Lemmas on the floor and ceiling functions

Laurent Lyaudet*

December 7, 2025

Abstract

A few simple lemmas on the floor and ceiling functions.

Initial version: 2025/12/03 Current version: 2025/12/07

Keywords: integer part, lower integer part, higher integer part, floor function, ceiling function, higher modulo, higher modulus, complemented modulo, complemented modulus, lack, fractionnal part, fractionnal lack.

This note contains simples lemmas on the floor and ceiling functions, and on switching from one function to the other. These lemmas are not in the book Graham et al. (1989), but they could very well be added to it, maybe as slightly easy exercises. We have done a bibliographic search without success, but it is possible that they are already in literature. In which case, we would be happy to recieve an email with the correct reference.

1 Définitions et notations

Definition 1.1. *Let* $r, s \in \mathbb{R}$. *We denote* $\mathrm{exch}_{r\leftrightarrow s}$*, the* exchange function *defined by* $\mathrm{exch}_{r\leftrightarrow s}(r) = s$*,* $\mathrm{exch}_{r\leftrightarrow s}(s) = r$*,* $\mathrm{exch}_{r\leftrightarrow s}(x) = x$ *otherwise.*

For two integers n and d, we denote $n \div d$ the integer quotient of n divided by d.

Definition 1.2. *Let* $n, d \in \mathbb{N}$.
We denote mod^+ *the* higher modulo *that computes the* higher modulus *of* n *and* d *defined by* $n \bmod^+ d = \mathrm{exch}_{0\leftrightarrow d}(n \bmod d)$.
We have $n \bmod^+ d \in \mathbb{N}$ *and* $1 \leq n \bmod^+ d \leq d$.

Definition 1.3. *Let* $n, d \in \mathbb{N}$.
We denote mod^c *the* complemented modulo *that computes the* complemented modulus *of* n *and* d *defined by* $n \bmod^c d = d - (n \bmod d)$.
We have $n \bmod^c d \in \mathbb{N}$ *and* $1 \leq n \bmod^c d \leq d$.

Definition 1.4. *Let* $n, d \in \mathbb{N}$.
We denote by $n \,\mathrm{lack}\, d = d - (n \bmod^+ d)$ *the* lack *of* n *by* d.
Informally, $n \,\mathrm{lack}\, d = n \bmod^{+c} d = n \bmod^{c+} d$.
We have $n \,\mathrm{lack}\, d \in \mathbb{N}$ *and* $0 \leq n \,\mathrm{lack}\, d \leq d - 1$*, as the classical modulus.*

*`https://lyaudet.eu/laurent/`, laurent.lyaudet@gmail.com

The fractionnal part of a real number x is denoted by $\{x\} = x - \lfloor x \rfloor$.

Definition 1.5. *Let $x \in \mathbb{R}$.*
We denote by $\{x\}^{+c}$ or $\{x\}^{c+}$ the fractionnal lack *of x, that is defined by $\{x\}^{+c} = \lceil x \rceil - x = \mathrm{exch}_{0\leftrightarrow 1}(1 - \{x\})$.*
We have $0 \leq \{x\}^{+c} < 1$, as the classical fractionnal part.

It is possible that the *exchange function*, the *higher modulo/modulus*, the *complemented modulo/modulus*, the *lack*, the *fractionnal lack* already exist in literature. Here again, we would be happy to recieve an email with the correct reference.

2 Results

Let's start with some observations that are already in the folklore: if $k \in \mathbb{N}$, $\lceil \frac{k}{2} \rceil = \lfloor \frac{k+1}{2} \rfloor$ (Graham et al. (1989) for example), or $\lceil k \times \frac{2}{3} \rceil = \lfloor (k+1) \times \frac{2}{3} \rfloor$ (OEIS Foundation Inc. (2025)). Apparently, nobody took the time to generalise these folklore observations, but we see that:

Lemma 2.1. *Let $k, a \in \mathbb{N}, a \geq 1$, then $\lceil k \times \frac{a-1}{a} \rceil = \lfloor (k+1) \times \frac{a-1}{a} \rfloor$.*
Note that $\frac{a-1}{a} = 1 - \frac{1}{a}$.

Proof:

Idea: The gap $(k+1) \times \frac{a-1}{a} - k \times \frac{a-1}{a} = \frac{a-1}{a} < 1$. Hence if these two values are on both sides of an integer b, then $b = \lceil k \times \frac{a-1}{a} \rceil = \lfloor (k+1) \times \frac{a-1}{a} \rfloor$.

Formal proof: $\lceil k \times \frac{a-1}{a} \rceil = \lceil k \times (1 - \frac{1}{a}) \rceil = \lceil k - \frac{k}{a} \rceil = k + \lceil -\frac{k}{a} \rceil = k - \lfloor \frac{k}{a} \rfloor$ and $\lfloor (k+1) \times \frac{a-1}{a} \rfloor = \lfloor (k+1) \times (1 - \frac{1}{a}) \rfloor = \lfloor (k+1) - \frac{k+1}{a} \rfloor = k + 1 + \lfloor -\frac{k+1}{a} \rfloor = k + 1 - \lceil \frac{k+1}{a} \rceil$. Let $q = k \div a$. We have $k - \lfloor \frac{k}{a} \rfloor = k - q$ and $k + 1 - \lceil \frac{k+1}{a} \rceil = k + 1 - (q+1) = k - q$.

