

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Презентация ПО

ЛАБОРАТОРНОЙ РАБОТЕ №8

дисциплина: Математические основы защиты информации и информационной безопасности

Студент: Хиссен Али Уэддей

Группа: НПМмд-02-20

Ст. билет № 1032209306

Цель работы

Исследование алгоритмов работы с большими целыми числами.

выполнение алгоритмрв

1 Сложение неотрицательных целых чисел

*Вход. Два неотрицательных числа $u = u_1u_2 \dots u_n$ и $v = v_1v_2 \dots v_n$; разрядность чисел n ; основание системы счисления b .

*Выход. Сумма $w = w_0w_1 \dots w_n$, где w_0 - цифра переноса, всегда равная 0 либо 1.

1. Присвоить $j = n, k = 0$ (j идет по разрядам, k следит за переносом).
2. Присвоить $w_j = (u_j + v_j + k) \pmod{b}$, где $k = \left\lfloor \frac{u_j + v_j + k}{b} \right\rfloor$.
3. Присвоить $j = j - 1$. Если $j > 0$, то возвращаемся на шаг 2; если $j = 0$, то присвоить $w_0 = k$ и результат: w .

```
1
2 def Algorithm1():
3
4     u=(input("faite entrer un nombre superieur a 0 \n"))
5     v=(input("faite entrer un nombre superieur a 0 \n"))
6     b=int((input("faite entrer un nombre superieur a 0 \n")))
7     list_u=[i for i in u]
8
9     list_v=[i for i in v]
10
11     list_w=[]
12
13     n=len(list_u)
14
15     j=n-1
16
17     k=0
18
19     if(len(list_u)!=len(list_v)):
20         print("impossible de calculer w")
21
22     else:
23
24         w = ''
25
26         while(j>=0):
27
28             resultat=int(list_u[j]) + int(list_v[j]) + k
29
30             list_w.append(resultat%b)
31
32             k=int(resultat/b)
```

```

31
32         k=int(resultat/b)
33
34         j=j-1
35
36         #print(list_w)
37
38         liste_w=[]
39
40         ss=len(list_w)-1
41
42         #print(ss ,k)
43
44         while(ss>=0):
45
46             #liste_w.append(list_w[ss])
47
48             w+=str(list_w[ss])
49
50             ss=ss-1
51         w=str(k) + w
52
53         #type(u[0]),list_u + list_v ,liste_w
54
55     return int(w)

```

```

[]: 1 Algorithm1()

```

```

faite entrer un nombre superieur a 0
123
faite entrer un nombre superieur a 0
123
faite entrer un nombre superieur a 0
10

```

```

[]: 246

```

2 Вычитание неотрицательных целых чисел

*Вход. Два неотрицательных числа $u = u_1u_2 \dots u_n$ и $v = v_1v_2 \dots v_n$, $u > v$; разрядность чисел n ; основание системы счисления b .

*Выход. Разность $w = w_0w_1 \dots w_n = u - v$.

1. Присвоить $j = n, k = 0$ (k – заём из старшего разряда).
2. Присвоить $w_j = (u_j - v_j + k) \pmod{b}$; $k = \left\lfloor \frac{u_j - v_j + k}{b} \right\rfloor$.
3. Присвоить $j = j - 1$. Если $j > 0$, то возвращаемся на шаг 2; если $j = 0$, то результат: w .

```

: 1 def Algorithm2():
2     u1=(input("faite entrer un nombre superieur a 0 \n"))
3     v1=(input("faite entrer un nombre superieur a 0 \n"))
4     b=int((input("faite entrer un nombre >2 \n")))
5
6     list_u=[i for i in u1]
7
8     list_v=[i for i in v1]
9
10    list_w=[]
11
12    n=len(list_u)
13
14    j=n-1
15
16    k=0
17
18    #print(list_u)
19
20    if(len(list_u)!=len(list_v)):
21        print("impossible de calculer w")
22
23    else:
24
25        w = ''
26
27        while(j>=0):
28
29            resultat=int(list_u[j]) - int(list_v[j]) + k
30
31            list_w.append(resultat%b)
32
33            k=int(resultat/b)
34

