Funkcje całkowitoliczbowe $(x \in \mathbb{R}, n \in \mathbb{Z})$

$\lfloor x \rfloor = n \iff n \le x < n+1$

$\lceil x \rceil = n \iff n-1 < x \le n$

$\{x\} = x - \lfloor x \rfloor$ (część ułamkowa x)

$\lfloor 3,14 \rfloor = 3$ $\quad \lfloor -3,14 \rfloor = -4$

$\lceil 3,14 \rceil = 4$ $\quad \lceil -3,14 \rceil = -3$

$\{3,14\} = 0.14$ $\quad \{-3.14\} = 0.86$

$\lfloor -x \rfloor = -\lceil x \rceil \quad (=n)$

$\lfloor -x \rfloor = n \iff n \le -x < n+1 \iff -n-1 < x \le -n \iff \lceil x \rceil = -n$

$\lfloor x+n \rfloor = \lfloor x \rfloor + n \qquad \lceil x+n \rceil = \lceil x \rceil + n$

$n = \left\lfloor \frac{n}{2} \right\rfloor + \left\lceil \frac{n}{2} \right\rceil$

Przybliżenie x do najbliższej liczby całkowitej:

$\lfloor x + \frac{1}{2} \rfloor, \lceil x - \frac{1}{2} \rceil$

Symbole asymptotyczne:

$f(n) = O(g(n)) \iff \exists c \ f(n) \le c \cdot g(n)$ dla prawie wszystkich n ($\exists n_0$, nierówność zachodzi dla $n > n_0$) (duże "O")

$f(n) = \Omega(g(n)) \iff \exists c>0 \ f(n) \ge c \cdot g(n)$ (omega)

$f(n) = \Theta(g(n)) \iff \exists c,d > 0 \ c \cdot g(n) \le f(n) \le d \cdot g(n)$ (theta)

$f(n) = o(g(n)) \iff \frac{f(n)}{g(n)} \xrightarrow[n \to \infty]{} 0$ (małe "o")

$f(n) \sim g(n) \iff \frac{f(n)}{g(n)} \xrightarrow[n \to \infty]{} 1$ (asymptotyczne)

Złożoność obliczeniowa algorytmów:

1 Znajdowanie max liczb $a_1, a_2, \dots, a_n$ $\to$ algorytm: $m \leftarrow a_1$
dla $i \in n$
$a_i > m$: $m \leftarrow a_i$

Czas działania $\approx c \cdot n$ $\qquad$ Czas działania $\equiv$ Złożoność obliczeniowa $O(n)$ (pr[illegible] $\Theta(n)$)

2 Mnożenie macierzy $n \times n$

$$\{a_{ij}\} = \begin{bmatrix} a_{11} & \cdots & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{nn} \end{bmatrix} \qquad \{b_{ij}\} = \begin{bmatrix} b_{11} & \cdots & b_{n1} \\ \vdots & \ddots & \vdots \\ b_{1n} & \cdots & b_{nn} \end{bmatrix} \qquad \{a_{ij}\} \cdot \{b_{ij}\} = \{c_{ij}\}$$

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Czas mnożenia macierzy z def $\approx c \cdot n^3$ $\quad O(n^3)$

Algorytm Strassena $\quad O(n^{\log_2 7}) = O(n^{2.81})$

Najlepsze algorytmy $\quad O(n^{2.37})$

„nowy komputer" 10x szybszy

$f(n)$	$f(10)$	$f(100)$	$f(1000)$	$10\,f(n)$
$10 \log n$	33.2	66.4	94.7	$f(n^{10})$
n	10	100	1000	$f(10n)$
$n \log n$	33.2	664	9970	$\sim f(10n)$
n^2	100	10000	10^6	$f(3.16n) \longleftarrow f(\sqrt{10}\,n)$
n^3	1000	1000000	10^9	$f(2.15n) \longleftarrow f(\sqrt[3]{10}\,n)$
1.1^n	2.6	13780	$2.46 \cdot 10^{41}$	$f(n+24.1)$

$T(n)$ - czas działania algorytmu dla problemu rozmiaru n

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

$|\sin(n)| = O(1)$, bo $|\sin(n)| \leq 1$

$(n+1)^2 \sim n^2$, bo $\frac{(n+1)^2}{n^2} \xrightarrow[n\to\infty]{} 1$

$n = o(n^2)$, bo $\frac{n}{n^2} \xrightarrow[n\to\infty]{} 0$

$(n+1)^2 = \Theta(n^2)$, bo $n^2 \leq (n+1)^2 = n^2 + 2n + 1 \leq 4n^2$ (?)

małe "o" oznacza relację porządku na funkcjach $\quad f(n) \prec g(n) \Leftrightarrow f(n) = o(g(n))$

$$1 \prec \log\log n \prec (\log n)^a \prec n^b \prec c^n \qquad a, b > 0,\ c > 1$$

$$\frac{x}{e^x} \xrightarrow[x \to \infty]{} 0$$

$$\lim_{n\to\infty} \frac{n}{e^n} = \lim_{n\to\infty} \underbrace{\left(\frac{n}{(n-1)e}\right)}_{\leq \frac{2}{e}} \cdot \underbrace{\left(\frac{n-1}{(n-2)e}\right)}_{\leq \frac{2}{e}} \cdot \ldots \cdot \underbrace{\left(\frac{2}{1e}\right)}_{\leq \frac{2}{e}} \cdot \underbrace{\frac{1}{e}}_{\leq \frac{2}{e}} \leq \lim_{n\to\infty} \left(\frac{2}{e}\right)^n = 0$$