$\blacksquare$

Note that it cannot be generalised further. Some examples of blocking: $\lceil 1 \times \frac{1}{3} \rceil > \lfloor 2 \times \frac{1}{3} \rfloor$, $\lceil 2 \times \frac{3}{5} \rceil > \lfloor 3 \times \frac{3}{5} \rfloor$, $\lceil 0 \times \frac{3}{5} \rceil < \lfloor 2 \times \frac{3}{5} \rfloor$, $\lceil 3 \times \frac{3}{5} \rceil < \lfloor 5 \times \frac{3}{5} \rfloor$, $\lceil 3 \times \frac{5}{7} \rceil > \lfloor 4 \times \frac{5}{7} \rfloor$, $\lceil 0 \times \frac{5}{7} \rceil < \lfloor 2 \times \frac{5}{7} \rfloor$, $\lceil 1 \times \frac{5}{7} \rceil < \lfloor 3 \times \frac{5}{7} \rfloor$. A last example that shows that k must be an integer: $\lceil \frac{a}{3 \times (a-1)} \times \frac{a-1}{a} \rceil = 1 \neq \lfloor (\frac{a}{3 \times (a-1)} + 1) \times \frac{a-1}{a} \rfloor = \lfloor \frac{1}{3} + \frac{a-1}{a} \rfloor = 0$.

Note that we have also the known lemma (see Graham et al. (1989)):

Lemma 2.2. *Let $k, a \in \mathbb{N}, a \geq 1$, then $\lceil k \times \frac{1}{a} \rceil = \lfloor (k + a - 1) \times \frac{1}{a} \rfloor$.*

Lemma 2.3. *Let $k, l, a \in \mathbb{N}, a \geq 2$. If $m = k + \lceil \frac{k}{a-1} \rceil$ and $M = k + \lceil \frac{k+1}{a-1} \rceil$, then $\lfloor l \times \frac{a-1}{a} \rfloor = k \Leftrightarrow m \leq l \leq M$.*

Proof:

Let $m = k + \lceil \frac{k}{a-1} \rceil$.
We have $\lfloor m \times \frac{a-1}{a} \rfloor = \lfloor (k + \lceil \frac{k}{a-1} \rceil) \times \frac{a-1}{a} \rfloor = \lfloor (k + \frac{k}{a-1} + \frac{k \operatorname{lack}(a-1)}{a-1}) \times \frac{a-1}{a} \rfloor = \lfloor k + \frac{k \operatorname{lack}(a-1)}{a} \rfloor = k$
and $\lfloor (m-1) \times \frac{a-1}{a} \rfloor = \lfloor (k + \lceil \frac{k}{a-1} \rceil - 1) \times \frac{a-1}{a} \rfloor = \lfloor (k + \frac{k}{a-1} + \frac{k \operatorname{lack}(a-1)}{a-1} - 1) \times \frac{a-1}{a} \rfloor = \lfloor k + \frac{k \operatorname{lack}(a-1)}{a} - \frac{a-1}{a} \rfloor = k - 1$.
Let $M = k + \lceil \frac{k+1}{a-1} \rceil$.
We have $\lfloor M \times \frac{a-1}{a} \rfloor = \lfloor (k + \lceil \frac{k+1}{a-1} \rceil) \times \frac{a-1}{a} \rfloor = \lfloor (k + \frac{k+1}{a-1} + \frac{(k+1) \operatorname{lack}(a-1)}{a-1}) \times \frac{a-1}{a} \rfloor = \lfloor k + \frac{1}{a} + \frac{(k+1) \operatorname{lack}(a-1)}{a} \rfloor = k$
and $\lfloor (M+1) \times \frac{a-1}{a} \rfloor = \lfloor (k + \lceil \frac{k+1}{a-1} \rceil + 1) \times \frac{a-1}{a} \rfloor = \lfloor (k + \frac{k+1}{a-1} + \frac{(k+1) \operatorname{lack}(a-1)}{a-1} + 1) \times \frac{a-1}{a} \rfloor = \lfloor k + \frac{1}{a} + \frac{(k+1) \operatorname{lack}(a-1)}{a} + \frac{a-1}{a} \rfloor = k + 1$. ∎

By Lemma 2.2, we also have $m = k + \lceil \frac{k}{a-1} \rceil = k + \lfloor \frac{k+a-2}{a-1} \rfloor$, and $M = k + \lceil \frac{k+1}{a-1} \rceil = k + \lfloor \frac{k+1+a-2}{a-1} \rfloor = k + \lfloor \frac{k}{a-1} \rfloor + 1$.

Lemma 2.4. *Let $k, l, a \in \mathbb{N}, a \geq 2$. If $m = k + \lfloor \frac{k-1}{a-1} \rfloor$ and $M = k + \lfloor \frac{k}{a-1} \rfloor$, then $\lceil l \times \frac{a-1}{a} \rceil = k \Leftrightarrow m \leq l \leq M$.*

Proof:

Let $m = k + \lfloor \frac{k-1}{a-1} \rfloor$.
We have $\lceil m \times \frac{a-1}{a} \rceil = \lceil (k + \lfloor \frac{k-1}{a-1} \rfloor) \times \frac{a-1}{a} \rceil = \lceil (k + \frac{k-1}{a-1} - \frac{(k-1) \bmod (a-1)}{a-1}) \times \frac{a-1}{a} \rceil = \lceil k - \frac{1}{a} - \frac{(k-1) \bmod (a-1)}{a} \rceil = k$
and $\lceil (m-1) \times \frac{a-1}{a} \rceil = \lceil (k + \lfloor \frac{k-1}{a-1} \rfloor - 1) \times \frac{a-1}{a} \rceil = \lceil (k + \frac{k-1}{a-1} - \frac{(k-1) \bmod (a-1)}{a-1} - 1) \times \frac{a-1}{a} \rceil = \lceil k - \frac{1}{a} - \frac{(k-1) \bmod (a-1)}{a} - \frac{a-1}{a} \rceil = k - 1$.
Let $M = k + \lfloor \frac{k}{a-1} \rfloor$.
We have $\lceil M \times \frac{a-1}{a} \rceil = \lceil (k + \lfloor \frac{k}{a-1} \rfloor) \times \frac{a-1}{a} \rceil = \lceil (k + \frac{k}{a-1} - \frac{k \bmod (a-1)}{a-1}) \times \frac{a-1}{a} \rceil = \lceil k - \frac{k \bmod (a-1)}{a} \rceil = k$
and $\lceil (M+1) \times \frac{a-1}{a} \rceil = \lceil (k + \lfloor \frac{k}{a-1} \rfloor + 1) \times \frac{a-1}{a} \rceil = \lceil (k + \frac{k}{a-1} - \frac{k \bmod (a-1)}{a-1} + 1) \times \frac{a-1}{a} \rceil = \lceil k - \frac{k \bmod (a-1)}{a} + \frac{a-1}{a} \rceil = k + 1$. ∎

