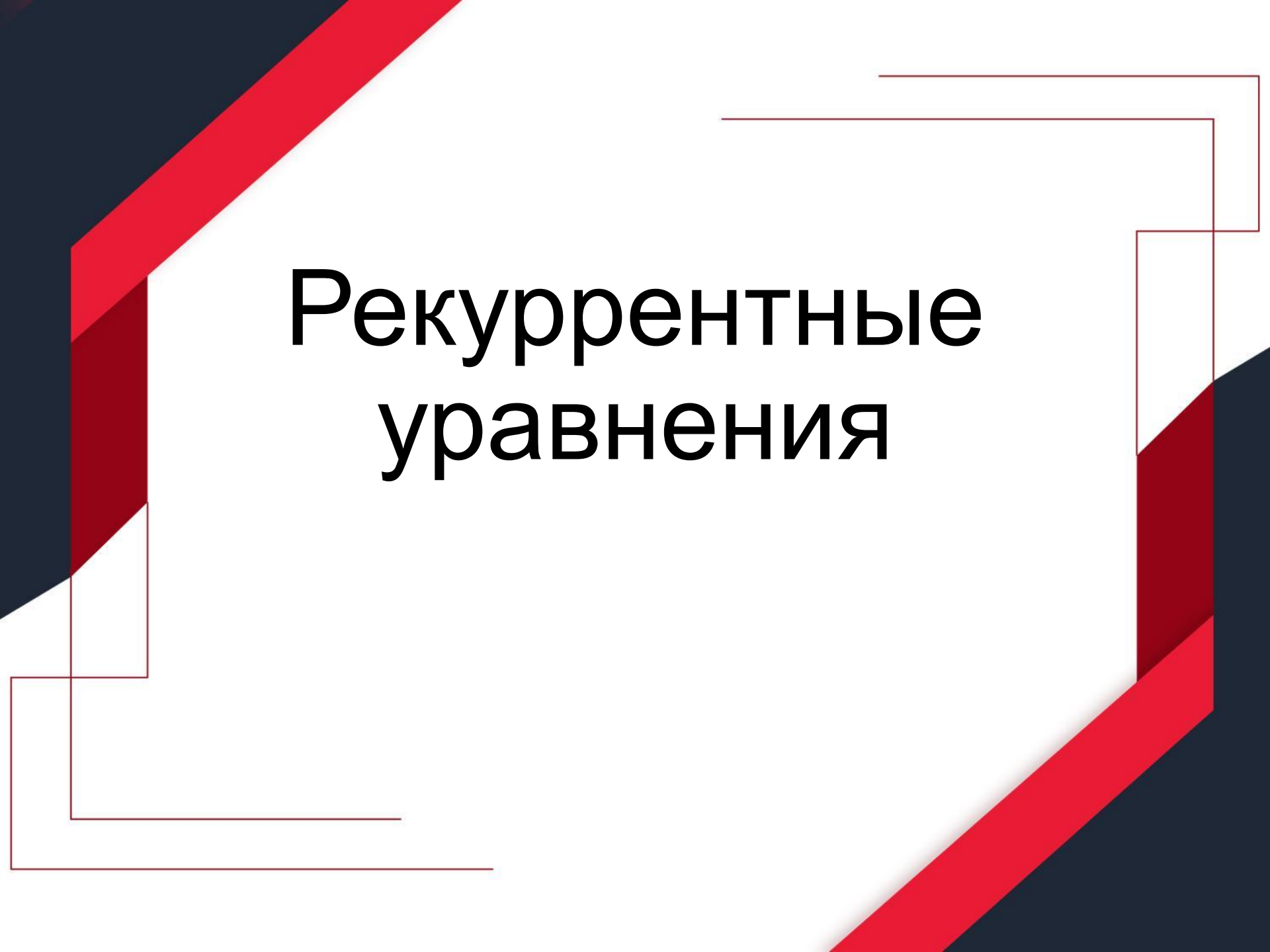


# Рекуррентные уравнения

The image features a white background with decorative elements. On the left and bottom-right corners, there are overlapping geometric shapes in red and dark blue. A thin, dark red line forms a stepped, L-shaped border around the central text area.

# Понятие

Для некоторой задачи под **подзадачей** мы будем понимать:

- ту же задачу, но меньшим числом параметров,

или

- задачу с тем же числом параметров, но при этом хотя бы один из параметров имеет меньшее значение.

