$7^{13} \mod 167$

$13_2 = 1101$

$7^2 \cdot 7 \equiv 343 \equiv 9 \mod 167$

$9^2 \equiv 81 \mod 167$

$81^2 \cdot 7 \equiv 81 \cdot 567 \equiv 2 \mod 167$

МОДУЛЬНАЯ АРИФМ.

$\mathbb{Z}$

$\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$

$1 + 2 = 3$

$3 + 4 \equiv 2 \mod 5$

$+ \qquad *$

$3 + 4 \equiv -3 \mod 5$

$$\begin{pmatrix} F_{k+1} \\ F_k \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}}_{A} \begin{pmatrix} F_k \\ F_{k-1} \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

$$\begin{pmatrix} F_k \\ F_{k-1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} F_{k+1} \\ F_k \end{pmatrix}$$

Основная теорема о рекуррентных соотношениях

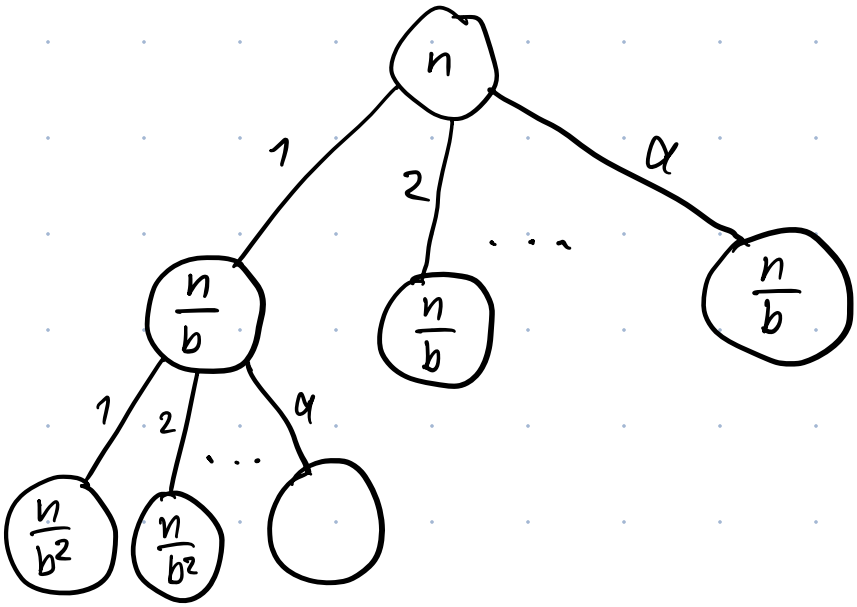
$$T(n) = \alpha T\left(\frac{n}{b}\right) + f(n)$$