By Lemma 2.2, we also have $m = k + \lfloor \frac{k-1}{a-1} \rfloor = k + \lceil \frac{k-1-(a-2)}{a-1} \rceil = k + \lceil \frac{k}{a-1} \rceil - 1$, and $M = k + \lfloor \frac{k}{a-1} \rfloor = k + \lceil \frac{k-(a-2)}{a-1} \rceil$.

These integer parts are quite frequent in combinatorics, for exemple $k \to \lceil k \times \frac{a-1}{a} \rceil$ corresponds to a recursive cutting of a set of size n where the largest part keeps some fraction of the elements of the set rounded to the integer above. This cutting must stop when $n = a - 1$. We can then count the number of steps to reach $a - 1$, *i.e.* the depth of the tree of the cuts. If we want to generate the sequence of the number of steps by increasing order of the values of n, the bound M above is very useful. Here is an example of pseudo-code of a very efficient generator:

```
// Input a or a-1
b = a-1;
n = b;
M = b;
k = 0;
while(true){
  while(n <= M){
    yield k;
    ++n;
  }
  M += M/b;
  ++k;
}
/*
Example with a = 3
n 2 3 4 5 6 7 8 9
M 2 3 4 6 6 9 9 9
k 0 1 2 3 3 4 4 4
*/
```

The same reasoning applies to a recursive cutting of a set of size n where the largest part keeps some fraction of the elements of the set rounded to the integer below. This cutting must stop when $n = 0$. The pseudo-code of the generator is slightly less efficient:

```
// Input a ou a-1
b = a-1;
n = 0;
M = 0;
k = 0;
while(true){
  while(n <= M){
    yield k;
    ++n;
  }
  M += (M+b)/b;
  ++k;
}
/*
Example with a = 3
n 0 1 2 3 4 5 6 7
M 0 1 2 4 4 7 7 7
k 0 1 2 3 3 4 4 4
*/
```

The proofs of the two last lemmas can illustrate some definitions. We can generalise them a little. Let $r \in \mathbb{R}, 0 < r < 1$. Let $a, b \in \mathbb{N}, b < a$. If $r = \frac{b}{a}$, $\frac{1-r}{r} = \frac{(1-\frac{b}{a})\times a}{b} =$

$\frac{\frac{a-b}{a}\times a}{b} = \frac{a-b}{b}$. If $r = \frac{a-1}{a}$, $\frac{1-r}{r} = \frac{1}{a-1}$.

If we consider recursive cuttings given by a function $l \to \lfloor l \times r \rfloor$, we can talk about lower recursive cutting of ratio r (ratio as for geometric series, it also works in French but we talk about the *raison* of a *série géométrique*). A lower recursive cutting always ends at 0. If we consider recursive cuttings given by a function $l \to \lceil l \times r \rceil$, we can talk about higher recursive cutting of ratio r. A higher recursive cutting loops for $n \times r > n - 1 \Leftrightarrow r > \frac{n-1}{n}$.

Lemma 2.5. *Let* $k, l \in \mathbb{N}, r \in \mathbb{R}, 0 < r < 1$.
If $m = k + \lceil \frac{k\times(1-r)}{r} \rceil$, $M = k + \lfloor \frac{k\times(1-r)}{r} \rfloor + \lceil \frac{1}{r} \rceil - 1$, $M' = k + \lceil \frac{k\times(1-r)}{r} \rceil + \lceil \frac{1}{r} \rceil - 1$, *then* $\lfloor l \times r \rfloor = k \Rightarrow m \le l \le M'$ *and* $m \le l \le M \Rightarrow \lfloor l \times r \rfloor = k$.

Proof:

$l \times r = k \Leftrightarrow l = k \times \frac{1}{r} \Leftrightarrow l = k + k \times (\frac{1}{r} - 1) \Leftrightarrow l = k + k \times \frac{1-r}{r}$, hence $l \times r \ge k \Leftrightarrow l \ge k \times \frac{1}{r} \Leftrightarrow l \ge k + k \times (\frac{1}{r} - 1) \Leftrightarrow l \ge k + k \times \frac{1-r}{r}$, because we do not multiply by a negative number. Let $m = k + \lceil \frac{k\times(1-r)}{r} \rceil$.
We have $\lfloor m \times r \rfloor \ge \lfloor (k + k \times \frac{1-r}{r}) \times r \rfloor \ge \lfloor k \rfloor = k$
and $m = k + \lceil \frac{k\times(1-r)}{r} \rceil \Rightarrow m - 1 < k + \frac{k\times(1-r)}{r} \Rightarrow (m-1) \times r < k \Rightarrow \lfloor (m-1) \times r \rfloor < k$.

Let $M = k + \lfloor \frac{k\times(1-r)}{r} \rfloor + \lceil \frac{1}{r} \rceil - 1$.

We have $\lfloor M \times r \rfloor \le \lfloor (k + k \times \frac{1-r}{r} + \lceil \frac{1}{r} \rceil - 1) \times r \rfloor = \lfloor (k + k \times \frac{1-r}{r}) \times r + (\lceil \frac{1}{r} \rceil - 1) \times r \rfloor = \lfloor k + (\lceil \frac{1}{r} \rceil - 1) \times r \rfloor = k + \lfloor (\lceil \frac{1}{r} \rceil - 1) \times r \rfloor = k + \lfloor (\frac{1}{r} + \{\frac{1}{r}\}^{+c} - 1) \times r \rfloor = k + \lfloor 1 + (\{\frac{1}{r}\}^{+c} - 1) \times r \rfloor = k$ because $-1 \le \{\frac{1}{r}\}^{+c} - 1 < 0 \Rightarrow -r \le (\{\frac{1}{r}\}^{+c} - 1) \times r < 0 \Rightarrow \lfloor 1 + (\{\frac{1}{r}\}^{+c} - 1) \times r \rfloor = 0$ since $0 < r < 1$.

