Long division of positive integers

Gábor P. Nagy

Budapest University of Technology and Economics (Hungary)

May 4, 2021

Outline

Example

2 Theorems

3 Long division algorithm

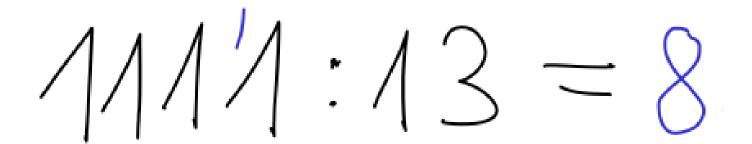
Outline

Example

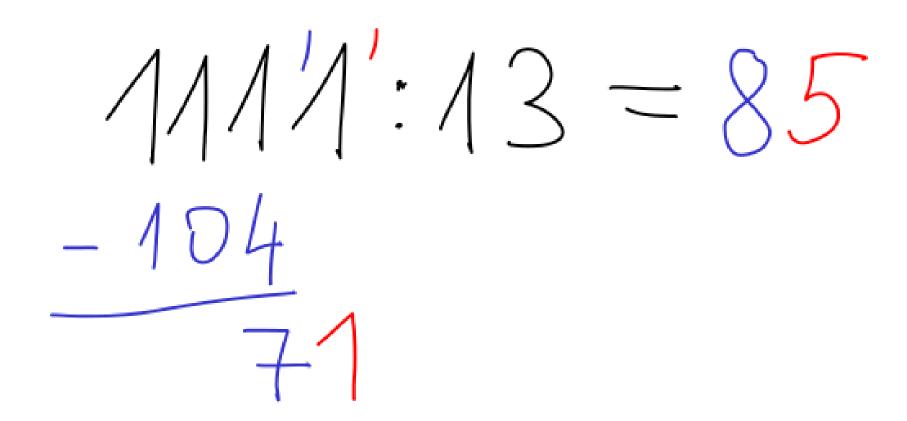
2 Theorems

3 Long division algorithm

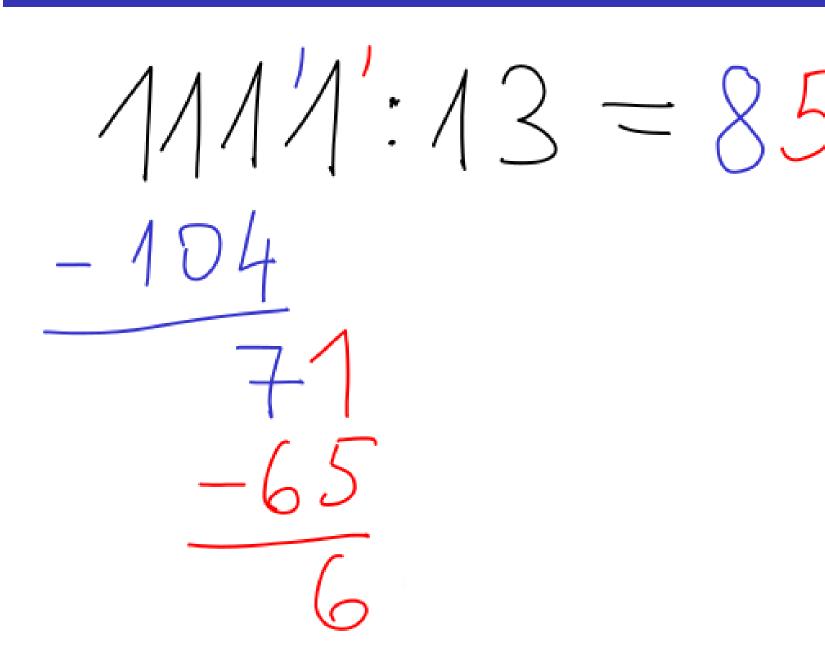


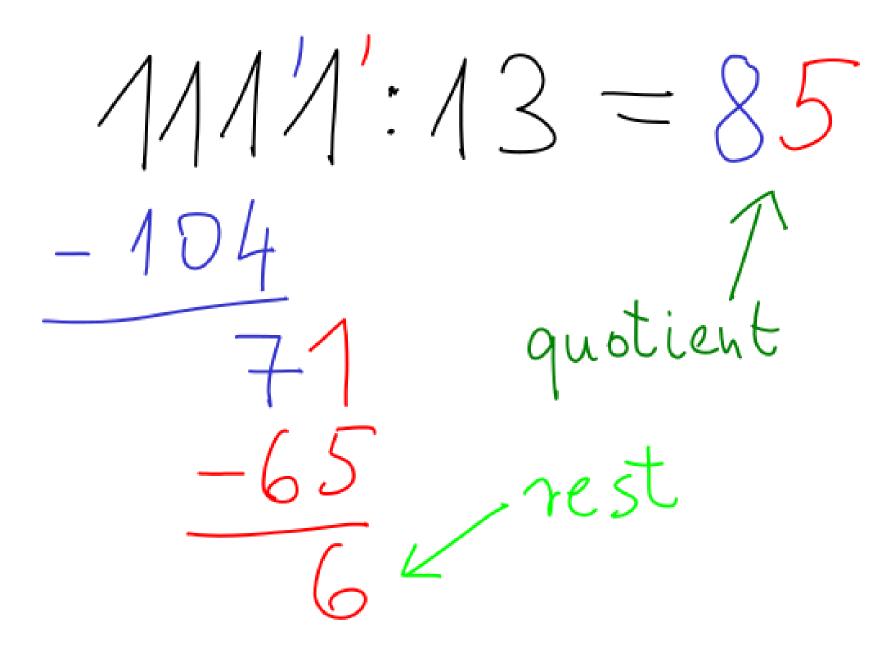


$$4/4/1:13 = 8$$
 -104
 -7



Long division of positive integers





1111 divided by 13 is 85 with remainder 6

85 13)1111 65

Long division bottleneck

MAGIC!!

Figure out the digits 8 and 5.

EASY: 8 = 111/13 with rest $7 = 111 - 8 \times 13$.

EASY: 5 = 71/13 with rest $6 = 71 - 5 \times 13$.

PROBLEM: This may need 3-digits arithmetic!!

IN GENERAL, with *n*-digit divisor, n + 1-digit arithmetic is needed.

Division with a one-digit number

```
t_NAT t_NAT::divide_by_digit(t_digit a) const {
 if (!a) {
    std::cerr << "Divison_by_zero!" << std::endl;</pre>
    exit(EXIT_FAILURE);
  t_digit rem = 0;
  t_NAT n = *this;
  for (auto i = n.arr.rbegin(); i != n.arr.rend(); ++i) {
    int b = RADIX * rem + (*i);
