ЗАНЯТИЕ 2. 13.01.25

ДЗ (ТЕОР)

№1. 1. $f(n) = O(n^2)$

$g(n) = \Omega(1)$

$g(n) = O(n)$

$$h(n) = \frac{f(n)}{g(n)}$$

1) $h(n) \overset{\checkmark}{=} \Theta(n \log n)$????

$\left.\begin{array}{l} f(n) = n \log n \\ g(n) = 1 \end{array}\right]$ ✓

$f(n) = n^2 \log n \neq O(n^2)$] ✗

$\left.\begin{array}{l} f(n) = n \log^2 n \\ g(n) = \log n \end{array}\right]$ ✓

$\left.\begin{array}{l} f(n) = n^2 \\ g(n) = \frac{n}{\log n} \end{array}\right]$ ✓

2) $h(n) = \Theta(n^3)$

$\underbrace{C_1 n^3 \le h(n)} \le C_2 n^3$) из опр. Θ

$$h(n) = \frac{f(n)}{g(n)}$$

$\exists N_f, \exists C_f$

$$h(n)\, g(n) = f(n) \le C_f n^2$$

$$g(n) \ge C_g \cdot 1$$

$$\left.\begin{aligned} h(n) \cdot C_g &\le C_f n^2 \\ h(n) &\ge C_1 n^3 \end{aligned}\right)$$ ПРОТИВОРЕЧИЕ

$$\frac{h(n)}{n^2} \le C_f / C_g$$

$$\frac{h(n)}{n^2} \ge C_1 n$$

N 1.2. ВЕРХН.

$$\left.\begin{aligned} f(n) &= n^2 \\ g(n) &= 1 \end{aligned}\right)$$

$h(n) = O(n^2)$; ДОСТИГ.

НИЖН.

$s(n) = 0$ НЕ ЛЕЖИТ В КЛАССЕ $\mathbb{N} \to \mathbb{R}_+$

$$f_1(n) = n^{-15}$$

$$f_2(n) = n^{-100500}$$

$$2.\quad \sum_{i=1}^{n} \sqrt{i^3+2i+5} \geq \sum_{i=1}^{n} \sqrt{i^3} \geq \cancel{\sum_{i=1}^{n} n^{\frac{3}{2}} = n^{\frac{5}{2}}}$$

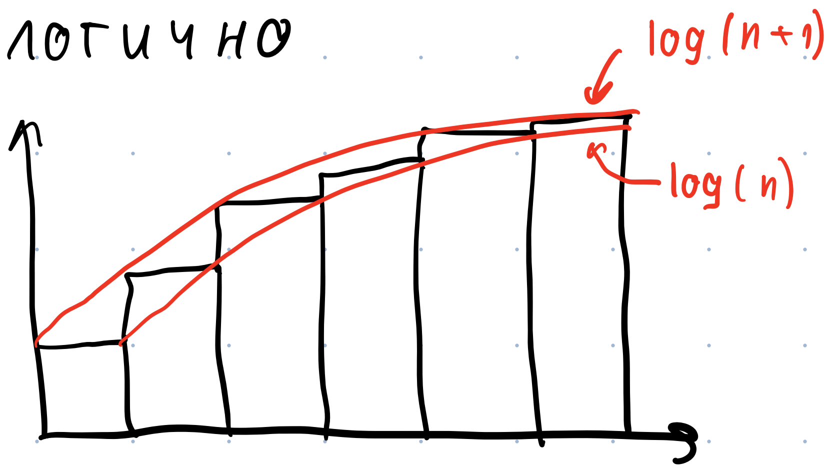
$$\geq \sum_{i=\frac{n}{2}}^{n} \sqrt{i^3} \geq \sum_{i=\frac{n}{2}}^{n} \sqrt{\left(\frac{n}{2}\right)^3} = C_1 n^{\frac{5}{2}}$$

$$\sum_{i=1}^{n} \sqrt{i^3+2i+5} \leq \sum_{i=1}^{n} \sqrt{i^3+2i^3+5i^3} = C_2 n^{\frac{5}{2}}$$

$$f(n) = \Theta\left(n^{\frac{5}{2}}\right)$$

3. АНАЛОГИЧНО

4.



