# Рекурсия и итерация

```
Function QPower(b, n) :
    if n > 1 then
        x = QPower (b, ⌊n/2⌋)
        if n нечётно then
            return b × x × x
        end
        return x × x
    end
    return b
end
```

```
Function F(x, m) :
    a = SX(m);
    y = 1;
    for i = 1 to a.size do
        if a[i] == X then
            y = y × x
        else
            y = y × y
        end
    end
    return y
end
```

```
Function G(n) :
    if n > 1 then
        print(«алгоритм»)
        G (⌊n/2⌋)
        G (⌊n/2⌋)
    end
end
```

**1.** Найдите с помощью расширенного алгоритма Евклида обратный остаток $7^{-1} \pmod{102}$.

**2.** 1. Оцените временную сложность алгоритма QPower, вычисляющего $b^n$. Считайте, что арифметические операции стоят $O(1)$.

2. Вычислите $3^{11} \mod 107$.

**3.** Опишем преобразование $\texttt{SX}(m)$ числа $m$. Каждая единица двоичной записи числа $m$ заменяется на $SX$, а ноль на $S$, после чего вычёркивается первая слева $S$. Так $\text{bin}(5) = 101 \to XSSX$. Алгоритм вычисления функции F от положительных целых чисел $(x, m)$ задан псевдокодом.

1. Вычислите $\texttt{F}(3, 5)$.

2. Какую математическую функцию реализует данный алгоритм?

3. Докажите корректность данного алгоритма (нужно доказать, что алгоритм действительно реализует отгаданную функцию).

4. Оцените время работы алгоритма, считая, что арифметические операции стоят $O(1)$.

**4.** Найдите асимптотическую оценку функции $g(n)$, которая возвращает число слов «алгоритм», напечатанных при вызове $\texttt{G}(n)$.

**5** [ Задача о Ханойской башне ]. Есть три стержня, на первом из них нанизано $n$ колец разного радиуса; чем ниже лежит кольцо, тем больше радиус. Кольца разрешено перекладывать со стержня на стержень, но только при условии что кольцо меньшего радиуса кладётся на кольцо большего радиуса. Найдите минимальное число перекладываний, требуемое для того, чтобы переложить все кольца с одного стержня на другой. (Найдите как верхнюю, так и нижнюю оценку).

**6** [ Шень **1.1.33** ]. Функция $f$ с натуральными аргументами и значениями определена так: $f(0) = 0$, $f(1) = 1$, $f(2n) = f(n)$, $f(2n+1) = f(n) + f(n+1)$.

1. Построить рекурсивный и итеративный алгоритмы, вычисляющий $f(n)$ и оценить их сложности, считая что арифметические операции стоят $O(1)$.

2. Постройте алгоритм, требующий $O(\log n)$ операций.

**7*.** Предложите полиномиальный от длины входа алгоритм нахождения минимального решения системы сравнений:

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \dots \\ x \equiv a_k \pmod{m_k} \end{cases}$$

где все $m_i$ — попарно взаимопростые положительные целые числа. На вход задачи подаётся число $k$ и система из $k$ сравнений.