Reciprocally $M' = k + \lceil \frac{k\times(1-r)}{r} \rceil + \lceil \frac{1}{r} \rceil - 1 \Rightarrow M' + 1 = k + \frac{k\times(1-r)}{r} + \{\frac{k\times(1-r)}{r}\}^{+c} + \frac{1}{r} + \{\frac{1}{r}\}^{+c} \Rightarrow (M'+1) \times r = k + 1 + (\{\frac{k\times(1-r)}{r}\}^{+c} + \{\frac{1}{r}\}^{+c}) \times r \Rightarrow \lfloor (M' + 1) \times r \rfloor > k$.

■

Note that if $r = \frac{1}{a}$, $\frac{1-r}{r} = a - 1 \in \mathbb{N}$ implies that $M = M'$. And note that if $r = \frac{a-1}{a}$, we have $\frac{1-r}{r} = \frac{1}{a-1}$, $\lceil \frac{1}{r} \rceil = \lceil \frac{a}{a-1} \rceil = 2$ implies that $M = k + \lfloor \frac{k\times(1-r)}{r} \rfloor + \lceil \frac{1}{r} \rceil - 1 = k + \lfloor \frac{k}{a-1} \rfloor + 1 = k + \lfloor \frac{k+a-1}{a-1} \rfloor = k + \lceil \frac{k+1}{a-1} \rceil$, the bound that we already know, and hence M' is useless in that case.

None of the two bounds M or M' can replace the other. Examples with $r = \frac{3}{5}$, $\frac{1-r}{r} = \frac{2}{3}$, $M = k + \lfloor \frac{k\times(1-r)}{r} \rfloor + \lceil \frac{1}{r} \rceil - 1 = k + \lfloor \frac{k\times 2}{3} \rfloor + \lceil \frac{5}{3} \rceil - 1 = k + \lfloor \frac{k\times 2}{3} \rfloor + 1$,

$M' = k + \lceil \frac{k\times 2}{3} \rceil + 1$. If $k = 0$, $M = 0 + \lfloor 0 \rfloor + 1 = 1$ and $M' = 0 + \lceil 0 \rceil + 1 = 1$, $M = M'$. If $k = 1$, $M = 1 + \lfloor \frac{2}{3} \rfloor + 1 = 2$ and $M' = 1 + \lceil \frac{2}{3} \rceil + 1 = 3$, $\lfloor M' \times \frac{3}{5} \rfloor = \lfloor 3 \times \frac{3}{5} \rfloor = 1 = k$. If $k = 2$, $M = 2 + \lfloor \frac{4}{3} \rfloor + 1 = 4$ and $M' = 2 + \lceil \frac{4}{3} \rceil + 1 = 5$, $\lfloor M' \times \frac{3}{5} \rfloor = \lfloor 5 \times \frac{3}{5} \rfloor = 3 > k$.

Lemma 2.6. *Let* $k, l \in \mathbb{N}, r \in \mathbb{R}, 0 < r < 1$.
If $m = k + \lceil \frac{k\times(1-r)}{r} \rceil - \lceil \frac{1}{r} \rceil + 1$, $m' = k + \lfloor \frac{k\times(1-r)}{r} \rfloor - \lceil \frac{1}{r} \rceil + 1$, $M = k + \lfloor \frac{k\times(1-r)}{r} \rfloor$, *then* $\lfloor l \times r \rfloor = k \Rightarrow m' \leq l \leq M$ *and* $m \leq l \leq M \Rightarrow \lfloor l \times r \rfloor = k$.

Proof:

$l \times r = k \Leftrightarrow l = k \times \frac{1}{r} \Leftrightarrow l = k + k \times (\frac{1}{r} - 1) \Leftrightarrow l = k + k \times \frac{1-r}{r}$, hence $l \times r \leq k \Leftrightarrow l \leq k \times \frac{1}{r} \Leftrightarrow l \leq k + k \times (\frac{1}{r} - 1) \Leftrightarrow l \leq k + k \times \frac{1-r}{r}$, because we do not multiply by a negative number. Let $M = k + \lfloor \frac{k\times(1-r)}{r} \rfloor$.
We have $\lceil M \times r \rceil \leq \lceil (k + k \times \frac{1-r}{r}) \times r \rceil \leq \lceil k \rceil = k$
and $M = k + \lfloor \frac{k\times(1-r)}{r} \rfloor \Rightarrow M + 1 > k + \frac{k\times(1-r)}{r} \Rightarrow M + 1 \times r > k \Rightarrow \lceil (M + 1) \times r \rceil > k$.
Let $m = k + \lceil \frac{k\times(1-r)}{r} \rceil - \lceil \frac{1}{r} \rceil + 1$.
We have $\lceil m \times r \rceil \geq \lceil (k + k \times \frac{1-r}{r} - \lceil \frac{1}{r} \rceil + 1) \times r \rceil = \lceil (k + k \times \frac{1-r}{r}) \times r + (-\lceil \frac{1}{r} \rceil + 1) \times r \rceil = \lceil k + (-\lceil \frac{1}{r} \rceil + 1) \times r \rceil = k + \lceil (-\lceil \frac{1}{r} \rceil + 1) \times r \rceil = k + \lceil (-\frac{1}{r} - \{\frac{1}{r}\}^{+c} + 1) \times r \rceil = k + \lceil -1 + (1 - \{\frac{1}{r}\}^{+c}) \times r \rceil = k$ because $0 < 1 - \{\frac{1}{r}\}^{+c} \leq 1 \Rightarrow 0 < (1 - \{\frac{1}{r}\}^{+c}) \times r \leq r \Rightarrow \lceil -1 + (1 - \{\frac{1}{r}\}^{+c}) \times r \rceil = 0$ since $0 < r < 1$.
Reciprocally $m' = k + \lfloor \frac{k\times(1-r)}{r} \rfloor - \lceil \frac{1}{r} \rceil + 1 \Rightarrow m' - 1 = k + \frac{k\times(1-r)}{r} - \{\frac{k\times(1-r)}{r}\} - \frac{1}{r} - \{\frac{1}{r}\}^{+c} \Rightarrow (m'-1) \times r = k - 1 - (\{\frac{k\times(1-r)}{r}\} + \{\frac{1}{r}\}^{+c}) \times r \Rightarrow \lceil (m'-1) \times r \rceil < k$. ∎