$$4.\quad (1+1)^n = 1^n + 1^{n-1} C_n^1 1 + 1^{n-2} C_n^2 1^2 + \ldots$$

$$C_n^1 = \binom{n}{1}$$

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

$$5.\quad T(n) = \sum_{k=0}^{\lfloor \log n \rfloor} \sum_{j=0}^{2^{3k}} 1 + \sum_{m=0}^{\lfloor \frac{n}{2} \rfloor} \sum_{j=0}^{(1+2m)^2} 1 =$$

$$// \ i = 2^k \qquad // \ i = 1+2m$$

$$= \sum_{k=0}^{\lfloor \log n \rfloor} \left(2^{3k} + 1\right) + \sum_{m=0}^{\lfloor \frac{n}{2} \rfloor} \left((1+2m)^2 + 1\right) =$$

$$= \left[\lfloor \log n \rfloor + 1\right] + 1 + 8 + 64 + \ldots + 2^{3\lfloor \log n \rfloor} +$$

$$\frac{8^{\lfloor \log n + 1 \rfloor} - 1}{8 - 1}$$

$$2^{3 \log n}$$

$$2^{\log n^3}$$

$$\Theta(n^3) \qquad n^3$$

$$+ \sum_{m=0}^{\lfloor \frac{n}{2} \rfloor} 4m^2 + 4m + 2 = \Theta(n^3)$$

$$n^3$$

N6. $f(n) = 1 + c + c^2 + \ldots + c^n = \dfrac{c^{n+1} - 1}{c - 1}$

$c > 0$

1) $c = 1$ РАССМ. ОТДЕЛЬНО:

$$f(n) = 1^0 + 1^2 + \ldots + 1^n = n + 1 = \Theta(n)$$

2) $c > 1$

$$f(n) = \frac{c^{n+1} - 1}{c - 1} = \left[\frac{1}{c-1}\right]\left(c \cdot c^n - 1\right) = \Theta(c^n)$$

3) $c<1$

$$f(n)=\frac{1-c^{n+1}}{1-c} \geq \frac{1-c^{n+1}}{1} > 1-c^{1} = const$$

$$\leq \frac{1}{1-c} = const$$

$$f(n)=\Theta(1)$$

```
for i in range(n):
    print("text")
    print(i)
```

$T(n)=n$

$T(n)=c+c+2c+2c+3c+\ldots$

```
def print_all_numbers(max_number):
    for i in range(max_number):
        print(i)
```

длина входа $= n = \lceil \log_2 max\_number \rceil$

//сложн. по max_number линейна

сложн. по n $O(2^n)$

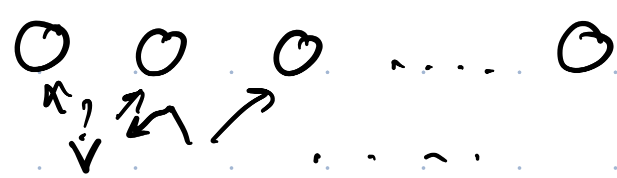
$$max\_number_2 = \underbrace{110\ldots101}_{n}$$

$$\left.\begin{array}{l} 00\ldots0 \\ 00\ldots1 \\ 00\ldots10 \\ 00\ldots11 \\ \ldots\ldots \\ 110\ldots101 \end{array}\right\} \text{max\_number печатей}$$

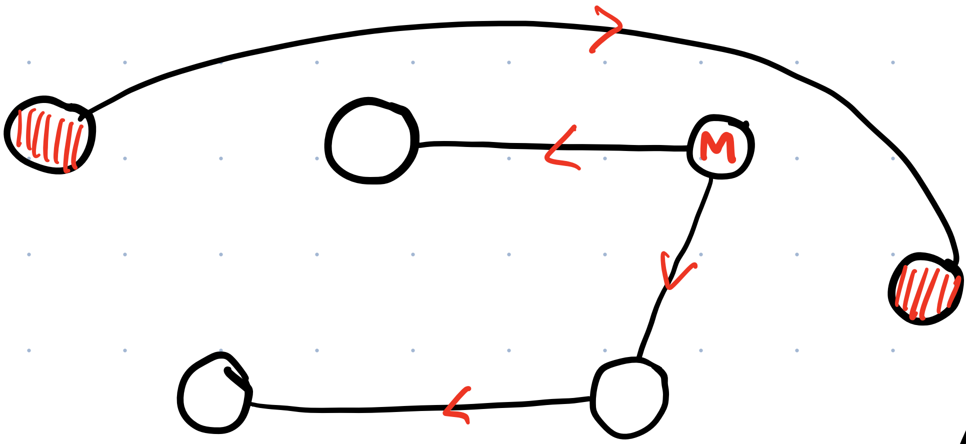
всевозм. строк бит. длины n 2^n

поиск max (оценка снизу)

n-1

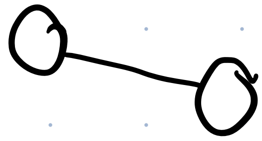


n-2



необх.

n-1



....

индукц.