# Counting

Joint Mathematics Meetings
New Orleans, LA
January 6, 2011

Patricia Baggett
Dept. of Mathematical Sciences
New Mexico State University
Las Cruces, NM 88003-8001
baggett@nmsu.edu

Andrzej Ehrenfeucht
Computer Science Dept.
University of Colorado
Boulder, CO 80302-0430
andrzej.ehrenfeucht@colorado.edu

http://www.math.nmsu.edu/~breakingaway

# Outline

1. Introduction
2. An algorithm: Counting by partitions
    - Description
    - Procedure
    - Time complexity of the algorithm
3. Counting taught in schools
    - Time complexity of the school algorithm
4. Final remarks

# 1. Introduction

We present an algorithm, “counting by partitions”, which takes as input a set of tokens (movable small objects) and returns its number (how many) in decimal notation.

The algorithm consists of two parts.
The first part (which takes most of the time) doesn’t require any math knowledge.
The second part requires knowledge of a number system, and of a doubling algorithm.

We compare “counting by partitions” to the algorithm currently taught to children in the early grades.

The algorithm that we are presenting is not new. And any speculation that it may have been used in preliterate societies that had well-developed number systems (the early Sumerians, Incas) is also not new.

We found a variant of the algorithm in an 1827 book by Prof. John Leslie, *The Philosophy of Arithmetic*, published in Edinburgh. Prof. Leslie describes how "savages" may have developed arithmetic.

THE

# PHILOSOPHY

OF

# ARITHMETIC;

EXHIBITING

A PROGRESSIVE VIEW

OF THE

THEORY AND PRACTICE OF CALCULATION,

WITH

AN ENLARGED TABLE OF THE PRODUCTS OF NUMBERS UNDER *ONE HUNDRED*.

BY

JOHN LESLIE, F. R. S. E.

PROFESSOR OF MATHEMATICS IN THE UNIVERSITY OF EDINBURGH.

EDINBURGH:

*Printed by Abernethy & Walker,*

FOR ARCHIBALD CONSTABLE AND COMPANY, EDINBURGH; AND LONGMAN, HURST, REES, ORME, AND BROWN LONDON.

1817.

Let us endeavour to trace the steps by which a child or a savage, prompted by native curiosity, would proceed in classing, for instance, *twenty-three* similar objects.—

1. He might be conceived to arrange them by successive *pairs*. Selecting *twenty-three* of the smallest shells or grains he could find, he might dispose these in *two* rows, containing each *eleven* counters, and *one* over.

• • • • • • • • • • •
• • • • • • • • • • • •

# 2. An algorithm, “Counting by partitions”

## Description

A collection of tokens can be partitioned into two equal parts by the method “one for you and one for me”, or “two for you and two for me”.  In each case, a person may use one or both hands. The result consists of two collections that have the same number of tokens, and possibly one object left unassigned, a “leftover” or “remainder”.

One can repeat this process on one of the two resulting collections, until the collection is small enough to see its number at a glance (for example, two or three tokens).

A person doing this task needs only to record, for each iteration, whether there is a leftover or not, and to record the number of tokens left at the end. It doesn't require any mathematical knowledge or linguistic literacy.

A whole record may look like this:

| | |
|---|---|
| o | no leftover |
| \| | one leftover |
| \| | one leftover |
| o | no leftover |
| \| | one leftover |
| /// | three tokens left at the end |

Do you see that this record came from partitioning 118 tokens?

Actual procedure for partitioning 118 tokens

Equipment needed: Three bowls, which we call L, M, and R (Left, Middle, and Right) and material for recording.
To begin, put all tokens in the middle bowl.

