			18889	

Notes: In the following two examples, **bold face** is used to identify the numbers used in each step.

[Source: See reference Sheltonian Subtraction.]

This example illustrates an alternative ordering of the numbers.

	Example 2		**Example 3**
Given:		**1287432091362772643**		**1280432091362002609**
	-	**35637828321289176**	-	**130637828321289999**
Step 1	-	~~1287432091362772643~~		~~1280432091362002609~~
	-	**1286999999999999999**		**1279999999999999999**
	-	35637828321289176	-	130637828321289999
Step 2		**[432091362772643+1]**		~~1280432091362002609~~
		~~1287432091362772643~~		**[0432091362002609+1]**
		1286999999999999999		1279999999999999999
	-	35637828321289176	-	130637828321289999
Step 3		[432091362772644]		~~1287432091362772643~~
		~~1287432091362772643~~		[0432091362002610]
		1286999999999999999		**1279999999999999999**
	-	**35637828321289176**	-	**130637828321289999**
		1251362171678710823		**1149362171678710000**
Step 4		**1251362171678710823**		**1149362171678710000**
	+	**432091362772644**	+	**0432091362002610**
		1251794263041483467		**1149794263040712610**

Why It works

According to the common subtraction procedure, when you want to subtract the subtrahend from the minuend, you must compare each and every digit of the subtrahend with the corresponding digit of the minuend. If the subtrahend's digit is larger than that of the minuend, you must perform a borrowing procedure on the minuend.

However, you never have to do any borrowing if the minuend is a number having digits each of which is larger than the corresponding digit of the subtrahend. It turns out that, for almost every minuend, you can easily find such a number and, moreover, finding it and adjusting for using it is so simple that it requires practically no computation.

This number is the largest possible that ends with a string of 9's yet it is still smaller than the original minuend. Once you find such a number put it to a simple test: Is each and every of its digits is equal or larger than the corresponding digit of the subtrahend? If so, this is it; if not, try for the next number.

For example, if the minuend is a number between 5,000 and 5,998 then, for most subtrahends, you can replace it with the number 4,999. (In this case the subtrahend should not be greater than 4,998.) Now, the difference between the original number and the substitute one always equals to the number consisting of the digits that were turned into 9's plus1. In the example above, it will be whatever was the minuend value above 5,000 plus 1.

Let's make it concrete. To subtract:

```
  5621
 -4756
```

First change 5671 to 4999 and note that the difference between the two numbers is 621 + 1 = 622. That is,

```
   5621 -  4999
 = 5621 - (5000 - 1)
 = 5621 -  5000 + 1
 =  621         + 1
 =  622
```

Now perform the simplified subtraction:

```
  4999
 -4756
   243
```

Finally add back the number by which you adjusted the original minuend to get the second one:

```
   243
 + 622
   865
```

This is the answer to the original subtraction problem.

Appendixes

Formal Algorithm

The formal procedure requires the following steps. (The digit conversion from N_1 to N_1' is described in only one direction, from left to right; the right-to-left direction is just as simple to specify.)

1. Find the left-most digit of N_1 that is either

    1.1. smaller than the digit in the same column of N_2, or

    1.2. equal to the digit in the same column of N_2 and the number that is formed by the digits that are immediately to its right is not larger than the number formed by the corresponding digits of N_2.

The result of this step is that you are assured that each and every digit of N_1 that is to the left of this digit is **greater** than the corresponding digit in the same column of N_2.

If $N_2 > N_1$, change the subtraction order, that is perform $N_2 - N_1$, revise the notations accordingly and restart the process of this algorithm. However, at the end, you must assign to the result the negative sign.

2. Replace this digit and each and every digit to its right with a ***9.*** (If the digit is already 9, leave it unchanged.)

3. If the first digit to its left of this digit is n, the replace it with $n-1$.

Note. Steps 1 through 3 can be simplified as follows: Let the left-most digit of N_1, be n. There are two possible cases:

(a) n is larger than the corresponding digit of N_2. In this case replace each and every digit to its right by a ***9*** and replace n by $n-1$.

(b) n is not larger than the corresponding digit of N_2. In this case, either

(b.1) $N_2 > N_1$, which means that the result of the subtraction will have a negative value. Therefore, reverse the subtraction order of N_1 and N_2, that is subtract N_1 from N_2, and restart the procedure, adjusting the notations accordingly. At the end, you must assign to the result the negative sign.

(b.2) By examining the digits to the right of n, apply Step (a) to the first digit that is larger than the corresponding digit of N_2.

If these 3 steps are simplified so, then each of the subtraction and addition steps described below will involve more digits. However, the difficulty level of these operations themselves remains unchanged.

4. Let us denote this new number N_1' and for clarity, cross out the original number N_1.

5. Above the ***9***'s write the original number in square brackets. The purpose of the square brackets is merely identification of this number.

