Geometric Approach to Error-Less Counting

Olga Kosheleva
Department of Teacher Education
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
olgak@utep.edu

Abstract

To decrease counting errors, it is often reasonable to arrange the counted objects into rectangles and/or parallelepipeds. In this paper, we describe how to design optimal arrangements of this type.

A geometric approach to counting: a description. Once, we went to buy 24 cups of easy-to-prepare "instant lunch" soup. When we counted these cups ourselves, we had to count several times to make sure that we did not make a mistake. A salesperson counted them very easily: she grouped them into a nice parallelepiped of length 4, width 3, and height 2, and then multiplied these three numbers to get $4 \times 3 \times 2 = 24$.

Why it is interesting. Of course, everyone knows that the volume of a parallelepiped is the product of the sides, and that, similarly, the area of a rectangle is the product of its sides. The corresponding geometric "area" approach is one of the main methods of teaching multiplication of integers (and fractions); see, e.g., Chapters 10 and 13 from [5], [1, 2, 3, 4], and references therein. What was interesting is that the salesperson used the same geometric arrangement not for multiplication, but for error-less counting.

What we do in this paper. We explain why this geometric approach reduced the errors, and what is the best way to use this approach if we want to decrease counting errors.

Main idea. Every time we perform an arithmetic operation, there is a probability that we make a mistake.

- When we count a new object – i.e., when we add one to the previous total – we can make a mistake.
- When we multiply two numbers, we can make a mistake.

- When we add two numbers, we can make a mistake.

Thus, to minimize the probability of a mistake, we should minimize the number of arithmetic operations.

This idea explains why the geometric approach leads to fewer errors. If we simply count 24 objects one after another, we start with the first one and then perform $24 - 1 = 23$ arithmetic operations – operations of adding 1. Thus, we have 23 chances of making a mistake.

On the other hand, once we have a $4 \times 3 \times 2$ arrangement, we need to count all the sides and then multiply the results. Counting 4 elements in the length requires $4 - 1 = 3$ computational steps, counting 3 elements in the width requires $3 - 1 = 2$ steps, counting 2 elements in the height requires $2 - 1 = 1$ steps, and we also need two multiplications. Thus, totally, we need $3 + 2 + 1 + 2 = 8$ arithmetic operations. Since $8 \ll 23$, we have much fewer chances to make a mistake.

General case: towards a precise formulation of the problem. With $n = 24$ objects, we were lucky, since 24 can be easily represented as a product of three numbers. With a prime number like $n = 23$, this would not have been possible – but we can still arrange $n = 23$ into several parallelepipeds and thus, decrease the total number of computational steps.

When the number of objects is small, instead of (or in addition to) parallelepipeds, it may be useful to combine them into rectangles.

If we know that we need to count n objects, what is the best arrangement that minimizes the number of computational steps?

For each parallelepiped of sizes $a_i \times b_i \times c_i$, counting the volume of each parallelepiped means that

- we first count the length a_i; this requires $a_i - 1$ counting steps;
- then, we count the width b_i; this requires $b_i - 1$ counting steps;
- after that, we count the height c_i; this requires $c_i - 1$ counting steps;
- then, we need two more multiplication steps: to multiply a_i by b_i and to multiply the result by c_i.

Thus, overall, we need $(a_i - 1) + (b_i - 1) + (c_i - 1) + 2 = a_i + b_i + c_i - 1$ steps.

Similarly, for each rectangle of sizes $a_i \times b_i$, counting the area of this rectangle means that

- we first count the length a_i; this requires $a_i - 1$ counting steps;
- then, we count the width b_i; this requires $b_i - 1$ counting steps;
- then, we need one multiplication step: to multiply a_i by b_i.

Thus, overall, we need $(a_i - 1) + (b_i - 1) + 1 = a_i + b_i - 1$ steps.

Once we have s such sets (parallelepipeds and rectangles), and we have computed the number of elements n_i in each of them, we need $s-1$ additions to compute the sum $n_1 + n_2 + \ldots + n_s$. Thus, we arrive at the following definitions.

