

Федеральное государственное автономное образовательное  
учреждение высшего образования  
Университет ИТМО

Факультет инфокоммуникационных технологий

## **Алгоритмы и структуры данных:**

**Отчёт по лабораторной работе №7: Динамическое  
программирование**

Выполнил:  
**Бочкарь Артём Артёмович**

Группа: **K32392**

Преподаватели:  
**Артамонова В. Е.**

Санкт-Петербург 2023 г.

## Задача №2: Примитивный калькулятор

В данном задании надо было реализовать калькулятор, который может выполнять следующие три операции с текущим числом  $x$ : умножить  $x$  на 2, умножить  $x$  на 3 или прибавить 1 к  $x$ . Во входном файле дано одно целое число  $n$ ,  $1 \leq n \leq 10^6$ . Надо посчитать минимальное количество операций, необходимых для получения  $n$  из числа 1. В первой строке надо вывести минимальное число  $k$  операций. Во второй – последовательность промежуточных чисел  $a_0, a_1, \dots, a_{k-1}$  таких, что  $a_0 = 1$ ,  $a_{k-1} = n$  и для всех  $0 \leq i < k - a_{i+1}$  равно или  $a_i + 1$ ,  $2 \cdot a_i$ , или  $3 \cdot a_i$ :

```
input_txt = open("input.txt")
output_txt = open("output.txt", "w")
n = int(input_txt.readline())
a = [0] * (n + 1)
a[1] = 0
k = 0
def counting(n):
    g = []
    h = n
    while h > 0:
        if a[h] == a[h - 1] + 1:
            g.insert(0, h)
            h -= 1
            continue
        if h % 2 == 0 and a[h] == a[h // 2] + 1:
            g.insert(0, h)
            h //= 2
            continue
        g.insert(0, h)
        h //= 3
    return g
for i in range(2, n + 1):
    f = a[i - 1] + 1
    if i % 2 == 0:
        f = min(f, a[i // 2] + 1)
        k += 1
    if i % 3 == 0:
        f = min(f, a[i // 3] + 1)
        k += 1
    a[i] = f
print(a[n])
print(*counting(n))
```

Тесты:

```
19
1 3 9 27 54 108 216 217 651 1953 3906 7812 15624 15625 31250 62500 125000 250000 500000 1000000

Process finished with exit code 0
```

## Задача №3: Редакционное расстояние

В этом задании нужно было найти редакционное расстояние между двумя строками (редакционное расстояние между двумя строками – это минимальное количество операций (вставки, удаления и замены символов) для преобразования одной строки в другую). Каждая из двух строк ввода содержит строку, состоящую из строчных латинских букв. Длина обеих строк - от 1 до 5000:

```
def Levenstein(word, word1):
    k = len(word)
    m = len(word1)
    table = [[0] * (m + 1) for _ in range(k + 1)]
    for i in range(k + 1):
        table[i][0] = i
    for j in range(m + 1):
        table[0][j] = j

    for i in range(1, k + 1):
        for j in range(1, m + 1):
            if word[i - 1] == word1[j - 1]:
                table[i][j] = table[i - 1][j - 1]
            else:
                table[i][j] = 1 + min(table[i - 1][j], table[i][j - 1], table[i - 1][j - 1])
    return table[-1][-1]

print(Levenstein('short', 'ports'))
```

Тесты:

3

Process finished with exit code 0

## Задача №4: Наибольшая общая последовательность двух последовательностей

В четвёртом нужно было вычислить длину самой длинной общей последовательности из двух последовательностей. В первой строке входного файла прописана  $n$  - длина первой последовательности, во второй -  $a_1, a_2, \dots, a_n$  (элементы первой последовательности) через пробел, в третьей -  $m$  - длина второй последовательности, в четвёртой -  $b_1, b_2, \dots, b_m$  (элементы второй последовательности) через пробел:

```
input_txt = open("input.txt")
output_txt = open("output.txt", 'w')

n = int(input_txt.readline())
mass = list(map(int, input_txt.readline().split()))
m = int(input_txt.readline())
mass1 = list(map(int, input_txt.readline().split()))
ans = []
def common_counter(mass, mass1):
    if len(mass) == 0 or len(mass1) == 0:
        return []
    if mass[-1] == mass1[-1]:
        return common_counter(mass[:-1], mass1[:-1]) + [mass[-1]]
    else:
        left = common_counter(mass[:-1], mass1)
        right = common_counter(mass, mass1[:-1])
        if len(left) > len(right):
            return left
        return right
output_txt.write(str(len(common_counter(mass, mass1))))
```

Тесты:

input.txt		output.txt	
1	3	1	2
2	2 7 5		
3	2		
4	2 5		