6. Add ***1*** to the number you wrote in Step 4. Let us denote this number N_{adj}.

   6.1. If the last digit is smaller than 9, just increment it by 1;

   6.2. If the last digit is 9, change it to 0 and

      6.2.1. Repeat Steps 6.1 and 6.2 with respect to the next digit to the left.

7. Subtract the number N_2 from the number N_1'. Let us denote the resulting number N_3'.

8. Add the number N_{adj} to the number N_3'. The result is N_3. Or,

1. $$N_3' + N_{adj} = N_3$$

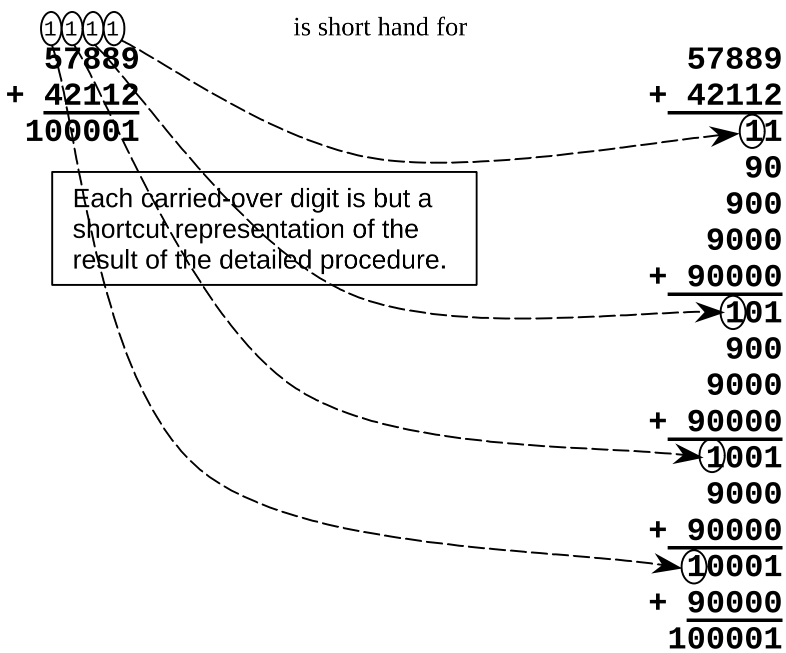
That is, the ordinary subtraction procedure is replaced with a simpler one followed by an addition:

$$\boxed{N_1 - N_2 = N_1' - N_2 + N_{adj} = N_3.}$$

Notes

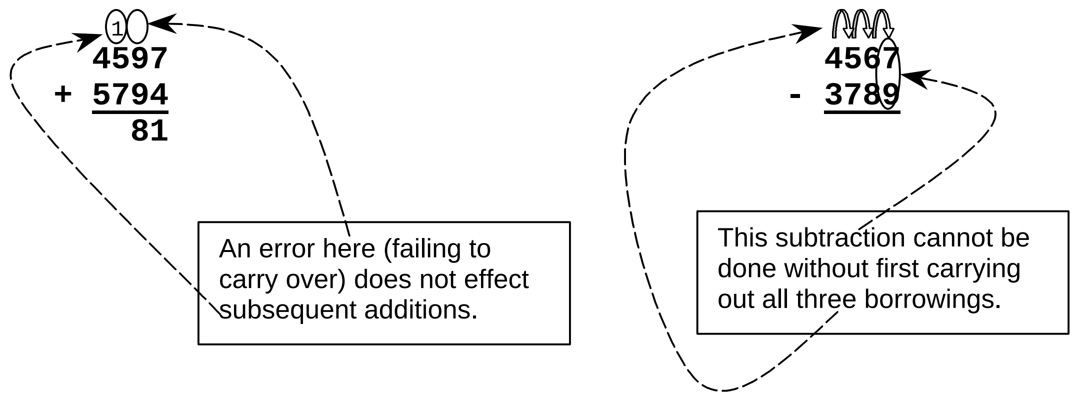
- **Borrowing vs. carryover.** Notice that the final addition step of this method often requires carryover. However, Carryover is significantly simpler than borrowing. On the other hand, borrowing is not the inverse of carryover for several reasons.

    1. Carryover is nothing but a procedural shortcut of the detailed addition procedure. For example, the following addition with carryover



    2. Carryover requires a single digit addition (assuming only two numbers are involved in either addition or subtraction. See Item 4 for more about this point.) Subtraction requires two operations subtracting ***1*** from the higher-order digit of the minuend and adding ***10*** to the current digit of the minuend.

3. A sequence of cascading borrowing may have to be executed in order to perform a single columnar subtraction. During addition, when carryover is required for each column, even if one addition causes a borrowing chain, each single addition depends only on the one that immediately precedes it.



If one asserts that in this example each subsequent borrowing is only necessary when it is time to perform the corresponding columnar subtraction operation, then consider the following subtraction that is requires completing a chain of borrowing before subtraction can commence:

```
  4007
- 3989

      1
      3007
    - 3989

           1
          3907
        - 3989

                1
              3997
            - 3989
```

4. The result of the three preceding differences is a significant operational difference: While addition is routinely carried on more than two numbers at once, subtraction is executed by subtracting only one number from another at a time. For example,

```
   63                  63                           63
+  38   to subtract  – 38   the common practice is – 38
+  21                – 21                           25
  122                                             – 21
                                                     4
```

- A teacher mentioned to me that a Portuguese had shown her this method and she thought it was taught in Portugal. [August 15, 2002]
- Other methods of subtraction without borrowing:
  - Sheltonian Subtraction requires competence working with negative numbers.

- Various other methods are based on splitting the two numbers involved in the subtraction into 2-digit columns. Once this is done, subtraction proceed within these 2-digit columns. This requires the ability to subtract **any** 2-digit number from **any** 2-digit number, including such cases where the unit digit of the subtrahend is larger from the corresponding unit digit of the minuend. These methods get around the fact that this requires borrowing by demanding from the person who perform the operation to *memorize* or *know* the difference between every 2-digit numbers so actual subtraction is not required.

Reference

- "Sheltonian Subtraction", by Dana Doe and Pat Doe, http://math.dartmouth.edu/~doyle/docs/sub/sub.pdf, Copyright © 2001 Peter G. Doyle**,** doyle@math.dartmouth.edu**,** Math Department, Dartmouth College, Hanover, NH 03755.