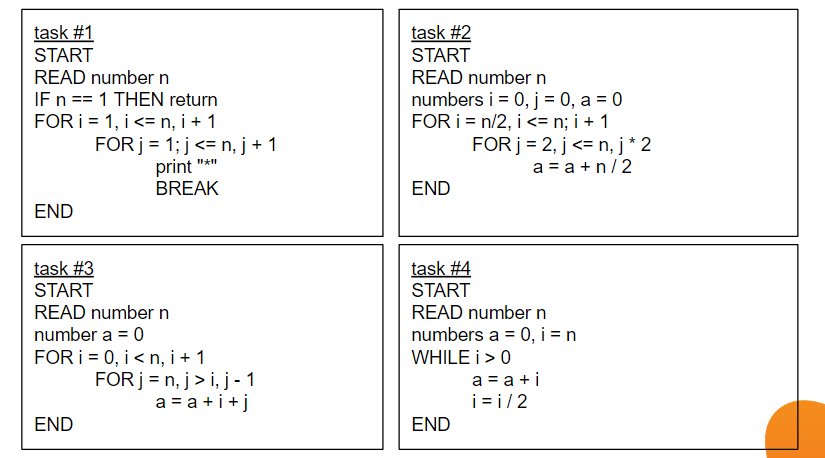
2. Познакомиться с [Задачей коммивояжёра](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BC%D0%B8%D0%B2%D0%BE%D1%8F%D0%B6%D1%91%D1%80%D0%B0). Объясните своими словами, какова ее сложность?

Задача коммивояжёра — это классическая задача в теории графов и оптимизации. В общем виде задача формулируется так: у нас есть набор городов, и нужно найти самый короткий маршрут, который позволяет посетить все города ровно один раз и вернуться в исходный город.

Проблема в том, что количество возможных маршрутов растет очень быстро с увеличением числа городов. Например, для 4 городов у нас есть 6 возможных маршрутов, для 5 — уже 120, а для 10 городов — более 3,6 миллиона. Это делает задачу коммивояжёра чрезвычайно трудной для решения при большом количестве городов.

Сложность этой задачи объясняется тем, что она относится к классу NP-полных задач. Это означает, что даже если мы находимся в ситуации, когда можем проверять, является ли маршрут оптимальным, нахождение самого оптимального маршрута требует экспоненциального времени. В общем случае нет известного алгоритма, который бы решал задачу коммивояжёра за разумное время для больших наборов данных.

3. Решить практические задачи 1-4: какова временная сложность?



Task 1

1. Чтение числа n O(1)
2. Проверка условия if n == 1 O(1)
3. Внутренний цикл всегда выполняется только один раз, так как команда Break немедленно прерывает его после первой итерации O(1)
4. внешний цикл выполняется nnn раз, общая временная сложность для вложенных циклов составляет O(n)
5. Итоговая временная сложность кода: O(n)

Task 2

1. Чтение числа n O(1).
2. Инициализация переменных i, j, a O(1)
3. Внешний цикл For i = n/2, i <= n, i + 1:

количество итераций внешнего цикла: от n/2 до n (включительно) O(n)

внутренний цикл For j = 2, j <= n, j \* 2:

значение j удваивается и растет экспоненциально O(log n)

1. Операция внутри внутреннего цикла a = a + n/2

простое присваивание O(1)

Итоговая временная сложность данного кода O(nlog⁡ n)

Task 3

1. Чтение числа n O(1)
2. Инициализация переменной a O(1)
3. Внешний цикл For i = 0, i < n, i + 1 O(n)
4. Внутренний цикл For j = n, j > i, j - 1

начинается с j = n и уменьшается на 1 в каждой итерации до тех пор, пока j > i

количество итераций внутреннего цикла зависит от текущего значения i в внешнем цикле. Для данного значения i внутренний цикл выполняется n−i раз

суммарное количество итераций внутренних циклов по всем итерациям внешнего цикла можно выразить как арифметическую прогрессию

Итоговая временная сложность кода O(n\*\*2)

Task 4

1. Чтение числа n O(1)
2. Инициализация переменных a и i O(1)
3. Цикл While i > 0 log2​(n)
4. Операции внутри цикла:

операция a = a + i O(1)

операция i = i / 2 O(1)

Итоговая временная сложность данного кода O(log⁡ n)