    *i = t_digit(b / a);
    rem = t_digit(b \% a);
 // cleaning leading zeros of n ...
 return n;
}
```

7/19

Base b representation recap

Generic *n*-digit integer in base *b*:

$$a = (a_{n-1}a_{n-2} \dots a_1 a_0)_b$$

$$= a_{n-1}b^{n-1} + \dots + a_1b + a_0 \qquad (0 \le a_i \le b - 1)$$

$$\left\lfloor \frac{a}{b} \right\rfloor = a_{n-1}b^{n-2} + \dots + a_1$$

$$= (a_{n-1}a_{n-2} \dots a_1)_b$$

$$b^n = (10 \dots 0)_b$$

$$= \text{smallest } n + 1 \text{-digit integer}$$

$$> a$$

$$\ge a_{n-1}b^{n-1} = (a_{n-1}0 \dots 0)_b$$

Outline

Example

2 Theorems

3 Long division algorithm

Dividing an n + 1-digit number by an n-digit number

Problem 1

Given positive integers u, v in base b:

$$u = (u_n u_{n-1} \dots u_0)_b$$

 $v = (v_{n-1} v_{n-2} \dots v_0)_b$

We assume $v_{n-1} \neq 0$ and u/v < b.

Find an algorithm to determine $\mathbf{q} = \lfloor \mathbf{u}/\mathbf{v} \rfloor$ (rapidly, even for large b).

- $u/v < b \iff u/b < v \iff \lfloor u/b \rfloor < v \iff (u_n u_{n-1} \dots u_1)_b < (v_{n-1} v_{n-2} \dots v_0)_b$
- If r = u qv, then q is the unique integer such that $0 \le r < v$.