Again, note that if $r = \frac{1}{a}$, $\frac{1-r}{r} = a - 1 \in \mathbb{N}$ implies that $M = M'$. And note that if $r = \frac{a-1}{a}$, we have $\frac{1-r}{r} = \frac{1}{a-1}$, $\lceil \frac{1}{r} \rceil = \lceil \frac{a}{a-1} \rceil = 2$ implies that $m = k + \lceil \frac{k\times(1-r)}{r} \rceil - \lceil \frac{1}{r} \rceil + 1 = k + \lceil \frac{k}{a-1} \rceil - 1 = k + \lceil \frac{k-a+1}{a-1} \rceil = k + \lfloor \frac{k-1}{a-1} \rfloor$, the bound that we already know, and hence m' is useless in that case.

None of the two bounds m or m' can replace the other. Examples with $r = \frac{3}{5}$, $\frac{1-r}{r} = \frac{2}{3}$, $m = k + \lceil \frac{k\times(1-r)}{r} \rceil - \lceil \frac{1}{r} \rceil + 1 = k + \lceil \frac{k\times 2}{3} \rceil - \lceil \frac{5}{3} \rceil + 1 = k + \lceil \frac{k\times 2}{3} \rceil - 1$, $m' = k + \lfloor \frac{k\times 2}{3} \rfloor - 1$. If $k = 2$ ($\frac{2}{3} > \frac{3}{5} > \frac{1}{2}$), $m = 2 + \lceil \frac{6}{5} \rceil - 1 = 3$ and

$m' = 2+\lfloor\frac{6}{5}\rfloor-1 = 2$, $\lceil m'\times\frac{3}{5}\rceil = \lceil 2\times\frac{3}{5}\rceil = 2 = k$. If $k = 3$, $m = 3+\lceil\frac{9}{5}\rceil-1 = 4$ and $m' = 3+\lfloor\frac{9}{5}\rfloor-1 = 3$, $\lceil m'\times\frac{3}{5}\rceil = \lceil 3\times\frac{3}{5}\rceil = 2 < k$. If $k = 4$, $m = 4+\lceil\frac{12}{5}\rceil-1 = 6$ and $m' = 4+\lfloor\frac{12}{5}\rfloor-1 = 5$, $\lceil m'\times\frac{3}{5}\rceil = \lceil 5\times\frac{3}{5}\rceil = 3 < k$. If $k = 5$, $m = 5+\lceil\frac{15}{5}\rceil-1 = 7$ and $m' = 5+\lfloor\frac{15}{5}\rfloor-1 = 7$, $m = m'$.

3 Historical notes, moderation abuse at the OEIS

On a close subject, I have been doing additions to the OEIS for slightly less than 10 days (since 26 November 2025), and in particular to the sequence A061420. I also proposed two new sequences linked with A061420 and with my article Lyaudet (2025b), that recieved the numbers A391044 and A391108. At the beginning, I thought they would be happy of my additions, but I quickly realised that they play the rotting of my proposals with self-contradicting sequences of comments, or maneuvers of ordinary sadists that seek to “castrate others”. It was one of the things that my father retained of his psychanalysis and that he taught me: identify those that seek to stop you in your momentum, steal your wings, or more vulgarly cut “yours”. I kindly did the modifications, found solutions, the situation seemed to improve because everything was dealed with on the discussion of A391044. Hence, I was just waiting that a “higher rank” editor validates and publishes the sequence A391044, but two days after the end of the modifications a “higher rank” editor “recycled” A391044, according to him because of `https://oeis.org/wiki/%22War_and_Peace%22_submissions`. But the webpage “War and peace” talks about a one month minimum delay and the censorship occured only 8 days after my addition of this sequence to the OEIS, everything was settled positively on A391044 and the exchanges were not that numerous, a “slight” abuse of power... Hence, when they started again the same circus on A391108, whilst there was only one solved comment and its response so far in the discussion, since I learned their kind of writing and needs from the corrections on A391044, I “kindly” asked them if they are spies or fence of spies that listen to the “mafia” (state or not) that harass me since 20 years. (My saves of my website in the Wayback Machine between September and the 3 of December disappeared the 5 of December, for example. Only the saves from before September are still there. I sent an email to the Internet Archive to know if it is just a bug and the saves are still somewhere, but I had no answer.) Maybe, they are only small sadists that abuse from their power, maybe they are thieves that disgust other people before stealing their ideas and publish them, or it is one more focused persecution. But clearly, they do not put Free Science first. On all the “open” websites of the web, we find this kind of phenomenon of “bullying” with some people that play the rotting or for time on what others propose. As I said to them: “If you serve science, you can correct and improve my addition proposals, but you cannot censor them.”. In order that the reader can judge with evidences, here are my 3 addition proposals (the only current proposal is A391108, but they started again the play of rotting and bad faith, hence it is a matter of hours or days before it is rejected).