$$n^b \prec c^n$$

$$\lim_{n\to\infty} \frac{n^b}{c^n} = \lim_{n\to\infty} \left(\frac{n}{c^{\frac{n}{b}}}\right)^b = \lim_{n\to\infty} \left(\underbrace{\frac{n \cdot \ln\frac{c}{b}}{e^{n \cdot \ln \frac{c}{b}}}}_{\to 0} \cdot \underbrace{\frac{1}{\ln \frac{c}{b}}}_{\to 0}\right)^b = 0$$

$$(\log n)^a \prec n^b \qquad (N = \log n)$$

z powyższej funkcji

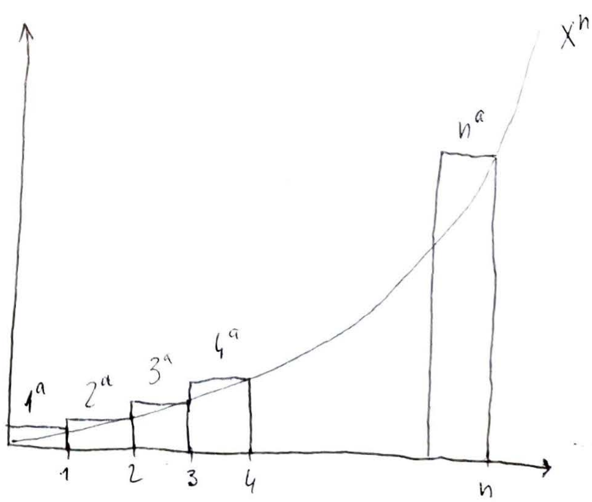
$$\lim_{n\to\infty} \frac{(\log n)^a}{n^b} = \lim_{N\to\infty} \frac{N^a}{(2^b)^N} = \lim_{N\to\infty} \frac{N^\square}{\triangle^N} = 0$$

$$(n+1)^2 = n^2 + 2n + 1 = n^2 + \Theta(n) = n^2 + O(n) = n^2 + o(n^2)$$

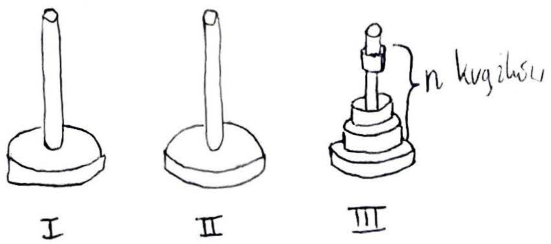
$$\sum_{k=1}^{n} k^a = \frac{1}{a+1} n^{a+1} + O(n^a)$$

$$\sum_{k=1}^{n} k^a = \int_0^n x^a + R, \quad R < n^a, \quad R = O(n^a)$$

$$\int_0^n x^a = \frac{1}{a+1} n^{a+1}$$



Zależności rekurencyjne (na przykładzie wieży Hanoi)



T(n) - liczba kroków potrzebnych do przestawienia n krążków z pręta na pręt

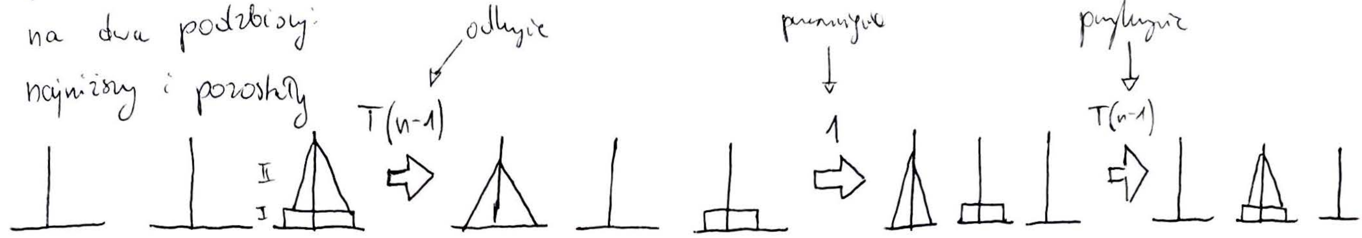
$T(0) = 0$

$T(1) = 1$

$T(2) = \ldots = 3$

$$T(n) \leq \underbrace{2T(n-1)+1}_{}$$

podzielenie krążków na dwa podzbiory: najniższy i pozostały



$$T(n) = 1 + 2T(n-1)$$

$T(0)=0$

$$= 1 + 2 + 4T(n-2)$$

$$= 1 + 2 + 4 + 8 + \ldots + 2^{k-1} + 2^k T(n-k)$$

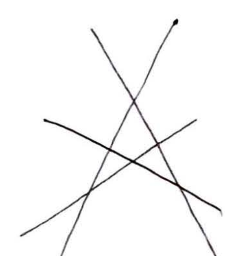
zał. $k=n$

$$= 1 + 2 + 4 + 8 + \ldots + 2^{n-1}$$

$$= 2^n - 1$$

Proste i obszary na płaszczyźnie

$L(n) \leftarrow$ liczba obszarów (maksymalna) na płaszczyźnie w zależności od ilości prostych



n prostych + dokładamy jedną nową prostą

$L(0) = 1$

$L(1) = 2$

$L(2) = 4$

$L(n) = ?$

$$L(n+1) = n + 1 + L(n)$$

$$L(n) = n + L(n-1)$$

$$= n + (n-1) + L(n-2)$$

$$= n + (n-1) + \ldots + (n-k-1) + L(n-k)$$

zał. $k=n$

$$= \underbrace{n + (n-1) + \ldots + 1}_{\text{suma n liczb naturalnych}} + 1 = \frac{n(n+1)}{2} + 1$$