Definition 1. *By a* parallelepiped*, we mean a triple $P = (a, b, c)$ of positive integers. For each parallelepiped $P = (a, b, c)$,*

- *the product $a \cdot b \cdot c$ is called its* volume *and denoted by $N(P)$;*
- *the expression $a + b + c - 1$ is called its* number of computational steps *and denoted by $C(P)$.*

Definition 2. *By a* rectangle*, we mean a pair $R = (a, b)$ of positive integers. For each rectangle $R = (a, b)$,*

- *the product $a \cdot b$ is called its* area *and denoted by $N(R)$;*
- *the expression $a + b - 1$ is called its* number of computational steps *and denoted by $C(R)$.*

Definition 3. *By an* arrangement *$S = (S_1, \ldots, S_s)$, we mean a finite list consisting of parallelepipeds and rectangles. For each arrangement $S = (S_1, \ldots, S_s)$,*

- *the sum $\sum\limits_{i=1}^{s} N(S_i)$ is called its* number of elements *and denoted by $N(S)$;*
- *the value $\sum\limits_{i=1}^{s} C(S_i) + s - 1$ is called its* number of computational steps *and denoted by $C(S)$.*

Definition 4. *For a given positive integer n, we say that an arrangement $S = (S_1, \ldots, S_s)$ is* optimal *if it has n elements $N(S) = n$, and among all the arrangements with n elements, it has the smallest number of computational steps $C(S)$. The number of computational steps in the optimal arrangement will be denoted by $C(n)$.*

Formulation of the computational problem. Given an integer n, find the optimal arrangement and the corresponding value $C(n)$.

In principle, one can use exhaustive search, but this is not practical. In principle, for each n, we can try all possible arrangements – there are only n elements to arrange, so there is a finite number of possible arrangements with n elements. However, for large n, there are too many possible arrangements to be practically possible to enumerate them all. Thus, we need a more efficient algorithm. Such an algorithm will be presented in this paper.

Idea. Before we target the problem of computing $C(n)$ for general arrangements, let us first solve the simpler problems in which arrangements are limited. We will then use the solutions to these simpler problems to solve the more general problem.

Simple counting. For a simple counting, we need $C_0(n) = n - 1$ computational steps.

First simplified problem: a single rectangle. Let us first consider the simplest possible case, when we are only allowing one set, and this set has to be a rectangle. For each n, all such possible rectangles correspond to values a and b for which $a \cdot b = n$, i.e., to values a that divide n: $a|n$; the value b is equal to n/a and is, thus, uniquely determined by the value a. The total number of computational steps in this arrangement is equal to $a + (n/a) - 1$. Thus, for each n, the smallest possible number $C_1(n)$ of computational steps in such an arrangement is equal to

$$C_1(n) = \min_{a:a|n} \left(a + \frac{n}{a} - 1 \right). \quad (1)$$

By enumerating all possible divisors of each integer n, we can therefore compute $C_1(n)$. For example, the integer $n = 10$ has 4 divisors $a = 1$, $a = 2$, $a = 5$, and $a = 10$, so we have

$$C_1(10) = \min(1 + 10 - 1, 2 + 5 - 1, 5 + 2 - 1, 10 + 1 - 1) = \min(10, 6, 6, 10) = 6,$$

with the optimal rectangle 2×5.

The values $C_1(n)$ corresponding to the first 30 integers are as follows:

n	1	2	3	4	5	6	7	8	9	10
$C_1(n)$	1	2	3	3	5	4	7	5	5	6
n	11	12	13	14	15	16	17	18	19	20
$C_1(n)$	11	6	13	8	7	7	17	8	19	8
n	21	22	23	24	25	26	27	28	29	30
$C_1(n)$	9	12	23	9	9	14	11	10	29	10

The corresponding optimal rectangles are:

1	2	3	4	5
1×1	1×2	1×3	2×2	1×5
6	7	8	9	10
2×3	1×7	2×4	3×3	2×5
11	12	13	14	15
1×11	3×4	1×13	2×7	3×5
16	17	18	19	20
4×4	1×17	3×6	1×19	4×5
21	22	23	24	25
3×7	2×11	1×23	4×6	5×5
26	27	28	29	30
2×13	3×9	4×7	1×29	5×6

Comment. One can easily check, by taking the derivative and checking its sign, that the function $a + (n/a) - 1$ decreases for $a \le \sqrt{n}$ and increases for $a \ge \sqrt{n}$. Thus, to find its smallest possible value, it is not necessary to check all divisors a of n: it is sufficient to only check the divisors which are the closest to $\sqrt{n}$. The divisor which is the closest to $\sqrt{n}$ from the right is equal to n divided by the closest divisor from the left, so they lead to the exact same rectangle.

Thus, it is sufficient to take, as a, the largest divisor which is still smaller than or equal to $\sqrt{n}$. For example, for $n = 1$, we have $\sqrt{10} = 3.\ldots$, so out of 4 possible divisors $a = 1$, $a = 2$, $a = 5$, and $a = 10$, we should take $a = 2$.

Second simplified problem: a single parallelepiped. Let us now assume that we allow a single set, and this set should be a parallelepiped. For each n, all such possible parallelepipeds correspond to values a, b, and c for which $(a \cdot b) \cdot c = n$. The corresponding values c have to divide n. The product $a \cdot b$ is then equal to n/c and is, thus, uniquely determined by the value c. The total number of computational steps in this arrangement is equal to $c + (a + b - 1)$.

Once c is fixed, and thus, the product $n' = a \cdot b$ is fixed (as $n' = n/c$), the smallest possible number of computational steps is attained when the value $a + b - 1$ is the smallest among all pairs (a, b) for which $a \cdot b = n'$.

We already know, from the previous problem, that this smallest number of computational steps is equal to $C_1(n') = C_1(n/c)$. Thus, for each n, the smallest possible number $C_2(n)$ of computational steps in such an arrangement is equal to

$$C_2(n) = \min_{c:c|n} \left(c + C_1\left(\frac{n}{c}\right)\right). \tag{2}$$

By enumerating all possible divisors c of each integer n, we can therefore compute $C_2(n)$. For example, the integer $n = 24$ has 8 divisors $c = 1$, $c = 2$, $c = 3$, $c = 4$, $c = 6$, $c = 8$, $c = 12$, and $c = 24$, so we have

$$C_2(24) = \min(1 + C_1(24), 2 + C_1(12), 3 + C_1(8), 4 + C_1(6), 6 + C_1(4),$$

$$8 + C_1(3), 12 + C_1(2), 24 + C_1(1)) =$$

$$\min(1+9, 2+6, 3+5, 4+4, 6+3, 8+3, 12+2, 24+1) =$$
$$\min(10, 8, 8, 8, 9, 11, 14, 25) = 8,$$

exactly as we observed earlier.

Here, the optimal value is attained for $c = 2$, so the optimal parallelepiped has $c = 2$ and $n' = n/c = 12$. For $n' = 12$, the optimal rectangle is 3×4, so for $n = 24$, the optimal parallelepiped is $2 \times 3 \times 4$.

The values $C_2(n)$ corresponding to the first 30 integers are as follows:

n	1	2	3	4	5	6	7	8	9	10
$C_2(n)$	2	3	4	4	6	5	8	5	6	7
n	11	12	13	14	15	16	17	18	19	20
$C_2(n)$	12	6	14	9	8	7	18	7	20	8
n	21	22	23	24	25	26	27	28	29	30
$C_2(n)$	10	13	24	8	10	15	8	10	30	9

The corresponding optimal parallelepipeds are:

1	2	3	4	5
$1 \times 1 \times 1$	$1 \times 1 \times 2$	$1 \times 1 \times 3$	$1 \times 2 \times 2$	$1 \times 1 \times 5$
6	7	8	9	10
$1 \times 2 \times 3$	$1 \times 1 \times 7$	$2 \times 2 \times 2$	$1 \times 3 \times 3$	$1 \times 2 \times 5$
11	12	13	14	15
$1 \times 1 \times 11$	$2 \times 2 \times 3$	$1 \times 1 \times 13$	$1 \times 2 \times 7$	$1 \times 3 \times 5$
16	17	18	19	20
$2 \times 2 \times 4$	$1 \times 1 \times 17$	$2 \times 3 \times 3$	$1 \times 1 \times 19$	$2 \times 2 \times 5$
21	22	23	24	25
$1 \times 3 \times 7$	$1 \times 2 \times 11$	$1 \times 1 \times 23$	$2 \times 3 \times 4$	$1 \times 5 \times 5$
26	27	28	29	30
$1 \times 2 \times 13$	$3 \times 3 \times 3$	$2 \times 2 \times 7$	$1 \times 1 \times 29$	$2 \times 3 \times 5$

Case when we allow simple counting, a single rectangle, or a single parallelepiped. In this case, we can select the option with the smallest number of computational steps, i.e., we get

$$C'(n) = \min(C_0(n), C_1(n), C_2(n)). \tag{3}$$

The values $C'(n)$ corresponding to the first 30 integers are as follows:

n	1	2	3	4	5	6	7	8	9	10
$C'(n)$	0	1	2	3	4	4	6	5	5	6
n	11	12	13	14	15	16	17	18	19	20
$C'(n)$	10	6	12	8	7	7	16	7	18	8
n	21	22	23	24	25	26	27	28	29	30
$C'(n)$	9	12	22	8	9	14	8	10	28	9

The corresponding optimal arrangements are (where a single number means a simple counting):

1	2	3	4	5
1	2	3	2×2	5
6	7	8	9	10
2×3	7	$2\times 2\times 2$	3×3	$1\times 2\times 5$
11	12	13	14	15
11	$2\times 2\times 3$	13	2×7	3×5
16	17	18	19	20
$2\times 2\times 4$	17	$2\times 3\times 3$	19	$2\times 2\times 5$
21	22	23	24	25
3×7	2×11	23	$2\times 3\times 4$	5×5
26	27	28	29	30
2×13	$3\times 3\times 3$	$2\times 2\times 7$	29	$2\times 3\times 5$

How to solve the original problem. Our objective is to find the number $C(n)$ of computational steps that correspond to the optimal arrangement of n elements. For every positive integer n, the optimal arrangement corresponds:

- either to simple counting or a single rectangle or parallelepiped, in which case we need $C'(n)$ computational steps,
- or to several rectangles or parallelepipeds, in which case we need $C'(n')$ for the first of them, an optimal arrangement for the remaining $n-n'$ elements – leading to $C(n-n')$ steps – plus 1 additional step to add up these numbers.

The optimal arrangement corresponds to the alternative with the smallest number of computational steps:

$$C(n)=\min\left(C'(n),\min_{n':0<n'<n}(C'(n')+C(n-n')+1)\right). \qquad (4)$$

We can use this formula to sequentially compute the values $C(1)$, $C(2)$, ...

The values $C(n)$ corresponding to the first 30 integers are as follows:

n	1	2	3	4	5	6	7	8	9	10
$C(n)$	0	1	2	3	4	4	5	5	5	6
n	11	12	13	14	15	16	17	18	19	20
$C(n)$	7	6	7	8	7	7	8	7	8	8
n	21	22	23	24	25	26	27	28	29	30
$C(n)$	9	10	11	8	9	10	8	9	10	9

The corresponding optimal arrangements are:

1	2	3	4	5
1	2	3	2×2	5
6	7	8	9	10
2×3	$1 + 2 \times 3$	$2 \times 2 \times 2$	3×3	$1 \times 2 \times 5$
11	12	13	14	15
$1 + 2 \times 5$	$2 \times 2 \times 3$	$1 + 2 \times 2 \times 3$	2×7	3×5
16	17	18	19	20
4×4	$1 + 4 \times 4$	$2 \times 3 \times 3$	$1 + 2 \times 3 \times 3$	$2 \times 2 \times 5$
21	22	23	24	25
3×7	$1 + 3 \times 7$	$2 + 3 \times 7$	$2 \times 3 \times 4$	5×5
26	27	28	29	30
$1 + 5 \times 5$	$3 \times 3 \times 3$	$1 + 3 \times 3 \times 3$	$2 + 3 \times 3 \times 3$	$2 \times 3 \times 5$

Comment. For some integers n, there are several optimal arrangements; in this case, we list one of them. For example, for $n = 21$, in addition to the arrangement 3×7 that we listed in the above table, the same number of computational steps can also be achieved if we use a different arrangement $1 + 4 \times 5$.

Computational complexity of computing $C(n)$. For each n, to compute $C_1(n)$, we need to check all numbers $a \le \sqrt{n}$ – whether they divide n. This requires $\sqrt{n}$ computational steps. Thus, to compute the values $C_1(n)$ for all n from 1 to N, we need $\sum\limits_{n=1}^{N} \sqrt{n} \sim N^{3/2}$ computational steps.

To compute $C_2(n)$ for each n, we also need to check all the numbers $c \le n$ – whether they divide n. Thus, to compute the values $C_2(n)$ for all n from 1 to N, we need $1 + 2 + \ldots + N \sim N^2$ computational steps.

Computing $C'(n)$ for all $n \le N$ requires $O(N)$ steps.

Finally, computing each value $C(n)$ according to the formula (4) requires n steps, to the total of $1 + 2 + \ldots + N = O(N^2)$ steps.

All these procedures require $O(N^{3/2}) + O(N^2) + O(N) + ON^{@}) = O(N^2)$ steps, so we can compute $C(N)$ in time quadratic in N.

Comment. Some objects has such a shape that we cannot place one on top of the others, so we cannot make a parallelepiped out of them. In this case, we have to limit ourselves to rectangles. For such objects, we can repeat the above computations, with the only difference that there is no need to compute $C_2(n)$ and $C'(n)$ is now computed according to a new formula $C'(n) = \min(C_0(n), C_1(n))$.

What if we do not know how many elements we have. In the above text, we assumed that the customer knows how many objects she picked, and the question was how to arrange these objects so as to check that the number of objects is correct.

But what if we do not know how many objects n we have picked? If we placed them into a parallelepiped of sizes a, b, and c, then among all the values a, b, and c for which $a \cdot b \cdot c = n$, we must select the triple for which $a + b + c - 1$ is the smallest possible. To get an approximate solution to this optimization problem, let us consider a simplified version of this problem, in which the values a, bb, and c can be arbitrary real numbers (not necessarily integers). This simplified problem can be easily solved, and its solution is $a = b = c = \sqrt[3]{n}$, i.e., a cube.

Thus, it is reasonable to stack the objects in such a way that they form a cube. First, we form a cube of linear size 2, then we add elements on all sides to get a cube of linear size 3, etc.

References

[1] M. Burns, *Math by All Means: Multiplication, Grade 3*, Math Solutions Publications, Sausalito, California, 1995.

[2] L. J. Morrow (Ed.), *The Teaching and Learning of Algorithms in School Mathematics*, National Council of Teachers of Mathematics, Reston, Virginia, 1998.

[3] J. Pallotta and R. Bolster, *Hershey's Milk Chocolate Multiplication Book*, Cartwheel Publ., 2002.

[4] Ji-Won Son, "A comparison of how textbooks teach multiplication of fractions and division of fractions in Korea and in the U.S.", In: H. L. Chick and J. L. Vincent (Eds.), *Proceedings of the 29th Conference of the International Group for the Psychology of Mathematics Education PME*, PME Publ., Melbourne, 2005, Vol. 4, pp. 201–208.

[5] J. A. Van de Walle, *Elementary and Middle School Mathematics: Teaching Developmentally*, Allyn & Bacon, Boston, Massachusetts, 2006.