For this first evidence, I only give my additions:

```
A061420
```

```
COMMENTS From Laurent Lyaudet, Nov 27 2025: (Start)
         a(n-2) is the number of steps to go from n to 2 under
           the map g: x -> ceiling(x*2/3).
         a(n-2) = L(n) or L(n)+1 where L(n) =
           ceiling(log(n/2)/log(3/2)),
           and those n-2 in A391108 are the former case,
           those n-2 in A391044 are the latter case.
         (For an integer k, ceiling(k/2) = floor((k+1)/2).)
         For an integer k, m = k + floor((k-1)/2) is the smallest
           integer such that g(m) = k.
         For an integer k, M = k + floor(k/2) is the largest integer
           such that g(M) = k.
         If h: x -> ceiling((x-1)*2/3) is the map in the definition of
           A061420 or f: x -> floor(x*2/3) in Benoit Cloitre’s comment
           (note that f != h, but f(k) = h(k) for any integer k)
           and k is an integer:
           m = k + floor((k+1)/2) is the smallest integer such that
             h(m) = k and/or f(m) = k.
           M = k + floor(k/2) + 1 is the largest integer such that
             h(M) = k and/or f(M) = k
             (see PHP program for an application).(End)
PROG (PHP)
<?php
// See 3rd paragraph in COMMENTS, in particular:
// "M = k + floor(k/2) + 1 is the largest integer such that h(M) = k".
// This program is very fast and uses constant memory
// (4 64 bits signed integers), apart from output,
// since it doesn’t require to keep previous terms,
// and is limited to integers less than 2^64.
// Inputs:
$nMax = 2000;
// Outputs to std output the b-file.
echo("0 0\n1 1\n2 2\n");
$n = 3;
$b = 4;
$k = 3;
while(true){
  while($n <= $b){
    echo("$n $k\n");
    ++$n;
  }
  $b += intdiv($b, 2) + 1;
  ++$k;
  if($b > $nMax){
    $b = $nMax;
```

```
    break;
  }
}
while($n <= $b){
  echo("$n $k\n");
  ++$n;
}
// Laurent Lyaudet, Dec 01 2025
?>

CROSSREFS Cf. A391044, A391108.
```

Previously, they forbid:

```
a_2(m) = a_2(ceiling(m*2/3)) + 1 with a_2(2) = 0 and offset 2 is
simpler and equivalent through the change of variable m = n + 2
and may seem more natural in some combinatorial context:
Start with n elements, keep only ceiling(n*2/3) of those,
repeat until you reach 2, where ceiling(2*2/3) = 2 loops.
From A391044, let L(n) = ceiling(log(n/2)/log(3/2)), a(n-2) = L(n) + 1
if n is in A391044, a(n-2) = L(n) otherwise,
  Laurent Lyaudet, Nov 27 2025
```

and they asked that I simplify the program that was:

```
<?php
// See 3rd paragraph in COMMENTS, in particular:
// "M = k + floor(k/2) + 1 is the largest integer such that h(M) = k".
// This program is very fast and uses constant memory
// (4 64 bits signed integers and a boolean), apart from output,
// since it doesn't require to keep previous terms,
// and is limited to integers less than 2^64.
// Inputs:
$bFile = true;
$nMax = 2000;
// Outputs to std output:
// - the comma separated sequence if $bFile === false,
// - the b-file if $bFile === true.
if($bFile){ echo("0 0\n1 1\n2 2\n"); }else{ echo("0, 1, 2, "); }
$n = 3;
$b = 4;
$k = 3;
while(true){
  while($n <= $b){
    if($bFile){ echo("$n $k\n"); }else{ echo("$k, "); }
    ++$n;
```

```
  }
  $b += intdiv($b, 2) + 1;
  ++$k;
  if($b > $nMax){
    $b = $nMax;
    break;
  }
}
while($n <= $b){
  if($bFile){ echo("$n $k\n"); }else{ echo("$k, "); }
  ++$n;
}
// Laurent Lyaudet, Dec 01 2025
?>
```

Here again, their request to simplify the program by removing $bFile parameter had no interest. The complexity of the two variants are of the same magnitude. The existing programs on A061420 are less efficient, since they must keep in memory all the previous terms. They decided to remove everything: “hop, nothing to keep in the work of others”.

Below, my first proposal of sequence, the name was modified by the editor, and it is less clear than what I did for the second sequence proposal.

```
A391044
NAME Numbers k >= 2 for which the number of steps x -> ceiling(x*2/3)
     to reach 2, being S(k) = A061420 (n-2) many,
     is S(k) = L(k)+1 where L(k) = ceiling(log(k/2)/log(3/2)).
DATA 10, 14, 15, 20, 21, 22, 29, 30, 31, 32, 33, 34, 43, 44, 45, 46,
     47, 48, 49, 50, 51, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74,
     75, 76, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106,
     107, 108, 109, 110, 111, 112, 113, 114, 115, 142, 143, 144, 145,
     146, 147, 148, 149, 150
OFFSET 1,1
COMMENTS The number of steps is either S(k) = L(k) or S(k) = L(k) + 1
         and this sequence is the latter.
         S(k) = L(k) is A391108.
         Empirical observation, but without proof yet,
         shows that terms are all and only
         A003312(i+1) <= k <= A147788(i) for i >= 4
         (noting that the steps in A003312 yield the smallest integer
         m such that ceiling(2m/3) = k).
LINKS Laurent Lyaudet, <a href="/A391044/b391044_1.txt">Table of n,
        a(n) for n = 1..10000</a>
      Laurent Lyaudet, <a href="https://lyaudet.eu/laurent/Publi/
        Journaux/LL2025LAQNonBijective/
        LL2025LAQNonBijective_en_v4.pdf">
        On non-bijective tree-questionable-width</a>
```

```
      Laurent Lyaudet, <a href="/A391044/a391044.php.txt">
        PHP program to generate the sequence or the b-file</a>
EXAMPLE k = 2 is not a term since ceiling(2*2/3) = 2, it loops,
          and ceiling(log(2/2)/log(3/2)) = 0.
        k = 3 is not a term since ceiling(2*3/3) = 2, 1 step,
          and ceiling(log(3/2)/log(3/2)) = 1.
        k = 4 is not a term since ceiling(2*4/3) = 3, 2 steps,
          and ceiling(log(4/2)/log(3/2)) = 2.
        ...
        k = 10 is the first term since ceiling(2*10/3) = 7,
          ceiling(2*7/3) = 5, ceiling(2*5/3) = 4, hence 5 steps,
          and ceiling(log(10/2)/log(3/2)) = 4,
          thus we need to add 1 to have the number of steps
          (10 is the smallest such number).
        PROG (PHP) // See Lyaudet link.
        CROSSREFS Cf. A003312, A061420, A147788, A391108.
        KEYWORD nonn, changed
        AUTHOR Laurent Lyaudet, Nov 26 2025
```

