**Практическая работа №5**

**Рекурсия, рекурсивные алгоритмы**

1. Изучите материал в презентации.
2. Дайте определение рекурсии (в широком смысле).
3. Приведите примеры рекурсии в жизни и опишите их (4-5 примеров, можно фото).
4. Дайте определение рекурсивного алгоритма.
5. Что такое рекурсивная триада?
6. Что такое полное дерево, глубина и объем рекурсии?
7. Как называется область памяти, выделяемая для хранения всех промежуточных значений локальных переменных? Опишите ее.
8. Приведите пример рекурсивной процедуры/функции. Постройте полное дерево рекурсии, определите глубину и объем рекурсии. Укажите рекурсивную триаду.
9. Приведите примеры мемчиков на рекурсии. Опишите их (желательно сделать свой).

2. Рекурсия (в широком смысле) - это определение объекта посредством ссылки на себя.

3.

1) Деревья и растения

Рекурсивная структура наблюдается в природе, например, в росте деревьев. Главный ствол дерева делится на ветви, которые затем делятся на меньшие ветки, и так далее. Каждая ветка — это аналогия рекурсивного вызова, который ведёт к новым ветвям.

2) Семейные древа

В генеалогии каждое новое поколение может быть представлено как уровень рекурсии. У родителя есть дети, каждый из которых также может стать родителем, создавая более сложные связи. Это позволяет представить родственные связи в виде рекурсивной структуры.

3) Поиск по интернету

Алгоритмы поисковых систем используют рекурсию для индексации и поиска информации. Когда пользователь отправляет запрос, система может рассмотреть множество веб-страниц, каждая из которых может ссылаться на другие страницы, создавая рекурсивные связи.

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

5. Рекурсивная триада — это концепция, которая описывает три уровня или аспекта, которые повторяются в процессе рекурсии. Обычно это включает:

1. Базовый случай: Условие, при котором рекурсия останавливается, чтобы избежать бесконечного цикла.

2. Рекурсивный случай: Условие, при котором функция вызывает саму себя с изменёнными параметрами, чтобы продвигаться к базовому случаю.

3. Объединение результатов: После достижения базового случая, результаты рекурсивных вызовов объединяются для получения окончательного результата.

6. Для оценки трудоемкости рекурсивных алгоритмов строится полное дерево рекурсии.

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

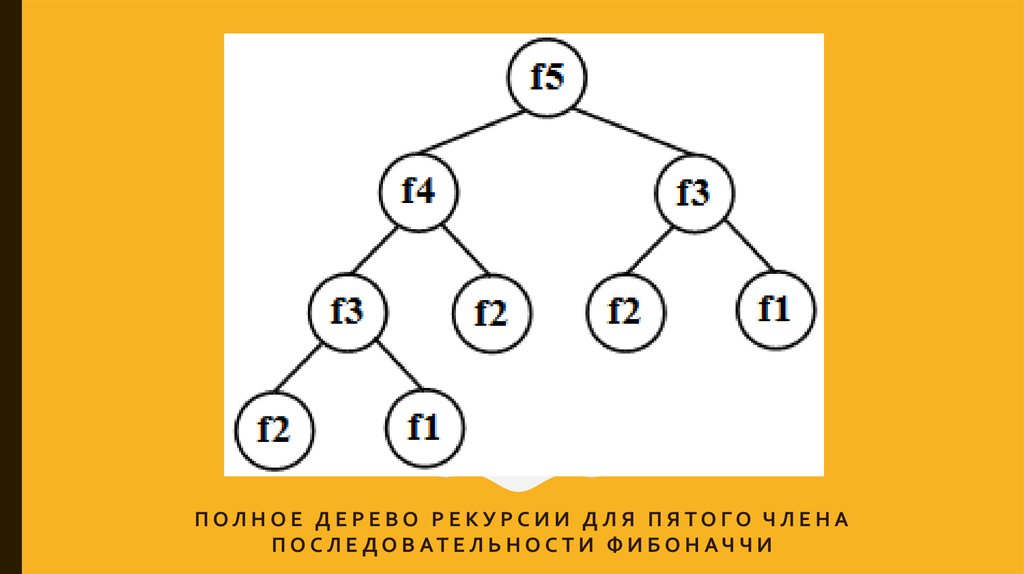
Объем рекурсии - количество вершин полного рекурсивного дерева без единицы

7. Рекурсивный стек

Для каждого текущего обращения формируется локальный слой данных стека (при этом совпадающие идентификаторы разных слоев стека независимы друг от друга и не отождествляются). ¬Завершение вычислений происходит посредством восстановления значений данных каждого слоя в порядке, обратном рекурсивным обращениям.

8. n! = n \* (n - 1)! для n > 0

0! = 1 (базовый случай)



Глубина рекурсии: 4

Объём рекурсии: 9

9.



