Algoritmus a jeho výpočtová zložitosť. Porovnávanie rastu funkcií.

Algoritmus

Krok za krokom špecifikovaný postup riešiaci daný problém v konečnom čase.

Algoritmus

Konečná množina definovaných krokov v pseudojazyku, riešiacich úlohu, ktorá zadaný vstupný stav, po zastavení, transformuje na požadovaný výstupný stav.

Základné vlastnosti algoritmu

- hromadnosť
- determinovanosť
- rezultatívnosť

Príklad

Nájdite algoritmus, ktorý zistí, či zadané číslo $n > 1$ je alebo nie je prvočíslo a vypíše túto informáciu na obrazovku.

Príklad

```
#include <stdio.h>
#include <math.h>

void main(void)
{
   int i, n;
   printf("Zadaj cislo:");
   scanf("%d", &n);

   for(i=2;i<=sqrt(n);++i)
      if (n % i == 0) {printf("%d nie je prvocislo", n); return ;}
   printf("%d je prvocislo", n);
}
```

Výpočtová zložitosť algoritmu

Funkcia dĺžky času, potrebného na výpočet, v závislosti od veľkosti vstupu (v bitoch).

Pamäťová zložitosť algoritmu

Funkcia veľkosti pamäte, potrebnej na výpočet, v závislosti od veľkosti vstupu (v bitoch).

Zložitosť algoritmu

Výmena času za pamäť.

Zložitosť algoritmu

- zložitosť v najhoršom prípade
- priemerná zložitosť

Výpočtová zložitosť algoritmu z predch. príkladu

Vstup: $n \in \mathbb{N}$
Cyklus beží najviac $\sqrt{n}$ krát
Na reprezentáciu n treba $m \sim \log_2 n$ bitov
Výpočtová zložitosť: $T(m) \sim \sqrt{2^m} = 2^{m/2}$

,,Rýchly'' algoritmus

Algoritmus je rýchly, keď existuje taký polynóm P, že pre každý vstup s dĺžkou B bitov je čas výpočtu menší než hodnota $P(B)$.

„Pomalý" algoritmus

Algoritmus je pomalý, keď pre každý polynóm P existuje taký vstup s dĺžkou B bitov, že čas výpočtu je väčší než hodnota $P(B)$.

Príklad

Problém ruksaku.
Superrastúci ruksak.

Poznámky

- Tvrdenie, že algoritmus má zložitosť $T(n)$ pre triedu problémov znamená, že $T(n)$ je horný odhad pre dĺžku výpočtu každého problému danej triedy.
- Algoritmus môže byť zložitý pre celú triedu problémov, ale ľahký pre jej časť.
- Tvrdenie, že algoritmus je rýchly pre triedu problémov znamená, že je rýchly pre každý problém danej triedy.

$f(n), g(n) : R^+ \to R^+$

Definition

$$f(n) \sim g(n) \Leftrightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = 1.$$

Príklad

Example

$n^2 + 5 \sim n^2$

Definition

$f(n) = \theta(g(n)) \Leftrightarrow \exists c_1, c_2 \in \mathbb{R}, c_1 > 0, c_2 > 0$ a $\exists n_0 \in \mathbb{N}$ také, že $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \quad \forall n \geq n_0$.

Príklad

Example

$n^2 - 1 = \theta(n^2)$

Example

$5 + \sin(x) = \theta(2)$

Definition

$$f(n) = o(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$

Definition

$f(n) = o(g(n)) \Leftrightarrow \forall c \in \mathbb{R}, c > 0, \exists n_0 \in \mathbb{N}$ také, že $0 \leq f(n) < cg(n), \forall n \geq n_0$.

Príklad

Example

$n = o(n^2)$

Definition

$f(n) = \mathcal{O}(g(n)) \Leftrightarrow \exists c \in \mathbb{R}, c > 0, n_0 \in \mathbb{N}$ také, že
$0 \leq f(n) \leq cg(n) \quad \forall n \geq n_0$.

Príklad

Example

$5n = \mathcal{O}(n)$

Example

$n = \mathcal{O}(n^5)$

Example

$e^{1/n} = \mathcal{O}(1)$

Theorem

$f(n) = o(g(n)) \Rightarrow f(n) = \mathcal{O}(g(n)).$

Definition

$f(n) = \Omega(g(n)) \Leftrightarrow \exists c \in \mathbb{R}, c > 0, n_0 \in \mathbb{N}$ také, že
$0 \leq cg(n) \leq f(n), \forall n \geq n_0$.

Príklad

Example

$5n = \Omega(n)$

Example

$n^5 = \Omega(n^4)$

Definition

$f(n) = \omega(g(n)) \Leftrightarrow \forall c \in \mathbb{R}, c > 0, \exists n_0 \in \mathbb{N}$ také, že
$0 \leq cg(n) < f(n)$.

Príklad

Example

$n^2/2 = \omega(n)$

Theorem

$f(n) = \theta(g(n)) \Leftrightarrow f(n) = \mathcal{O}(g(n))$ *a zároveň* $f(n) = \Omega(g(n)))$.

Analógia medzi porovnávaním rastu funkcií f, g a reálnych čísel a, b:

$$
\begin{array}{lcl}
f(n) = \mathcal{O}(g(n)) & \approx & a \leq b \\
f(n) = o(g(n)) & \approx & a < b \\
f(n) = \theta(g(n)) & \approx & a \cong b \\
f(n) \sim g(n) & \approx & a = b \\
f(n) = \Omega(g(n)) & \approx & a \geq b \\
f(n) = \omega(g(n)) & \approx & a > b
\end{array}
$$

Tranzitívnosť

$$f(n) = \theta(g(n)) \text{ a } g(n) = \theta(h(n)) \Rightarrow f(n) = \theta(h(n))$$
$$f(n) = \mathcal{O}(g(n)) \text{ a } g(n) = \mathcal{O}(h(n)) \Rightarrow f(n) = \mathcal{O}(h(n))$$
$$f(n) = \Omega(g(n)) \text{ a } g(n) = \Omega(h(n)) \Rightarrow f(n) = \Omega(h(n))$$
$$f(n) = o(g(n)) \text{ a } g(n) = o(h(n)) \Rightarrow f(n) = o(h(n))$$
$$f(n) = \omega(g(n)) \text{ a } g(n) = \omega(h(n)) \Rightarrow f(n) = \omega(h(n))$$

Reflexívnosť

$$f(n) = \theta(f(n))$$
$$f(n) = \mathcal{O}(f(n))$$
$$f(n) = \Omega(f(n))$$

Symetrickosť

$$f(n) = \theta(g(n)) \Leftrightarrow g(n) = \theta(f(n)).$$

Transponovaná symetrickosť

$$f(n) = \mathcal{O}(g(n)) \Leftrightarrow g(n) = \Omega(f(n)).$$
$$f(n) = o(g(n)) \Leftrightarrow g(n) = \omega(f(n)).$$