They asked that I replace ln notation by log notation, but `https://oeis.org/wiki/Style_Sheet` says explicitly that we can use one or the other without any preference.

I had put the program in a file "/A391044/a391044.php.txt" uploaded separately. It is OEIS that added .txt after .php, *a priori* to avoid that the program is run on the server instead of being dowloaded.

```
<?php
/*
  _Laurent Lyaudet_, Nov 29 2025
  If this program halts without error message,
  then the output is correct.
*/
$aSteps = [2 => 0, 3 => 1];
// Outputs to std output:
// - the comma separated sequence if $bFile === false,
// - the b-file if $bFile === true.
$bFile = true;
$fThreeHalves = 3 / 2;
$iFound = 0;
$iMaxFound = 10000;
for($k = 4; $iFound < $iMaxFound; ++$k){
  if($k >= 2**53){
    exit("You may not have enough precision.");
  }
  if($k >= 2**62 - 2){
    exit("Signed integer overflow for the computations below.");
  }
```

```
// $iPreviousRank = (int) ceil($k*2/3)
$iPreviousRank = intdiv($k*2 + 2, 3);
$iSteps = 1 + $aSteps[$iPreviousRank];
/* IEEE 754 64-bit double : 53 bits of precision for $k
Rule of thumb:
 - we lose at most one bit of precision for each +,-,*,/
 - since we never have any double near the subnormals,
   an ULP less than 1 yields less than 1 bit of precision lost
>= 52 bits of precision for $k/2 and $fThreeHalves
PHP log uses C log function log($k/2)/log($fThreeHalves)
https://homepages.loria.fr/PZimmermann/papers/glibc240.pdf
shows that log functions in various variants of libc have known
errors that are less than 1ULP.
But the exact result is not always exact for double precision,
whilst exhaustive search is possible for single precision.
For the particular case of glibc:
-----------------------------------------------------------------------
Worst case error if |y| > 0x1p-4: 0.519 ULP (0.520 ULP without fma).
  0.5 + 2.06/N + abs-poly-error*2^56 ULP (+ 0.001 ULP without fma).
-> return y;
So 0x1p-4 = 0.0625 and $k/2 >= 2, log(2) ~= 0.693147181 > 0.0625
$fThreeHalves = 1.5, log(1.5) ~= 0.405465108 > 0.0625
Thus, we are in the case, where we lose less than 0.520 ULP,
wich is less than 1 bit of precision.
-----------------------------------------------------------------------
Otherwise, we take a big margin of 10 bits of precision lost in case
the implementation of log has large ULPs.
So we may assume that the log has at least 41 bits of precision.
*/
$fLog = log($k/2, $fThreeHalves);
$iCeil = (int) ceil($fLog);
// Both differences below are less than 1, so no bit of precision
// is for the integer part.
// We take again a big margin of 5 bits of precision lost.
if(
  ($iCeil - $fLog) < 1/(2**35)
  || ($fLog - ($iCeil - 1)) < 1/(2**35)
){
  exit(
    "For k = $k, Log is too close from an integer"
    ." to be certain there is no precision error."
  );
}
$iBound = 1 + $iCeil;
$aSteps[$k] = $iSteps;
if($iSteps === $iBound){
```

```
    ++$iFound;
    if($bFile){
      echo "$iFound $k\n";
    }
    else{
      echo "$k, ";
    }
  }
}
?>
```

Here is my second sequence addition proposal that will probably be rejected soon by the editors:

```
A391108
NAME Numbers k >= 2 for which the number of steps x -> ceiling(x*2/3)
     to reach 2 is S(k) = L(k)
     where L(k) = ceiling(log(k/2)/log(3/2)).
DATA 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 16, 17, 18, 19, 23, 24, 25,
     26, 27, 28, 35, 36, 37, 38, 39, 40, 41, 42, 52, 53, 54, 55, 56,
     57, 58, 59, 60, 61, 62, 63, 77, 78, 79, 80, 81, 82, 83, 84, 85,
     86, 87, 88, 89, 90, 91, 92, 93, 94, 116, 117, 118, 119, 120, 121,
     122, 123, 124, 125, 126
OFFSET 1,1
COMMENTS The number of steps is S(k) = A061420(k-2).
         S(k) is either S(k) = L(k) or S(k) = L(k) + 1
         and this sequence is the former.
         S(k) = L(k) + 1 is A391044.
         Empirical observation, but without proof yet,
         shows that terms are all and only
         k in {2,3,4,5,6,7,8,9} or
         A147788(i) < k < A003312(i+1) for i >= 5
         (noting that the steps in A003312 yield the smallest integer
         m such that ceiling(2m/3) = k).
LINKS Laurent Lyaudet, <a href="/A391108/b391108_1.txt">Table of n,
        a(n) for n = 1..10000</a>
      Laurent Lyaudet, <a href="https://lyaudet.eu/laurent/Publi/
        Journaux/LL2025LAQNonBijective/
        LL2025LAQNonBijective_en_v4.pdf">
        On non-bijective tree-questionable-width</a>
      Laurent Lyaudet, <a href="/A391108/a391108.php.txt">
        PHP program to generate the sequence or the b-file</a>
EXAMPLE k = 2 is a term since ceiling(2*2/3) = 2, it loops,
          and ceiling(log(2/2)/log(3/2)) = 0.
        k = 3 is a term since ceiling(2*3/3) = 2, 1 step,
          and ceiling(log(3/2)/log(3/2)) = 1.
        k = 4 is  a term since ceiling(2*4/3) = 3, 2 steps,
```

```
  and ceiling(log(4/2)/log(3/2)) = 2.
...
k = 10 is not a term since ceiling(2*10/3) = 7,
  ceiling(2*7/3) = 5, ceiling(2*5/3) = 4, hence 5 steps,
  and ceiling(log(10/2)/log(3/2)) = 4,
  thus we need to add 1 to have the number of steps
  (10 is the smallest such number).
PROG (PHP) // See Lyaudet link.
CROSSREFS Cf. A003312, A061420, A147788, A391044.
KEYWORD nonn, changed
AUTHOR Laurent Lyaudet, Nov 29 2025
```

