
Front matter

lang: ru-RU title: Целочисленная арифметика многократной точности author: Воробьев А.О.

institute: RUDN University, Moscow, Russian Federation

date: 2023 Moscow, Russia

Formatting

toc: false slide_level: 2 theme: metropolis header-includes:

- \metroset
- \makeatletter
- \beamer@ignorenonframefalse
- \makeatother aspectratio: 43 section-titles: true

Цель работы

Цель работы

Ознакомление с алгоритмами целочисленной арифметики многократной точности, а также их последующая программная реализация.

Задачи

Задачи

1. Реализовать алгоритм сложения неотрицательных целых чисел.
2. Реализовать алгоритм вычитания неотрицательных целых чисел.
3. Реализовать алгоритм умножения неотрицательных целых чисел столбиком.
4. Реализовать алгоритм быстрого столбика.
5. Реализовать алгоритм деления многоразрядных целых чисел.

Реализация

блок данных

1. Написал блок данных (рис. -@fig:001)

```

import math

2 # надо ввести данные сначала
3 u = "12345"
4 v = "56789"
5 b = 10
6 n = 5

```

Алгоритм сложения неотрицательных целых чисел

2. Написал алгоритм сложения неотрицательных целых чисел (рис. -@fig:002)

```

7 n
8 k = 0
9
10
11
12
13 w = []
14 for i in range(1, n + 1):
15     w.append((int(u[n - i]) + int(v[n - i]) + k) % b)
16
17     k = (int(u[n - i]) + int(v[n - i]) + k) // b
18     j = j - 1
19 w.reverse()
20 return int(w)

```

Алгоритм вычитания неотрицательных целых чисел

3. Написал алгоритм вычитания неотрицательных целых чисел (рис. -@fig:003)

```

        56789"
v = "12345"
j
26 j = n
27 k = 0
28 w = []
29 for i in range(1, n + 1):
30     w.append((int(u[n - i]) - int(v[n - i]) + k) % b)
31
32     k = (int(u[n - i]) - int(v[n - i]) + k) // b
33     j = j - 1
34 v.reverse()
35 ^i(w)

```

Алгоритм умножения неотрицательных целых чисел столбиком первая часть

4. Написал алгоритм умножения неотрицательных целых чисел столбиком(рис. -@fig:004)(рис. -@fig:005)

```

        3456"
        7890"
        = 6
        = 4
w = []
44 for i in range(m + n):
45     w.append(0)
46     j = m
47
48
49 def step6():
50     global j
51     global w
52     j = j - 1
53     if j > 0:
54         step2()
55     if j == 0:
56         print(w)
57
58
59 def step2():
60     global v
61     global w
62     global j
63     if j == m:
64         j = j - 1
65         if int(v[j]) == 0:
66             w[j] = 0
67             step6()

```

Алгоритм умножения неотрицательных целых чисел столбиком первая часть

```
        k
    global t
    global i
    if i == n:
        t = t - 1
        t = int(u[i]) * int(v[j]) + w[i + j] + k
        w[i + j] = t % b
        k = t // b
7
78
79
80
81 def step5():
82     global i
83     global w
84     global j
85     global k
86     i = i - 1
87     if i > 0:
88         step4()
89     else:
90         w[j] = k
91
92
93 step2()
94 i = n
95 k = 0
96 t = 1
97 tep4()
98 ns()
99 )
```

Алгоритм Быстрого столбика

5. Написал алгоритм быстрого столбика (рис. -@fig:006)

```
    s "6789"
    n = 4
    b = 10
    w1 = []
    100 for i in range(n + 2):
    110     w1.append(0)
    111     t1 = 0
    112 for s1 in range(0, m + n):
    113     for i1 in range(0, s1 + 1):
    114         if n - i1 > n or m - s1 + i1 > m or n - i1 < 0 or m - s1 + i1 < 0 or m - s1 + i1
            - 1 < 0:
                continue
            t1 = t1 + (int(u[n - i1 - 1]) * int(v[m - s1 + i1 - 1]))
            "m + n - s1 - 1 = t1 % b
            "h, floor(t1 / b)
```

Алгоритм деления многоразрядных целых чисел

6. Написал алгоритм деления многоразрядных целых чисел (рис. -@fig:007)(рис. -@fig:008)

```

    '56789'
    a = 4
    b = 10
    127 q = []
    128 for j in range(n - t):
    129     q.append(0)
    130 r = []
    131 for j in range(t):
    132     r.append(0)
    133
    134 while int(u) >= int(v) * (b**(n - t)):
    135     q[n - t] = q[n - t] + 1
    36 u = int(u) - int(v) * (b**(n - t))
    u = str(u)
    for i in range(n, t + 1, -1):
        u = str(v)
        if int(u) > int(v[t]):
            h = 1

```

Алгоритм деления многоразрядных целых чисел

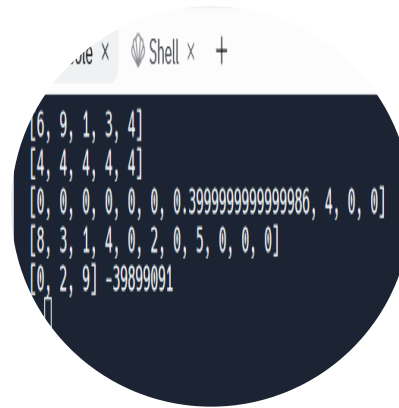
```

    u[i - t - 1] = math.floor((int(u[i]) * b + int(u[i - 1])) /
    147 while (int(q[i - t - 1]) * (int(v[t]) * b + int(v[t - 1])) > int(u[i],
    (b**2) + int(u[i - 1]) * b + int(u[i - 2]))):
    148 q[i - t - 1] = q[i - t - 1] - 1
    149 u = (int(u) - q[i - t - 1] * b**(i - t - 1) * int(v))
    150 if u < 0:
    151 u = int(u) + int(v) * (b**(i - t - 1))
    q[i - t - 1] = q[i - t - 1] + 1
    u = u
    r)

```

Результат

7. Получил результат (рис. -@fig:009)

A terminal window with a dark blue background and white text. The window has a title bar with 'Shell x' and a plus icon. The output consists of five lines of array representations: [6, 9, 1, 3, 4], [4, 4, 4, 4, 4], [0, 0, 0, 0, 0, 0.39999999999999986, 4, 0, 0], [8, 3, 1, 4, 0, 2, 0, 5, 0, 0], and [0, 2, 9] -39899091.

```
Shell x +  
[6, 9, 1, 3, 4]  
[4, 4, 4, 4, 4]  
[0, 0, 0, 0, 0, 0.39999999999999986, 4, 0, 0]  
[8, 3, 1, 4, 0, 2, 0, 5, 0, 0]  
[0, 2, 9] -39899091
```

Вывод

Изучал задачу представления больших чисел, познакомились с вычислительными алгоритмами и реализовали их.