# Рекуррентное уравнение

соотношения, связывающие одни и те же функции, но с различными аргументами, называются рекуррентными уравнениями.

# Правильное рекуррентное уравнение

*Рекуррентное уравнение* называется **правильным** если значения аргументов у любой из функций правой части соотношения меньше значения аргументов любой из функций левой части соотношения; если аргументов несколько, то достаточно уменьшение одного из них.

*Правильное рекуррентное уравнение* называется **полным**, если оно определено для всех допустимых значений аргументов.

# Примеры

1.  $T(n) = T(n - 1) + T(n - 2)$
2.  $T(n) = F(n - 2) + T(n - 1)$
3.  $T(n) = T(n - 1) + T(n + 1)$
4.  $T(i) = T(i-1) + a_i, i > 1; T(1) = a_1$

# Примеры рекуррентных уравнений

1. Нахождение максимального элемента из  $n$  элементов массива

$$T(n) = T(n-1) + C = (T(n-2) + C) + C = \dots = \\ = C \cdot (n-1), T(1) = 0, \text{ где } C = O(1)$$

2. Нахождение наибольшего и наименьшего из  $n$  элементов массива

$$T(n) = 1, \text{ при } n = 2;$$

$$T(n) = 2T(n/2) + 2, \text{ при } n > 2.$$

# Решение рекуррентных уравнений

1. Метод итераций
2. Подстановочный метод
3. Метод рекурсивных деревьев

# Метод итераций

Метод заключается в том, что данное рекуррентное уравнение расписывается через множество других и затем происходит суммирование полученного выражения.



# Пример 1

Найти методом итераций решение для рекуррентного уравнения:

$$T(n) = 2T(n/2) + 5n^2$$

$$T(1) = 7$$

# Решение

Для простоты мы предположим, что  $n$  является степенью 2, т.е.  $n = 2^k$

$$\begin{aligned}T(n) &= 2T(n/2) + 5n^2 = \\&= 2(2T(n/4) + 5(n/2)^2) + 5n^2 = \\&= 2(2(2T(n/8) + 5(n/4)^2) + 5(n/2)^2) + 5n^2 = \dots = \\&= 2^k T(1) + 2^{k-1} \cdot 5(n/2^{k-1})^2 + \dots + 2 \cdot 5(n/2)^2 + 5n^2\end{aligned}$$

# Решение

$$\dot{T}(n) = 7n + 5 \sum_{i=0}^{k-1} \frac{n^2}{2^i} = 7n + 5n^2 \sum_{i=0}^{k-1} \frac{1}{2^i} =$$

$$= 7n + 5n^2 \left(2 - \frac{1}{2^{k-1}}\right)$$

$$T(n) = 7n + 5n^2 \left(2 - \frac{1}{n}\right) = 10n^2 - 3n$$

$$T(n) = O(n^2)$$

# Пример 2

Найти методом итераций решение для рекуррентного уравнения:

$$T(n) = aT(n-1) + bn,$$
$$0 < a < 1, n > 1, T(1) = 0$$

# Решение

$$\begin{aligned} \bullet T(n) &= aT(n-1) + bn = \\ &= a(aT(n-2) + b(n-1)) + bn = \\ &= bn + ab(n-1) + a^2T(n-2) = \\ &= bn + ab(n-1) + a^2b(n-2) + a^3T(n-3) = \dots = \\ &= b \sum_{i=0}^{n-2} a^i (n-i) \leq \sum_{i=0}^{n-2} a^i n \leq \frac{bn}{1-a} \end{aligned}$$

$$T(n) = O(n)$$

# Подстановочный метод

Метод заключается в том, что методом подбора находится такая функция  $g(n)$ , при подстановке которой в рекуррентное уравнение вместо  $T(n)$  получается верное неравенство в котором

**$g(n)$  = Левая часть  $\geq$  правой части**

**!** Функция должна быть наименьшего возможного порядка.

# Пример

Предположим, что необходимо решить следующее рекуррентное уравнение

$$T(n) = 2T(n/2) + n$$

$$g(n) \geq 2g(n/2) + n$$

1.  $g(n) = cn$
2.  $g(n) = cn^2$
3.  $g(n) = cn \log_2 n$

# Пример

Предположим, что необходимо решить следующее рекуррентное уравнение

$$T(n) = 2T(n/2) + b$$

$$g(n) \geq 2g(n/2) + b$$

1.  $g(n) = cn$
2.  $g(n) = cn^2$
3.  $g(n) = c_1n + c_2$



# Метод рекурсивных деревьев

1. На первой итерации формируется дерево следующего вида:
  - в корень дерева заносится свободный член исходного рекуррентного уравнения,
  - сыновьями этого корня являются рекуррентные функции правой части исходного соотношения.
2. На последующих итерациях для каждого из сыновей строится аналогичная древовидная структура.