```

```
32
33         k=int(resultat/b)
34
35         j=j-1
36
37         #print(list_w)
38
39         liste_w=[]
40
41         ss=len(list_w)-1
42
43         #print(ss ,k)
44
45         while(ss>=0):
46
47             w+=str(list_w[ss])
48
49             ss=ss-1
50
51         return int(w)  #,int(u1)-int(v1)
```

1 Algorithm2()

faite entrer un nombre superieur a 0

320

faite entrer un nombre superieur a 0

300

faite entrer un nombre >2

10

20

3 Умножение неотрицательных целых чисел столбиком

*Вход. Числа $u = u_1 u_2 \dots u_n$, $v = v_1 v_2 \dots v_m$; основание системы счисления b .

*Выход. Произведение $w = uv = w_1 w_2 \dots w_{m+n}$.

1. Выполнить присвоения: $w_{m+1} = 0, w_{m+2} = 0, \dots, w_{m+n} = 0, j = m$ (j перемещается по номерам разрядов числа v от младших к старшим).
2. Если $v_j = 0$, то присвоить $w_j = 0$ и перейти на шаг 6.
3. Присвоить $i = n, k = 0$ (значение i идет по номерам разрядов числа u , k отвечает за перенос).
4. Присвоить $t = u_i \cdot v_j + w_{i+j} + k, w_{i+j} = t \pmod{b}, k = \left\lfloor \frac{t}{b} \right\rfloor$.
5. Присвоить $i = i - 1$. Если $i > 0$, то возвращаемся на шаг 4, иначе присвоить $w_j = k$.
6. Присвоить $j = j - 1$. Если $j > 0$, то вернуться на шаг 2. Если $j = 0$, то результат: w .

```

|: 1 def Algorithm3():
2     u=(input("faite entrer un nombre superieur a 0 \n"))
3     v=(input("faite entrer un nombre superieur a 0 \n"))
4     b=int((input("faite entrer un nombre >2 \n")))
5     list_u=[i for i in u]
6     list_v=[i for i in v]
7     n=len(list_u)
8     m=len(list_v)
9     mn=m+n
10    list_w=[0]*(mn)
11
12    for j in range(m-1, -1, -1):
13        if list_v[j] != 0:
14            k = 0
15            for i in range (n-1, -1, -1):
16                t = int(list_u[i]) * int(list_v[j]) + k + list_w[i+j+1]
17                list_w[i + j + 1] = t % b
18                k = t // b
19            list_w[j] = k
20
21    k=0
22    j=m
23
24    return (list_w),
25

```

```
: 1 Algorithm3()
   faite entrer un nombre supperieur a 0
   123
   faite entrer un nombre supperieur a 0
   10
   faite entrer un nombre >2
   10
: ([0, 1, 2, 3, 0],)
```


4 Быстрый столбик

*Вход. Числа $u = u_1 u_2 \dots u_n$, $v = v_1 v_2 \dots v_m$; основание системы счисления b .

*Выход. Произведение $w = uv = w_1 w_2 \dots w_{m+n}$.

1. Присвоить $t = 0$.
2. Для s от 0 до $m + n - 1$ с шагом 1 выполнить шаги 3 и 4.
3. Для i от 0 до s с шагом 1 выполнить присвоение $t = t + u_{n-i} \cdot v_{m-s+i}$.