Obvious approach

Make a guess about q, based on the two most significant digits of u and v.

Long division of positive integers

SPOILER: Good idea, except when v_{n-1} is small.

Make a guess about $q = \lfloor u/v \rfloor$

Problem 1

Given positive integers *u*, *v* in base *b*:

$$u = (u_n u_{n-1} \dots u_0)_b$$

 $v = (v_{n-1} v_{n-2} \dots v_0)_b$

We assume $v_{n-1} \neq 0$ and u/v < b.

Find an algorithm to determine $q = \lfloor u/v \rfloor$ (rapidly, even for large b).

Define
$$\hat{q} = \left[\frac{u_n b + u_{n-1}}{v_{n-1}} \right]$$
. If $\hat{q} \ge b$ then set $\hat{q} = b - 1$.

In the previous example, u = 111, v = 13,

$$u_n b + u_{n-1} = 11$$
, $v_{n-1} = 1$, $\lfloor 11/1 \rfloor = 11$,

hence $\hat{q} = 10 - 1 = 9$.

Lower bound on \hat{q}

Theorem A

In the notation $\hat{q} = \min\left(\left\lfloor \frac{u_n b + u_{n-1}}{v_{n-1}} \right\rfloor, b-1\right)$, we have $\hat{q} \geq q$.

Proof. Certainly true if $\hat{q} = b - 1$. Otherwise

$$\frac{u_n b + u_{n-1}}{v_{n-1}} - \left\lfloor \frac{u_n b + u_{n-1}}{v_{n-1}} \right\rfloor \le 1 - \frac{1}{v_{n-1}} \Longrightarrow \hat{q} v_{n-1} \ge u_n b + u_{n-1} - v_{n-1} + 1.$$

From $b^{n-1} > u_{n-2}b^{n-2} + \cdots + u_0 = (u_{n-2} \dots u_0)_b$ follows

$$u - \hat{q}v \le u - \hat{q}v_{n-1}b^{n-1}$$

$$\le u_nb^n + \dots + u_0 - (u_nb^n + u_{n-1}b^{n-1} - v_{n-1}b^{n-1} + b^{n-1})$$

$$= u_{n-2}b^{n-2} + \dots + u_0 - b^{n-1} + v_{n-1}b^{n-1} < v_{n-1}b^{n-1} \le v.$$

Long division of positive integers

Since $u - \hat{q}v < v$, we must have $\hat{q} > u/v - 1$.

$$u/v - 1 \ge \lfloor u/v \rfloor - 1 = q - 1$$
 implies $\hat{q} > q - 1$.

A lemma

Lemma

- ① If $v = b^{n-1} = (10...0)_b$, then $q = \hat{q}$.
- 2 If $v \neq b^{n-1}$ then

$$\hat{q} < \frac{u}{v - b^{n-1}}.$$

Proof. (1) $q = \lfloor u/v \rfloor = \lfloor u_n b + u_{n-1} + (\cdots)/b^{n-1} \rfloor = \lfloor u_n b + u_{n-1} \rfloor = \hat{q}$. (2) We have

$$\hat{q} \leq \frac{u_n b + u_{n-1}}{v_{n-1}} = \frac{u_n b^n + u_{n-1} b^{n-1}}{v_{n-1} b^{n-1}} \leq \frac{u}{v_{n-1} b^{n-1}} < \frac{u}{v - b^{n-1}}.$$

The **last step** follows from

$$v - v_{n-1}b^{n-1} = (v_{n-2} \dots v_0)_b < (10 \dots 0) = b^{n-1}.$$

Upper bound on \hat{q}

Lemma

- ① If $v = b^{n-1} = (10...0)_b$, then $q = \hat{q}$.
- 2 If $v \neq b^{n-1}$ then $\hat{q} < \frac{u}{v-b^{n-1}}$.

Theorem B

If $v_{n-1} \ge \lfloor b/2 \rfloor$, then $\hat{q} \le q + 2$.