# Метод рекурсивных деревьев

3. Процесс построения древовидной структуры заканчивается, когда все значения висячих вершин равны  $T(1)$

- при этом значения внутренних вершин дерева есть некоторые явные функции (не рекуррентные) от размера задачи;

- висячие вершины построенной древовидной структуры не обязательно одинаково удалены от корня.

# Метод рекурсивных деревьев

4. После построения дерева суммирование значений в вершинах производится следующим образом:

1. Определяются суммы значений для равноудаленных от корня вершин (эти вершины находятся на одном уровне),

2. Находится максимальная сумма по уровням. 3. Общая трудоемкость алгоритма ограничена сверху одним из следующих значений:

a) максимальной суммой, умноженной на количество уровней,

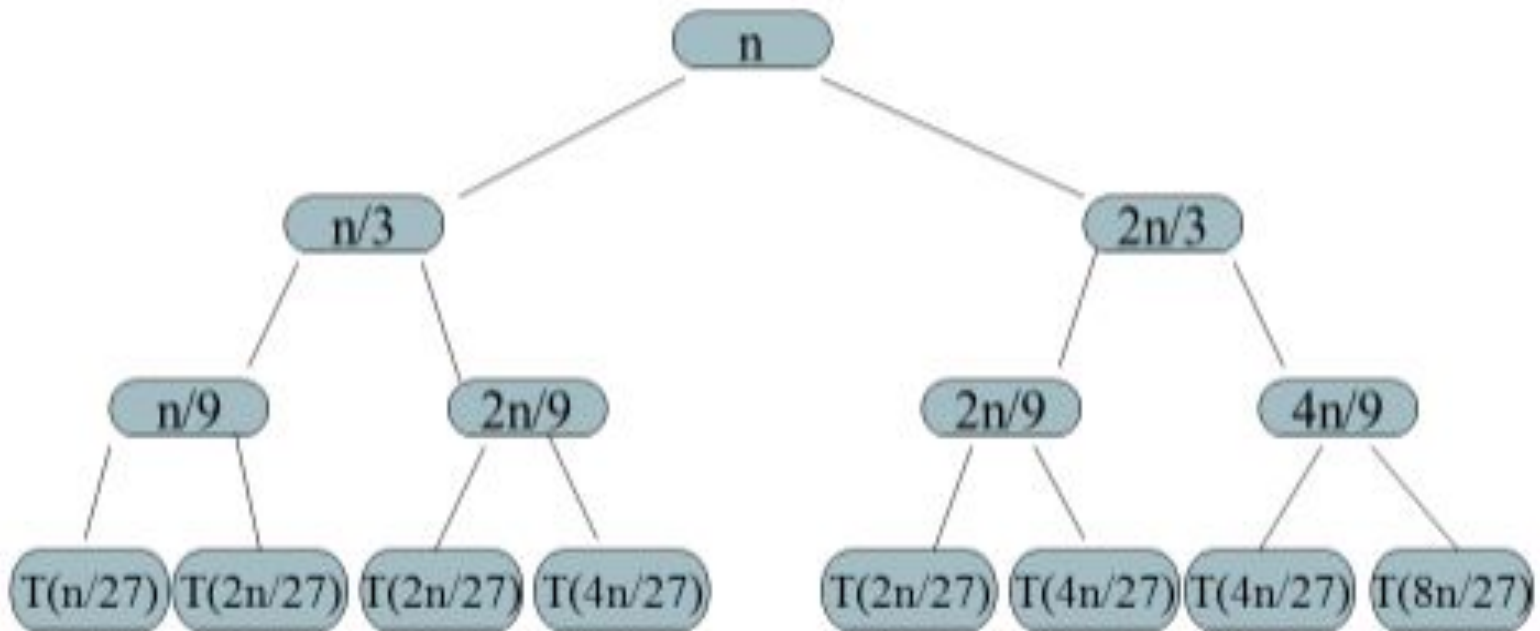
b) суммой, полученной в результате суммирования сумм значений по уровням