4. Присвоить $w_{m+n-s} = t \pmod{b}$, $t = \left\lfloor \frac{t}{b} \right\rfloor$. Результат: w .

```
def Algorithm4():  
  
    u1=(input("faite entrer un nombre supperieur a 0 \n"))  
    v1=(input("faite entrer un nombre supperieur a 0 \n"))  
    b=int((input("faite entrer un nombre >2 \n")))  
  
    w=''   
  
    list_u=[i for i in u1]  
    list_v=[i for i in v1]  
  
    n=len(list_u) - 1  
    m=len(list_v) - 1  
  
    mn=m+n  
  
    list_w=[0]*(mn+1)  
  
    t=0  
  
    for s in range(0,mn+1):  
        for i in range(0,s+1):  
            t= t+ int(list_u[n-i]) * int(list_v[m-s+i])  
  
            #print(t)  
  
            list_w[mn-s]=t%b
```

```
32     list_w[mn-s]=t%b
33
34     w=str(list_w[mn-s]) + w
35     #list_w.appendst_w[mn-s]+w
36     (t%b)
37
38     t=t//b
39
40
41     return int(w)
```

9]: 1 Algorithm4()

faite entrer un nombre superieur a 0
100
faite entrer un nombre superieur a 0
100
faite entrer un nombre >2
10

9]: 10000

5 Деление многоразрядных целых чисел

*Вход. Числа $u = u_n \dots u_1 u_0$, $v = v_t \dots v_1 v_0$, $n \geq t \geq 1$, $v_t \neq 0$.

*Выход. Частное $q = q_{n-t} \dots q_0$, остаток $r = r_t \dots r_0$.

1. Для j от 0 до $n - t$ присвоить $q_j = 0$.
2. Пока $u \geq vb^{n-t}$, выполнять: $q_{n-t} = q_{n-t} + 1$, $u = u - vb^{n-t}$.
3. Для $i = n, n-1, \dots, t+1$ выполнять пункты 3.1 – 3.4: 3.1. если $u_i \geq v_t$, то присвоить $q_{i-t-1} = b - 1$, иначе присвоить $q_{i-t-1} = \frac{u_i b + u_{i-1}}{v_t}$. 3.2. пока $q_{i-t-1}(v_t b + v_{t-1}) > u_i b^2 + u_{i-1} b + u_{i-2}$ выполнять $q_{i-t-1} = q_{i-t-1} - 1$. 3.3. присвоить $u = u - q_{i-t-1} b^{i-t-1} v$. 3.4. если $u < 0$, то присвоить $u = u + v b^{i-t-1}$, $q_{i-t-1} = q_{i-t-1} - 1$.
4. $r = u$. Результат: q и r .

```

1 def Algorithm5():
2     u1=(input("faite entrer un nombre superieur a 0 \n"))
3     v1=(input("faite entrer un nombre superieur a 0 \n"))
4     b=int((input("faite entrer un nombre >2 \n")))
5
6     list_u=[i for i in u1]
7
8     list_v=[i for i in v1]
9
10    n=len(list_u) - 1
11
12    t=len(list_v) - 1
13
14    #ist_q=np.ones((mn+1))
15
16    u_int,v_int=int(u1),int(v1)
17
18    list_q=[0]*(n-t+1)
19
20    while(u_int>=v_int*b**(n-t)):
21
22        list_q[n-t]=list_q[n-t] + 1
23
24        u_int-=v_int*b**(n-t)
25
26    for i in range(n,t,-1):
27        if list_u[i] >= list_v[t]:
28            list_q[i-t-1] = b-1
29
30        else:
31
32            list_q[i-t-1] = (list_u[i]*b + list_u[i-1] )// list_v[t]
33

```

```

33
34
35     while list_q[i-t-1]*(list_v[t]*b + list_v[t-1]) > (list_u[i]*b**2 + list_u[i-1]*b + list_u[i-2])
36         list_q[i-t-1] -= 1
37     u_int -= list_q[i-t-1] * (b ** (i-t-1)) * v_int
38 if u_int < 0:
39     u_int += v_int * b**(i-t-1)
40     list_q[i-t-1] -= 1
41
42 q = int("".join(map(str, list_q)))
43 r = u_int
44
45 print(list_q, f"q = {q} r = {r}")
46

```

1 Algorithm5()

faite entrer un nombre superieur a 0

55

faite entrer un nombre superieur a 0

25

faite entrer un nombre >2

Вывод

так мы изучили исследование алгоритмов работы с большими целыми, познакомились с вычислительными алгоритмами.