**Практическая работа №5**

**Рекурсия, рекурсивные алгоритмы**

1. Рекурсия -это определение объекта посредством ссылки на себя.

2. Примеры:



Зеркало отражает само себя



Матрешка повторяет фигуру, большую по размерам



Картина повторяется многократно



Мужчина создает свой портрет, в котором он повторяется

3. Рекурсивный алгоритм – это алгоритм, в определении которого содержится прямой или косвенный вызов этого же алгоритма.

4. Рекурсивная триада включает в себя три основных этапа решения задач с помощью рекурсии: параметризация, выделение базы (или выделение начальной базы и правил её изменения), декомпозиция.

5. Полное дерево – это оценка трудоемкости рекурсивных алгоритмов.

Глубина рекурсии – это наибольшее количество рекурсивных обращений функции, определяющее максимальное количество слоев рекурсивного стека, в котором осуществляется хранение отложенных вычислений.

Объем рекурсии – количество вершин полного рекурсивного дерева без единицы.

6. Область памяти, выделяемая для хранения всех промежуточных значений локальных переменных – это рекурсивный стек. Каждый раз, когда вызывается рекурсивная функция, генерируется и сохраняется в памяти новая запись активации. Эти записи активации хранятся в специальном стеке, называемом рекурсивным стеком. Эти записи активации удаляются после завершения выполнения функции.

7. Пример рекурсивной процедуры:

def factorial(n):

if n == 0 or n == 1: # базовый случай

return 1

else:

return n \* factorial(n - 1) # рекурсивный случай

**Рекурсивная триада:**

Базовый случай: в примере базовый случай наступает при n=0n = 0n=0 или n=1n = 1n=1, когда возвращается 1.

Рекурсивный случай: В примере это выражение n×factorial(n−1)n \times factorial(n - 1)n×factorial(n−1).

Прогрессивное приближение к базовому случаю: Каждый рекурсивный вызов уменьшает аргумент nnn, приближаясь к базовому случаю.

**Полное дерево рекурсии:**

factorial(4)

/ \

factorial(3) 4 \*

/ \

factorial(2) 3 \*

/ \

factorial(1) 2 \*

/ \

1. 1

**Глубина дерева**

глубина рекурсивного дерева равна 4

**Объем рекурсии**

Объем рекурсии - O(h), глубина это h и она равна 4. Поэтому объем рекурсии будет O(4).

8. Эта картинка демонстрирует рекурсию – концепцию в программировании и математике, где функция вызывает саму себя.

В данном контексте, юмор основан на бесконечной рекурсии: человек на собеседовании не может рассказать о себе, кроме повторяющихся шуток про рекурсию.