α - const; $\mathbb{N}$

$b > 1$

$f: \mathbb{N} \to \mathbb{R}_+$

$c = \log_b \alpha$



2. $f(n) = \Theta(n^c)$

$c = \log_2 2 = 1$

$n = \Theta(n^1)$

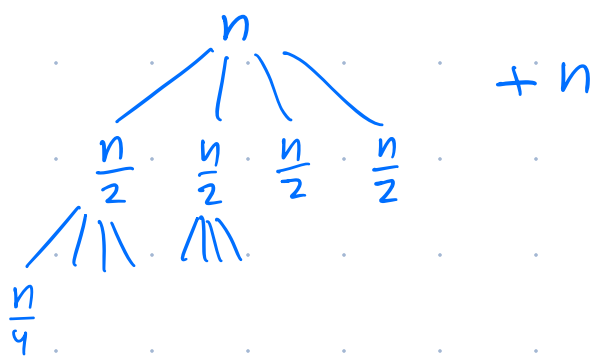
$T(n) = \Theta(n^c \log n)$

$T(n) = \Theta(n \log n)$

$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + cn$$

1. $f(n) = O(n^{c-\varepsilon})$; $\varepsilon > 0$

$T(n) = 4T\left(\frac{n}{2}\right) + n$



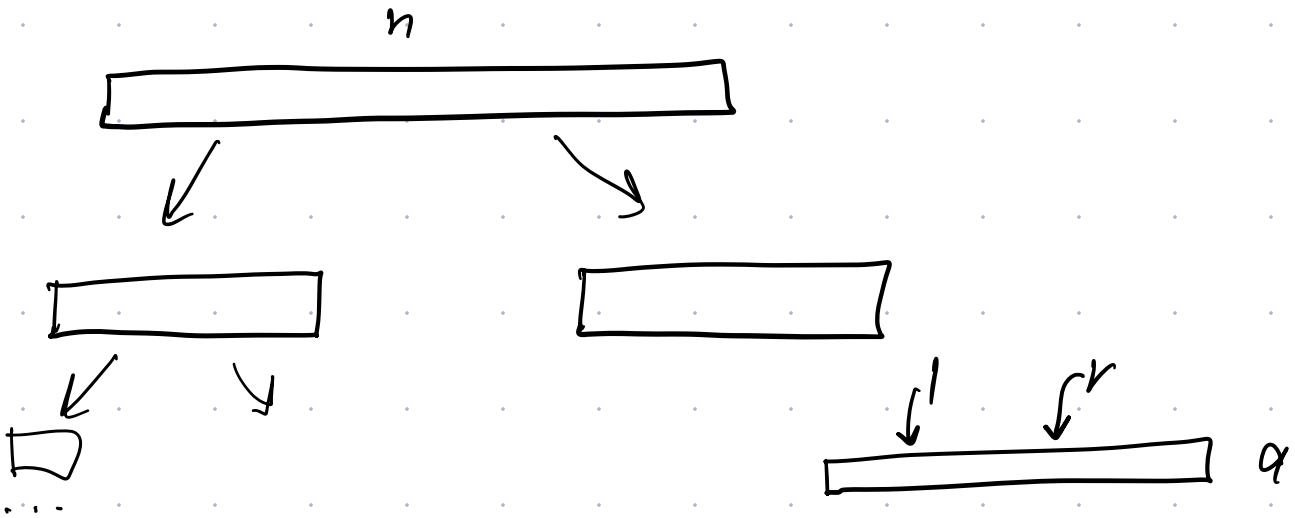
$c = \log_2 4 = 2$

$n = O(n^{2-\frac{1}{3}})$

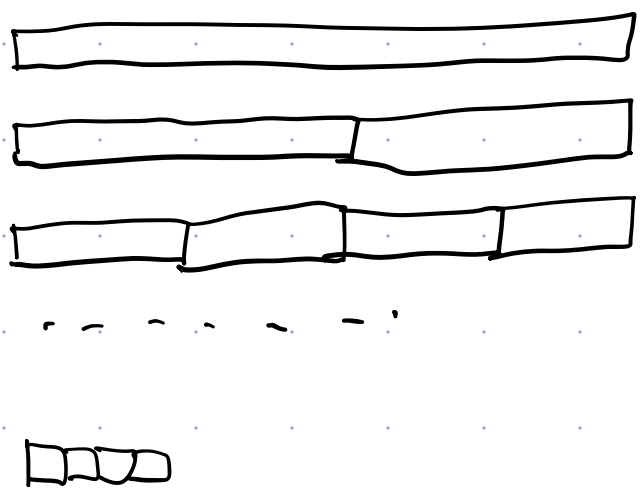
$T(n) = \Theta(n^{\log_b \alpha})$; $T(n) = \Theta(n^2)$

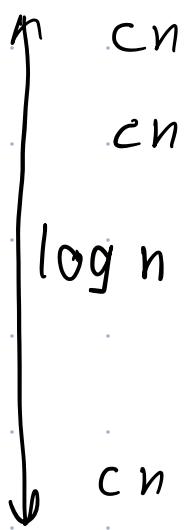
3. $f(n) = \Omega(n^{c+\varepsilon})$; $\varepsilon > 0$

Сортировка слиянием (merge sort)

