Отчет по лабораторной работе 8

Дисциплина: Математические основы защиты информации и информационной безопасности

Дяченко З. К.

23 декабря 2022

Российский университет дружбы народов, Москва, Россия

Прагматика выполнения лабораторной работы

Данная лабораторная работа выполнялась мной для приобретения практических навыков реализации алгоритмов целочисленной арифметики многократной точности.

Цель выполнения лабораторной работы

Ознакомится и реализовать программно алгоритмы целочисленной арифметики многократной точности.

Задачи выполнения лабораторной работы

Реализовать программно алгоритм сложения неотрицательных чисел (рис. - fig. 1).

```
In [54]: def alg_1(u, v, b):
             u=[int(a) for a in str(u)]
             v=[int(a) for a in str(v)]
             n=len(u)
             k=0
             w=[0]*(n+1)
             for j in range (n-1, -1, -1):
                 w[j+1]=(u[j]+v[j]+k)%b
                 k=math.floor((u[j]+v[j]+k)/b)
             w[0]=math.floor(k)
             print(w)

In [55]: alg_1(125,125,10)

         [0, 2, 5, 0]

In [56]: alg_1(125000,125000,10)

         [0, 2, 5, 0, 0, 0, 0]
```

Рис. 1: Реализация алгоритма сложения неотрицательных чисел

Задачи выполнения лабораторной работы

Реализовать программно алгоритм вычитания неотрицательных чисел (рис. - fig. 2).

```
In [57]: def alg_2(u,v,b):
             if (u>v):
                 u=[int(a) for a in str(u)]
                 v=[int(a) for a in str(v)]
                 n=len(u)
                 k=0
                 w=[0]*n
                 for j in range (n-1, -1, -1):
                     w[j]=(u[j]-v[j]+k)%b
                     k=math.floor((u[j]-v[j]+k)/b)
                 print(w)
             else:
                 print("Первое число должно быть больше второго")

In [61]: alg_2(125,100,10)
         [0, 2, 5]

In [63]: alg_2(125000,100000,10)
         [0, 2, 5, 0, 0, 0]
```

Рис. 2: Реализация алгоритма вычитания неотрицательных чисел

Задачи выполнения лабораторной работы

Реализовать программно алгоритм умножения неотрицательных целых чисел столбиком (рис. - fig. 3).

```
In [83]:
def alg_3(u,v,b):
    u=[int(a) for a in str(u)]
    v=[int(a) for a in str(v)]
    n=len(u)
    m=len(v)
    w=[0]*(n+m)
    for j in range(m-1, -1, -1):
        if (v[j]==0):
            w[j]=0
            continue
        k=0
        for i in range(n-1, -1, -1):
            t=u[i]*v[j]+w[i+j+1]+k
            w[i+j+1]=t%b
            k=math.floor(t/b)
        w[j]=k
    print(w)

In [85]:
alg_3(11,11,10)

[0, 1, 2, 1]

In [86]:
alg_3(11000,11000,10)

[0, 1, 2, 1, 0, 0, 0, 0, 0, 0]
```

Рис. 3: Реализация алгоритма умножения неотрицательных целых чисел столбиком

Задачи выполнения лабораторной работы

Реализовать программно алгоритм быстрого столбика (рис. - fig. 4).

In [96]:
```
def alg_4(u,v, b):
    u=[int(a) for a in str(u)]
    v=[int(a) for a in str(v)]
    n=len(u)
    m=len(v)
    w=[0]*(n+m)
    t=0
    for s in range(0, m+n-1, 1):
        for i in range(0, s):
            t=t+u[n-i-1]*v[m-s+i-1]
        w[m+n-s-1]=t%b
        t=math.floor(t/b)
    print(w)
```

In [98]:
```
alg_4(1000, 1500, 10)
```

```
[0, 1, 5, 0, 0, 0, 0, 0]
```

Рис. 4: Реализация алгоритма быстрого столбика

Задачи выполнения лабораторной работы

Реализовать программно алгоритм деления многоразрядных целых чисел (рис. - fig. 5).

In [144]:
```
def alg_5(u,v,b):
    u_p=[int(a) for a in str(u)]
    v_p=[int(a) for a in str(v)]
    n=len(u_p)-1
    t=len(v_p)-1
    if (n>=t and t>=1 and v_p[0]!=0):
        q=[0]*(n-t+1)
        while (u>=v*math.pow(b, (n-t))):
            q[n-t]=q[n-t]+1
            u=u-v*math.pow(b, (n-t))
        u=int(u)
        u_p=[int(a) for a in str(u)]
        v_p=[int(a) for a in str(v)]
        n=len(u_p)-1
        t=len(v_p)-1
        for i in range(n, t+1-1, -1):
            if (u_p[i]>=v_p[t]):
                q[i-t-1]=b-1
            else:
                q[i-t-1]=(u_p[i]*b+u_p[i-1])/v_p[t]
            while ((q[i-t-1]*(v_p[t]*b+v_p[t-1]))>(u_p[i]*b*b+u_p[i-1]*b+u_p[i-2])):
                q[i-t-1]=q[i-t-1]-1
            u=int(''.join(map(str, u_p)))
            v=int(''.join(map(str, v_p)))
            u=u-q[i-t-1]*math.pow(b, (i-t-1))*v
            if(u<0):
                    u=u+v*math.pow(b,(i-t-1))
                    q[i-t-1]=q[i-t-1]-1
        r=u
        q.reverse()
        q=int(''.join(map(str, q)))
        print(q, r)
    else:
        print("Неверно введены числа")
```

In [145]:
```
alg_5(100, 10, 10)
```

```
10 0
```

Рис. 5: Реализация алгоритма деления многоразрядных целых чисел

Результаты выполнения лабораторной работы

Результатом выполнения работы стала реализация алгоритмов целочисленной арифметики многократной точности, что отражает проделанную мной работу и полученные новые знания.