**Теория (10 баллов).**

**Ответьте на следующие вопросы:**

1. По шагам опишите ход решения типичной задачи динамического программирования.

**Мой план решения задач методом Дин. Программирования**

1. Действительно ли задача на дин. Программирование?

* Можно ли уменьшить задачу?
* Помогает ли решение меньшей задачи получить ответ на большую?

1. Вывод рек. Формулы получения ответа на задачу через решение подзадач.
2. Подбор базы. (Начальных значений, чтобы рек. работала).
3. Начало кодирования.
4. Нужно понять, куда сохранять решения на подзадачи. Как их использовать при решении задачи.
5. Где лежит ответ и как его выдать?

2. На простом примере покажите отличие восходящего метода динамического программирования от рекурсии с запоминанием. Какие недостатки/преимущества у каждого метода?

*Рекурсия с запоминанием* – рекурсивно решает поставленную задачу, сохраняя результаты вычислений элементов, которые будут использоваться для решения больших задач.

По сути своей – рекурсия с запоминанием, это динамическое программирование с подходом сверху вниз (сначала решаются более большие задачи, а затем более маленькие.)

''' Рекурсия с запоминанием '''

M = {0: 0, 1: 1} # Сразу храним результаты для первых чисел

def fib2(n):

    if n in M:

        return M[n]

    M[n] = fib2(n - 1) + fib2(n - 2)

    return M[n]

**Преимущества:**

* Выполняется за линейное время.

**Недостатки:**

* Тратит больше памяти для сохранения результатов.

*Динамическое программирование* – более широкое понятие, чем рекурсия с запоминанием. В нём не всегда используется рекурсия. Оно используется , когда:

1) Подзадачи повторяются (имеет смысл сохранять результаты)

2) Подзадачи немного проще задачи

''' Динамическое программирование '''

def fib4(n):

    a = 0

    b = 1

    for \_ in range(n):

        a, b = b, a + b

    return a

print(fib4(10))

**Преимущества:**

* Выполняется быстрее, чем рекурсия с запоминанием, но всё ещё за линейное время.
* Фиксированное использование памяти (т.к. для ответа на поставленный вопрос нужно знать только 2 предшествующих числа, а не все.)
* Отсутствуют рекурсивные вызовы функций.

**Недостатки:**

* По-прежнему линейная скорость выполнения.

3. Вася получил задачу - вычислить последнюю цифру n-го числа Фибоначчи. Может ли Вася решить ее динамическим программированием? Ответ обоснуйте.

Вычислить n-ое число Фибоначчи можно, используя динамическое программирование (см. пример выше). Чтобы получить последнюю цифру этого числа достаточно взять остаток от деления на 10.