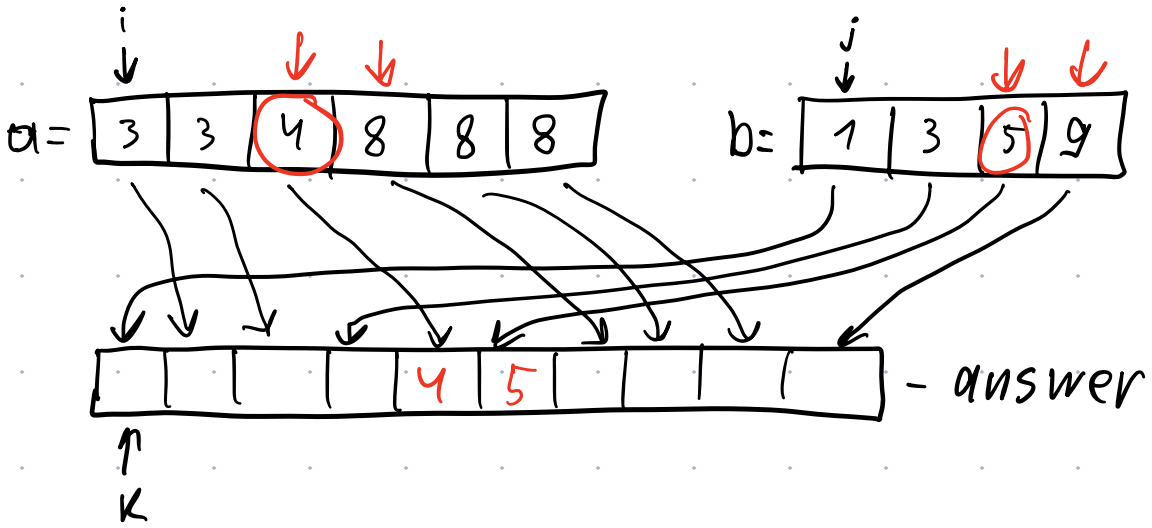



```
def merge_sort(a, l, r):                    ] merge(n)
    if (r-l ≤ 1):                           ]
        return                              ] O(1)
    m = ⌊(l+r)/2⌋                           ]
    merge_sort(a, l, m)                     ] merge(n/2)
    merge_sort(a, m, r)                     ] merge(n/2)
    merge(a[l:m], a[m:r])                   ] O(n)
```





$\Theta(n \log n)$




```
def merge(a, b):                                    Θ(|a| + |b|)
    i, j = 1, 1

    for k in range(len(a)+len(b)):
        if (i == len(a) + 1):
            answer[k] = b[j]
            j += 1
        elif (j == len(b) + 1):
            answer[k] = a[i]
            i += 1
        else:
            if (a[i] > b[j]):
                answer[k] = b[j]
                j += 1
            else:                                   # a[i] <= b[j]
                answer[k] = a[i]
                i += 1

    return answer
```

3. $f(n) = \Omega(n^{c+\varepsilon})$; $\varepsilon > 0$

УСЛОВИЕ РЕГУЛЯРНОСТИ: $af\left(\frac{n}{b}\right) < k \cdot f(n)$

$k < 1$

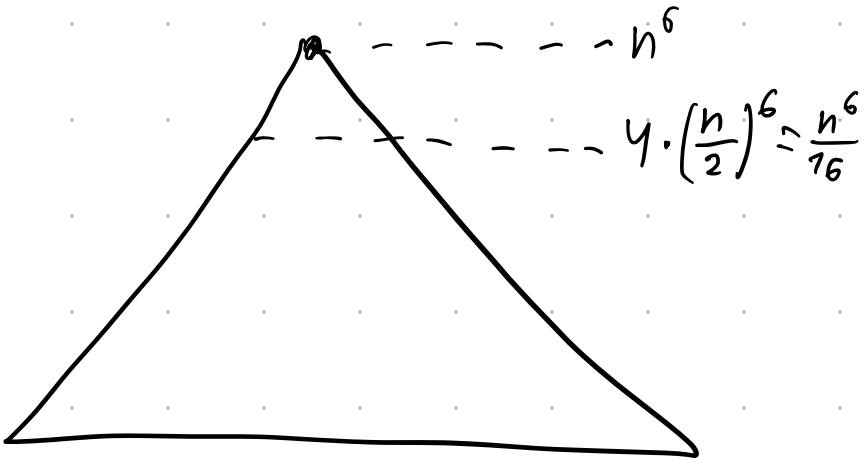
$4 \cdot \left(\frac{n}{2}\right)^6 = \frac{n^6}{16} < \left(\frac{1}{2}\right) n^6$ (L1)