And here is the uploaded program:

```
<?php
/*
  _Laurent Lyaudet_, Nov 29 2025
  If this program halts without error message,
  then the output is correct.
*/
$aSteps = [2 => 0, 3 => 1];
// Outputs to std output:
// - the comma separated sequence if $bFile === false,
// - the b-file if $bFile === true.
$bFile = true;
if($bFile){
  echo "1 2\n2 3\n";
}
else{
  echo "2, 3, ";
}
$fThreeHalves = 3 / 2;
$iFound = 2;
$iMaxFound = 10000;
for($k = 4; $iFound < $iMaxFound; ++$k){
  if($k >= 2**53){
    exit("You may not have enough precision.");
  }
  if($k >= 2**62 - 2){
    exit("Signed integer overflow for the computations below.");
  }
  // $iPreviousRank = (int) ceil($k*2/3)
  $iPreviousRank = intdiv($k*2 + 2, 3);
  $iSteps = 1 + $aSteps[$iPreviousRank];
  /* IEEE 754 64-bit double : 53 bits of precision for $k
  Rule of thumb:
   - we lose at most one bit of precision for each +,-,*,/
```

```
 - since we never have any double near the subnormals,
   an ULP less than 1 yields less than 1 bit of precision lost
>= 52 bits of precision for $k/2 and $fThreeHalves
PHP log uses C log function log($k/2)/log($fThreeHalves)
https://homepages.loria.fr/PZimmermann/papers/glibc240.pdf
shows that log functions in various variants of libc have known
errors that are less than 1ULP.
But the exact result is not always exact for double precision,
whilst exhaustive search is possible for single precision.
For the particular case of glibc:
---------------------------------------------------------------------
Worst case error if |y| > 0x1p-4: 0.519 ULP (0.520 ULP without fma).
  0.5 + 2.06/N + abs-poly-error*2^56 ULP (+ 0.001 ULP without fma).
-> return y;
So 0x1p-4 = 0.0625 and $k/2 >= 2, log(2) ~= 0.693147181 > 0.0625
$fThreeHalves = 1.5, log(1.5) ~= 0.405465108 > 0.0625
Thus, we are in the case, where we lose less than 0.520 ULP,
wich is less than 1 bit of precision.
---------------------------------------------------------------------
Otherwise, we take a big margin of 10 bits of precision lost in case
the implementation of log has large ULPs.
So we may assume that the log has at least 41 bits of precision.
*/
$fLog = log($k/2, $fThreeHalves);
$iCeil = (int) ceil($fLog);
// Both differences below are less than 1, so no bit of precision
// is for the integer part.
// We take again a big margin of 5 bits of precision lost.
if(
  ($iCeil - $fLog) < 1/(2**35)
  || ($fLog - ($iCeil - 1)) < 1/(2**35)
){
  exit(
    "For k = $k, Log is too close from an integer"
    ." to be certain there is no precision error."
  );
}
$iBound = $iCeil;
$aSteps[$k] = $iSteps;
if($iSteps === $iBound){
  ++$iFound;
  if($bFile){
    echo "$iFound $k\n";
  }
  else{
    echo "$k, ";
```

```
      }
    }
  }
?>
```

Currently, the worse of them is blaming me for using floats with double precision to compute a logarithm... And they are blind to the comment at the beginning:

```
If this program halts without error message,
then the output is correct.
```

For astronomical numbers, my program still has exact output without errors. And the generated b-file doesn't need to go beyond 20000...

Thanks God! Thanks Father! Thanks Jesus! Thanks Holy-Spirit!

References

R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, Reading, 1989.

L. Lyaudet. On non-bijective tree-questionable-width. *preprint*, abs/202506-4, 2025a. URL `https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/LL2025LAQNonBijective_en_v4.pdf`.

L. Lyaudet. Sur la largeur arborescente questionnable non-bijective. *preprint*, abs/202506-3, 2025b. URL `https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/LL2025LAQNonBijective_fr_v4.pdf`.

OEIS Foundation Inc. (2025). Entry A061420 in The On-Line Encyclopedia of Integer Sequences. URL `https://oeis.org/A061420`.

Archive

FR V1 2025/12/03:
`https://lyaudet.eu/laurent/Publi/Journaux/LL2025LPP/LL2025LPP_fr_v1.pdf`
EN V1 2025/12/03:
`https://lyaudet.eu/laurent/Publi/Journaux/LL2025LPP/LL2025LPP_en_v1.pdf`
FR V2 2025/12/07:
`https://lyaudet.eu/laurent/Publi/Journaux/LL2025LPP/LL2025LPP_fr_v2.pdf`
EN V2 2025/12/07 this version:
`https://lyaudet.eu/laurent/Publi/Journaux/LL2025LPP/LL2025LPP_en_v2.pdf`