# Пример

Решить следующее рекуррентное уравнение

$$T(n) = T(n/3) + T(2n/3) + n$$

# Решение



# Решение

- сумма на каждом уровне равна  $n$
- количество уровней равно  $\log_{3/2} n$

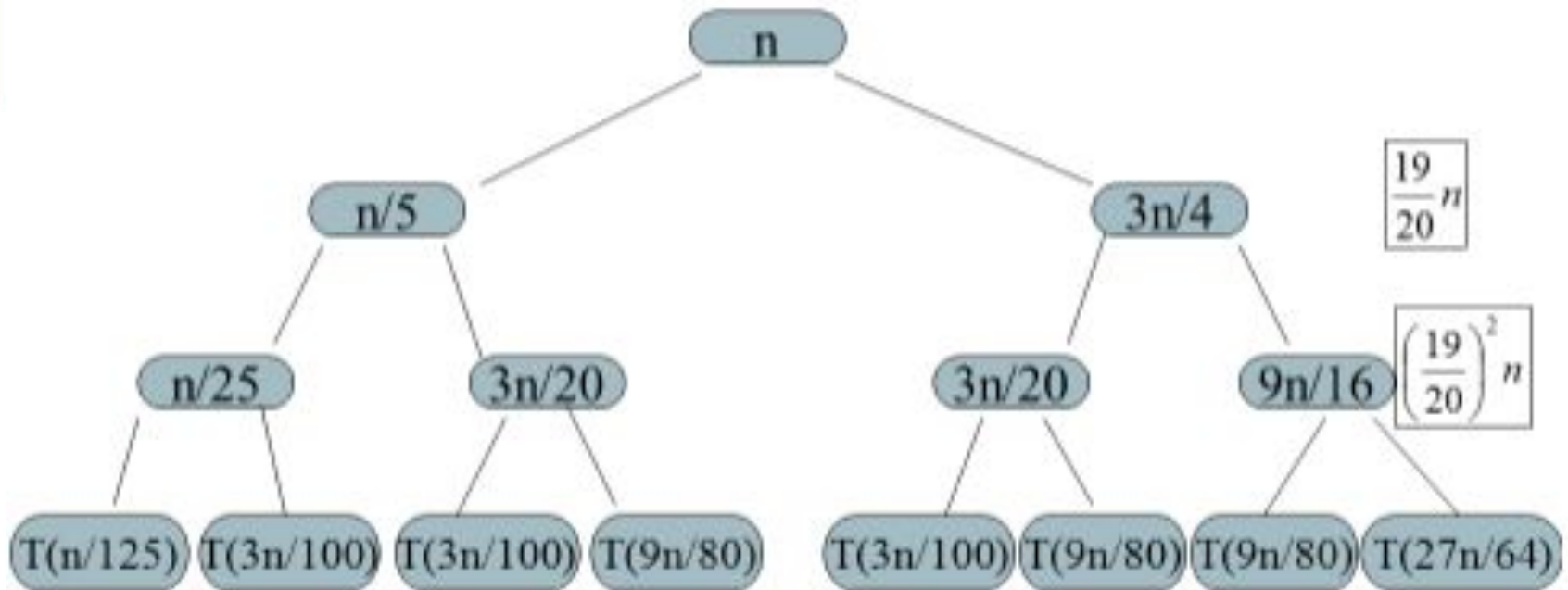
$$T(n) = cn \log_2 n$$

# Пример

Решить следующее рекуррентное уравнение

$$T(n) = T(n/5) + T(3n/4) + n$$

# Решение





# Решение

$$\begin{aligned} T(n) &= T(n/5) + T(3n/4) + n \leq \\ &\leq n(1 + q + q^2 + q^3 + \dots + q^{\log_{4/3} n}) \leq 20n \end{aligned}$$

где  $q = 19/20$ .

$$T(n) = O(n)$$

# Основная теорема

**Теорема.** Пусть  $a, b, c, k$  - некоторые константы, причем  $a \geq 1, b > 1$ . Тогда решение рекуррентного уравнения

$$T(n) = aT(n/b) + cn^k ; T(1) = c$$

имеет вид:

$$T(n) = \Theta(n^{\log_b a}), \text{ если } a > b^k$$

$$T(n) = \Theta(n^k \log_2 n), \text{ если } a = b^k$$

$$T(n) = \Theta(n^k), \text{ если } a < b^k$$