$a \cdot f\left(\frac{n}{b}\right)$ — $f(n)$

$T(n) = 4T\left(\frac{n}{2}\right) + n^6$

$T(n) = \Theta(n^6)$

$T(n) = \Theta(f(n))$



МАСТЕР-ТЕОРЕМА

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$

$a \in \mathbb{N}$

$b \in \mathbb{R}_{>1}$

$f: \mathbb{N} \to \mathbb{R}_+$

$c = \log_b a$

1. $\exists\, \varepsilon > 0 : \; f(n) = O(n^{c-\varepsilon})$

   $T(n) = \Theta(n^{\log_b a})$

2. $f(n) = \Theta(n^c)$

   $T(n) = \Theta(n^c \log n)$

3. $f(n) = \Omega(n^{c+\varepsilon})$

   $\exists\, \varepsilon > 0 :$

   $\exists\, k < 1 : \; af\left(\frac{n}{b}\right) < k f(n)$

   $T(n) = \Theta(f(n))$

НЕПРИМЕНИМА:

$T(n) = \sqrt{n}\, T\left(\frac{n}{3}\right) + \dots$

$T(n) = T(n-5) + T\left(\frac{n}{2}\right) + \dots$

1. $f(n) = O(n^{\log_b a - \varepsilon})$

$T(n) = 8T(\frac{n}{2}) + \frac{n^3}{\log n}$

$f(n) = O(n^{3-\varepsilon})$

1. $f(n) = O(n^3)$
2. $\nexists\ \varepsilon > 0 : f(n) = O(n^{3-\varepsilon})$

$f(n) \neq \Theta(n^3)$

$T(n) = 26T(\frac{n}{3}) + 5n^3$

3 случай

$f(n) = \Omega(n^{\log_3 26 + 0{,}0001})$

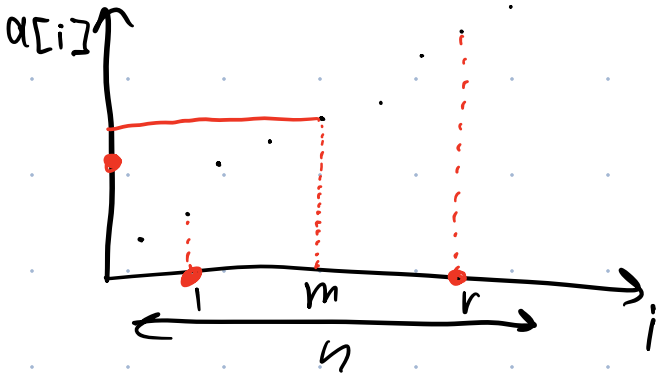
усл. рег.: $26(\frac{n}{3})^3 = \frac{26}{27}n^3 < \frac{53}{54}n^3$

$T(n) = \Theta(n^3)$

$f(n) \neq O(n^{\log_3 26 - \varepsilon})$

1. f мал.
2. f и рек. экв.
3. f бол.

Бинарный поиск.




```
def bin_search(array, l, r, x):
    if (r == l):
        if (array[l] == x):
            return l
        else:
            return None

    m = ⌊(r+l)/2⌋

    if (array[m] == x):
        return m
    elif (array[m] > x):
        return bin_search(array, l, m)
    else:
        return bin_search(array, m+1, r)
```

$T(n) = 1 \cdot T(\frac{n}{2}) + \text{же}$

$c = \log_2 1 = 0$

$\text{же} = \text{же} \cdot n^0$

$f(n) = \Theta(n^0)$

$\text{же} = \Theta(1)$

$T(n) = \Theta(n^0 \log n)$

```
def f(n):
    if (n < 9000):
        for i in range(n):
            print("AAA")
    else:
        f(n/3)
        for i in range(9000):
            print("AAA")
        f(n/3)
```

$$T(n) = 2T\left(\frac{n}{3}\right) + 9000$$

$c = \log_3 2$

$f(n) = O(n^{c-0,001})$

$T(n) = \Theta(n^{\log_3 2})$

Akra-Bazzi

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{n}{2}\right) + 9001$$

$< 2T\frac{n}{2} + 9001$

$> 2T\left(\frac{n}{3}\right)$

c