

Санкт-Петербургский Национальный Исследовательский  
Университет Информационных Технологий, Механики и Оптики  
Мегафакультет компьютерных технологий и управления

Дисциплина  
«Алгоритмы и структуры данных»  
Лабораторная работа №3  
“Динамическое программирование”

Выполнил:  
Студент группы Р3218  
Рябов Сергей Витальевич  
Преподаватель:  
Муромцев Дмитрий Ильич

Санкт-Петербург,  
2018

## 1. Наибольшая последовательнократная подпоследовательность

Дано целое число  $1 \leq n \leq 10^3$  и массив  $A[1 \dots n]$  натуральных чисел, не превосходящих  $2 \cdot 10^9$ . Выведите максимальное  $1 \leq k \leq n$ , для которого найдётся подпоследовательность  $1 \leq i_1 < i_2 < \dots < i_k \leq n$  длины  $k$ , в которой каждый элемент делится на предыдущий (формально: для всех  $1 \leq j < k$ ,  $A[i_j] | A[i_{j+1}]$ ).

### Исходный код:

```
using System;
using System.Collections.Generic;
using System.IO;

namespace Stepik.DynamicProgramming
{
    public class SequenceTask
    {
        static void Main(string[] args)
        {
            int n = int.Parse(Console.ReadLine());
            string[] str = Console.ReadLine().Split(' ');
            int[] array = new int[n];
            for (int i = 0; i < n; i++)
                array[i] = int.Parse(str[i]);

            int[] sequences = new int[array.Length];
            for (int i = 0; i < sequences.Length; i++)
            {
                for (int j = 0; j < i; j++)
                {
                    if (array[i] % array[j] == 0)
                        sequences[i] = Math.Max(sequences[i], sequences[j] + 1);
                }
            }
            int max = 0;
            for (int i = 1; i < sequences.Length; i++)
                max = Math.Max(max, sequences[i]);
            Console.WriteLine(max + 1);
        }
    }
}
```

## 2. Расстояние редактирования

Вычислите расстояние редактирования двух данных непустых строк длины не более 100, содержащих строчные буквы латинского алфавита.

```
using System;
using System.Collections.Generic;
using System.IO;

namespace Stepik.DynamicProgramming
{
    public class RedactionDistanceTask
    {
        static void Main(string[] args)
        {
            string str1 = Console.ReadLine();
            string str2 = Console.ReadLine();

            int[,] table = new int[str2.Length + 1, str1.Length + 1];

            for (int i = 0; i <= str1.Length; i++)
                table[0, i] = i;
```

```

        for (int i = 0; i <= str2.Length; i++)
            table[i, 0] = i;

        for (int j = 1; j <= str2.Length; ++j) {
            for (int i = 1; i <= str1.Length; ++i) {
                int c = (str1[i - 1] == str2[j - 1]) ? 0 : 1;
                int min = Math.Min(Math.Min(table[j, i - 1] + 1, table[j - 1, i] + 1),
                                    table[j - 1, i - 1] + c);
                table[j, i] = min;
            }
        }
        Console.WriteLine(table[str2.Length, str1.Length]);
    }
}

```

### 3. Рюкзак

Первая строка входа содержит целые числа  $1 \leq W \leq 10^4$  и  $1 \leq n \leq 300$  — вместимость рюкзака и число золотых слитков. Следующая строка содержит  $n$  целых чисел  $0 \leq w_1, \dots, w_n \leq 10^5$ , задающих веса слитков. Найдите максимальный вес золота, который можно унести в рюкзак.

#### Исходный код:

```

using System;
using System.Collections.Generic;
using System.IO;

namespace Stepik.DynamicProgramming
{
    public class BackpackTask
    {
        static void Main(string[] args)
        {
            string[] str = Console.ReadLine().Split(' ');
            int w = int.Parse(str[0]);
            int n = int.Parse(str[1]);

            str = Console.ReadLine().Split(' ');
            int[] array = new int[n];
            for (int i = 0; i < n; i++)
                array[i] = int.Parse(str[i]);

            int[, ] weights = new int[w + 1, n + 1];
            for (int i = 1; i <= n; ++i)
            {
                for (int j = 1; j <= w; ++j)
                {
                    if (array[i - 1] > j)
                        weights[j, i] = weights[j, i - 1];
                    else
                        weights[j, i] = Math.Max(weights[j, i - 1], weights[j - array[i - 1], i - 1] +
array[i - 1]);
                }
            }
            Console.WriteLine(weights[w, n]);
        }
    }
}

```

## 4. Лестница

Даны число  $1 \leq n \leq 100$  ступенек лестницы и целые числа  $-10^4 \leq a_1, \dots, a_n \leq 10^4$ , которыми помечены ступеньки. Найдите максимальную сумму, которую можно получить, идя по лестнице снизу вверх (от нулевой до n-й ступеньки), каждый раз поднимаясь на одну или две ступеньки.

### Исходный код:

```
using System;
using System.Collections.Generic;
using System.IO;

namespace Stepik.DynamicProgramming
{
    public class StairsTask
    {
        static void Main(string[] args)
        {
            int n = int.Parse(Console.ReadLine());
            string[] str = Console.ReadLine().Split(' ');
            int[] array = new int[n];
            for (int i = 0; i < n; i++)
                array[i] = int.Parse(str[i]);

            if (n > 1)
                array[1] = Math.Max(array[1], array[0] + array[1]);
            if (n > 2)
                for (int i = 2; i < n; i++)
                    array[i] += Math.Max(array[i - 1], array[i - 2]);

            Console.WriteLine(array[n - 1]);
        }
    }
}
```

## 5. Калькулятор

У вас есть примитивный калькулятор, который умеет выполнять всего три операции с текущим числом  $x$ : заменить  $x$  на  $2x$ ,  $3x$  или  $x+1$ . По данному целому числу  $1 \leq n \leq 10^5$  определите минимальное число операций  $k$ , необходимое, чтобы получить  $n$  из 1. Выведите  $k$  и последовательность промежуточных чисел.

### Исходный код:

```
using System;
using System.Collections.Generic;
using System.IO;

namespace Stepik.DynamicProgramming
{
    public struct Element
    {
        public int Previous { get; }
        public int Length { get; }

        public Element(int previous, int length)
        {
            Previous = previous;
            Length = length;
        }
    }

    public class CalculatorTask
```

```

{
    static void Main(string[] args)
    {
        int n = int.Parse(Console.ReadLine());
        Element[] array = new Element[n + 1];

        for (int i = 2; i <= n; i++)
        {
            array[i] = new Element(i - 1, array[i - 1].Length + 1);
            if (i % 2 == 0 && array[i / 2].Length + 1 < array[i].Length)
                array[i] = new Element(i / 2, array[i / 2].Length + 1);
            if (i % 3 == 0 && array[i / 3].Length + 1 < array[i].Length)
                array[i] = new Element(i / 3, array[i / 3].Length + 1);
        }
        Console.WriteLine(array[n].Length);
        Stack<int> stack = new Stack<int>();
        while (n > 0)
        {
            stack.Push(n);
            n = array[n].Previous;
        }
        while (stack.Count > 0)
            Console.Write("{0} ", stack.Pop());
    }
}

```