

Algorytmy i struktury danych - Złożoność algorytmów

Marcin Żurowski

09 grudnia 2025

Potęga iteracyjnie

```
POTEGA(a, n)
```

```
  a  $\in$  R
```

```
  n  $\in$  N
```

```
  pot = 1
```

```
  for i = 1 to n
```

```
    pot = pot * a
```

```
  return pot
```

Potęga binarnie

```
POTEGA(a, n)
  a ∈ ℝ
  n ∈ ℕ
  i = n
  pot = 1
  pod = a
  while i ≠ 0
    if i MOD 2 ≠ 0
      pot = pot * pod
    pod = pod * pod
    i = i DIV 2
  return pot
```

Potęga rekurencyjnie

```
POTEGA(a, n)
  a ∈ ℝ
  n ∈ ℕ
  if n = 0
    return 1
  else
    return a * POTEGA(a, n - 1)
```

Szacowanie funkcji

$$\begin{aligned} \exists_{n_0 \in \mathbb{N}} \forall_{n \geq n_0} 1 \leq \lg n \leq n \leq n \lg n \leq n^2 \leq n^3 \leq \dots \leq n^k \leq \\ \leq \dots \leq 2^n \leq n! \leq n^n \end{aligned}$$

Zadania

- 1 $f(n) = 13n^2 + 4n - 73$
- 2 $f(n) = (n^2 + 1)(2n^4 + 3n - 8)$
- 3 $f(n) = (n^3 + 3n - 1)^4$
- 4 $f(n) = \sqrt{n+1}$
- 5 $f(n) = (n^2 - 1)^7$
- 6 $f(n) = \sqrt{n^2 - 1}$
- 7 $f(n) = \sqrt{n^2 + n}$
- 8 $f(n) = (n^2 + n + 1)(n^3 + 5)$
- 9 $f(n) = 3^n$
- 10 $f(n) = 2^{n+1}$
- 11 $f(n) = 2^{2n}$
- 12 $f(n) = (n+1)^2$
- 13 $f(n) = (200n)^2$

Notacje asymptotyczne

- $O(g(n)) = \{f(n) : \exists_{c>0} \exists_{n_0 \in \mathbb{N}} \forall_{n \geq n_0} 0 \leq f(n) \leq cg(n)\}$
- $\Theta(g(n)) =$
 $\{f(n) : \exists_{c_1>0} \exists_{c_2>0} \exists_{n_0 \in \mathbb{N}} \forall_{n \geq n_0} 0 \leq c_1g(n) \leq f(n) \leq c_2g(n)\}$
- $\Omega(g(n)) = \{f(n) : \exists_{c>0} \exists_{n_0 \in \mathbb{N}} \forall_{n \geq n_0} 0 \leq cg(n) \leq f(n)\}$

Twierdzenie o rekurencji uniwersalnej

Niech $a \geq 1$ i $b > 1$ będą stałymi, niech $f(n)$ będzie pewną funkcją i niech $T(n)$ będzie zdefiniowane dla nieujemnych liczb całkowitych przez rekurencję

$$T(n) = aT(n/b) + f(n),$$

gdzie n/b interpretujemy jako $\lfloor n/b \rfloor$ lub $\lceil n/b \rceil$. Wtedy funkcja $T(n)$ może być ograniczona asymptotycznie w następujący sposób:

- 1 Jeśli $f(n) = O(n^{\log_b a - \varepsilon})$ dla pewnej stałej $\varepsilon > 0$, to $T(n) = \Theta(n^{\log_b a})$.
- 2 Jeśli $f(n) = \Theta(n^{\log_b a})$, to $T(n) = \Theta(n^{\log_b a} \lg n)$.
- 3 Jeśli $f(n) = \Omega(n^{\log_b a + \varepsilon})$ dla pewnej stałej $\varepsilon > 0$ oraz $af(n/b) \leq cf(n)$ dla pewnej stałej $c < 1$ i wszystkich dostatecznie dużych n , to $T(n) = \Theta(f(n))$.

Przykłady:

- ❶ $T(n) = 9T(n/3) + n$
- ❷ $T(n) = T(2n/3) + 1$
- ❸ $T(n) = 3T(n/4) + n \lg n$
- ❹ $T(n) = 2T(n/2) + n \lg n$
- ❺ $T(n) = 4T(n/2) + n$
- ❻ $T(n) = T(n-1) + 1$
- ❼ $T(n) = 4T(n/2) + n^2$
- ❽ $T(n) = 4T(n/2) + n^3$
- ❾ $T(n) = 2T(n/2) + n^2$
- ❿ $T(n) = T(9n/10) + n$
- ⓫ $T(n) = 16T(n/4) + n^2$
- ⓬ $T(n) = 7T(n/3) + n^2$
- ⓭ $T(n) = 2T(n/4) + \sqrt{n}$
- ⓮ $T(n) = T(n-1) + n$
- ⓯ $T(n) = 9T(n/3) + n^3$
- ⓰ $T(n) = T(n/3) + 1$