Move the tokens by taking one or two tokens from the middle container in each hand and putting them simultaneously into the two side containers. You may look at what you are doing, or do it just by touch.

| No. of tokens in containers: | L | M | R | Record written horizontally: |
|---|---|---|---|---|
| 1st cycle | 0 | 118 | 0 | |
| | 1 | 116 | 1 | |
| | 2 | 114 | 2 | |
| | … | …… | … | |
| | 58 | 2 | 58 | |
| | 59 | 0 | 59 | o |

After the 1st cycle:

| | L | M | R | |
|---|---|---|---|---|
| $2^{nd}$ cycle | 59 | 59 | 0 | |
| | 60 | 57 | 1 | |
| | …. | …. | …. | |
| | 88 | 1 | 29 | o \| |
| $3^{rd}$ cycle | 89 | 29 | 0 | |
| | 90 | 27 | 1 | |
| | …. | …. | …. | |
| | 103 | 1 | 14 | o \| \| |
| $4^{th}$ cycle | 104 | 14 | 0 | |
| | 105 | 12 | 1 | |
| | …. | …. | …. | |
| | 111 | 0 | 7 | o \| \| o |

After the 4th cycle:

| | L | M | R | |
|---|---|---|---|---|
| $5^{th}$ cycle | 111 | 7 | 0 | |
| | …. | …. | …. | |
| | 114 | 1 | 3 | o \| \| o \| |
| $6^{th}$ cycle | 115 | 3 | 0 | |
| | 118 | 0 | 0 | o \| \| o \| /// |

At the beginning of the $6^{th}$ cycle:

## Computation

To compute the number of elements in a collection on the basis of its record requires two operations:

Double: $k \to 2k$, and

Double-and-one: $k \to 2k + 1$

Computation is carried out in the direction opposite to the way the record is kept. The person doing the computation must be mathematically literate.

In the example above, the record was

o | | o | ///

The computation is carried out (from right to left) as follows:

$$3 \to 7 \to 14 \to 29 \to 59 \to 118$$

# Time complexity of the algorithm

The main portion of time is spent on partitioning the current collection. So when a person uses both hands, holding one token in each, he/she needs k/2 moves to split a collection of k elements.

The total number of moves for creating a record for a collection of n elements is $n/2 + n/4 + n/8 + \ldots < n$.
So approximately n moves are needed to create a record of n tokens.
In any case the total time is proportional to n.
The time to count n tokens = C*n.
We have experimented with counting up to 1000 tokens and found that $C \approx 2$ seconds/token.

# 3. Counting taught in schools

Students are taught to recite the names of positive whole numbers from one to at least 20. (This is called "verbal counting".)

At the same time, or later, they are taught to synchronize verbal counting with touching, pointing to, or moving the objects that are counted.

It is stressed that exactly one object has to be associated with each number.

The last number spoken during this process is the total number of elements in a counted collection.

In practice, counting a collection of small objects (for example, pebbles) may look as follows.

A student has some pebbles on a paper plate and another empty plate. He/she moves one pebble from the first plate to the second, and at the same time counts silently, or aloud, one, two, three, … . After all pebbles are moved, the student writes down the last number he/she mentioned. This method may be "improved" by moving more than one object at a time (for example, two) and "skip counting", two, four, six, … .
Using this algorithm requires using a number system fluently, up to the number of objects being counted. This in turn indicates at least a reasonable knowledge of arithmetic.

# Time complexity of the school algorithm

The time spent on moving n objects is proportional to n. But counting verbally is proportional to n*log(n). The log(n) factor enters because the lengths of names of numbers are roughly proportional to the number of decimal digits. (Compare the number of syllables in “five”, “thirty-five”, and “two hundred thirty-five”.)

But the main limitation of the school algorithm lies in the fact that it puts a big stress on the person's working memory.  Any lapse of attention may create an error, such as … 87, 87, 88, …. or … 78, 89, … .

And the chance of such errors increases with the time spent on counting.  (The numbers spoken earlier interfere with the numbers spoken later.)

So in practice, the school algorithm is limited to counting only small collections (up to 50 objects), and even then it is rather error-prone.

# 4. Final remarks:
# A question and a comment

## Question

Why is it that an algorithm that is inferior to the algorithm by partitions (and to many other counting algorithms) plays such a prominent role during the first four years (K-3) of teaching mathematics?

## Comment

We have a rather broad knowledge of the number systems that were used by the Sumerians and Incas. But we have no knowledge of their counting algorithms.