Proof. Assume $\hat{q} \ge q + 3$. By Lemma (1), $v - b^{n-1} \ne 0$, and

$$3 \leq \hat{q} - q < \frac{u}{v - b^{n-1}} - \left(\frac{u}{v} - 1\right) = \frac{uv - uv + ub^{n-1}}{v(v - b^{n-1})} + 1 = \frac{u}{v} \left(\frac{b^{n-1}}{v - b^{n-1}}\right) + 1.$$

Therefore

$$\frac{u}{v} > 2\left(\frac{v - b^{n-1}}{b^{n-1}}\right) \ge 2\left(\frac{v_{n-1}b^{n-1} - b^{n-1}}{b^{n-1}}\right) = 2(v_{n-1} - 1).$$

This implies $q = \lfloor u/v \rfloor \ge 2(v_{n-1} - 1)$.

Upper bound on \hat{q} (cont.)

Theorem B

If $v_{n-1} \geq \lfloor b/2 \rfloor$, then $\hat{q} \leq q + 2$.

The inequalities $\hat{q} \le b - 1$, $\hat{q} \ge q + 3$ and $q \ge 2(v_{n-1} - 1)$ imply $b - 4 \ge \hat{q} - 3 \ge q \ge 2(v_{n-1} - 1)$.

We have therefore

$$v_{n-1} \le b/2 - 1 < \lfloor b/2 \rfloor$$
,

which finishes the proof.

Corollary

If $v_{n-1} \ge \lfloor b/2 \rfloor$, then $\hat{q} - 2 \le q \le \hat{q}$.

- Important: The conclusion is independent on the radix b.
- The condition that $v_{n-1} \ge \lfloor b/2 \rfloor$ is a **normalization requirement.**
- To ensure it, **multiply both** u and v by $d = 2^k$ such that $\lfloor b/2 \rfloor b^{n-1} \le dv < b^n$.

Algorithm C to find q

Algorithm C (Division of an n + 1-digit number by an n-digit number)

Given positive integers *u*, *v* in base *b*:

$$u = (u_n u_{n-1} \dots u_0)_b$$

 $v = (v_{n-1} v_{n-2} \dots v_0)_b$

We assume u/v < b and $v_{n-1} \ge \lfloor b/2 \rfloor$. The following algorithm returns $\lfloor u/v \rfloor$.

- Set $q \leftarrow \min(\lfloor (u_n b + u_{n-1})/v_{n-1} \rfloor, b-1)-2$ and $r \leftarrow u qv$.
- While $r \ge v$, set $q \leftarrow q + 1$ and $r \leftarrow r v$.
- Return q.

Important: By the Corollary, no matter how large the radix b is, one makes step (2) at most twice.

Remark: One can make Algorithm C more efficient by testing on v_{n-2} .

Outline

Example

2 Theorems

3 Long division algorithm

Algorithm D – division of non-negative integers

Given non-negative integers

$$u = (u_{m+n-1}u_{m+n-2}...u_1u_0)_b,$$
 $v = (v_{n-1}v_{n-2}...v_1v_0)_b,$ where $v_{n-1} \neq 0$ and $n > 1$. We form the radix- b quotient $q = \lfloor u/v \rfloor$ and the remainder $r = u \pmod{v}$ $q = (q_m q_{m-1}...q_1q_0)_b,$ $r = (r_{n-1}r_{n-2}...r_1r_0)_b.$

Algorithm D

- (Normalize.) Set $d \leftarrow 2^k$ such that $\lfloor b/2 \rfloor b^{n-1} \leq dv < b^n$. Set $(u_{m+n}u_{m+n-1} \dots u_1 u_0)_b \leftarrow du$. Set $(v_{n-1}v_{n-2} \dots v_1 v_0)_b \leftarrow dv$.
- ② (Initialize *j*.) Set $j \leftarrow m$.
- (Calculate q_j .) Set the subarray $u^* \leftarrow (u_{j+n}u_{j+n-1} \dots u_j)_b$. Apply Algorithm C with input u^* and v to compute q_i .
- 4 (Multiply and subtract.) Replace the subarray u^* by $u^* q_i v$.
- (Loop on j.) Decrease j by one. Now if $j \ge 0$, go back to (D3).
- (Unnormalize.) Return q as quotient and u^*/d as remainder.