MC358 - Fundamentos matemáticos da computação

Prof. Dr. Hilder Vitor Lima Pereira

Instituto de computação

UNICAMP

01 de outubro de 2023

1. Comentários sobre propriedades do $\mathbb{Z}_n$

2. Notação assintótica para funções

3. Perguntas, observações, comentários?

Comentários sobre propriedades do $\mathbb{Z}_n$

Algoritmo de Euclides estendido para achar inversos

Como visto no começo do curso, o seguinte algoritmo calcula o `mdc` e os coeficientes de Bézout.

Algorithm: `AlgoEuclidesEstendido`

Input: $a, b \in \mathbb{N}$ com $a \geq b \geq 0$.

Output: $(d, u, v) \in \mathbb{Z}^3$ tais que $d = \texttt{mdc}(a, b)$ e $d = u \cdot a + v \cdot b$

if $0 == b$ then
    return $(a, 1, 0)$
$q, r = \texttt{DivEuc}(a, b)$
$(d, u, v) = \texttt{AlgoEuclidesEstendido}(b, r)$
return $(d, v, u - q \cdot v)$

Algoritmo de Euclides estendido para achar inversos

Como visto no começo do curso, o seguinte algoritmo calcula o `mdc` e os coeficientes de Bézout.

Algorithm: `AlgoEuclidesEstendido`

Input: $a, b \in \mathbb{N}$ com $a \geq b \geq 0$.
Output: $(d, u, v) \in \mathbb{Z}^3$ tais que $d = \mathtt{mdc}(a, b)$ e $d = u \cdot a + v \cdot b$

```
if 0 == b then
    return (a, 1, 0)
q, r = DivEuc(a, b)
(d, u, v) = AlgoEuclidesEstendido(b, r)
return (d, v, u − q · v)
```

Vamos usá-lo para calcular $7^{-1} \pmod{100}$.

a	b	q	r	d	u	v
100	7					

Mapear para $\mathbb{Z}_n$ pode ser útil

Se uma equação com coeficientes em $\mathbb{Z}$ tem solução inteira, então ela tem solução em $\mathbb{Z}_n$ para todo $n \geq 2$.

Contrapositiva: Se existe $n \geq 2$ tal que a equação não tem solução em $\mathbb{Z}_n$, então ela não tem solução inteira.

Mapear para $\mathbb{Z}_n$ pode ser útil

Se uma equação com coeficientes em $\mathbb{Z}$ tem solução inteira, então ela tem solução em $\mathbb{Z}_n$ para todo $n \geq 2$.

Contrapositiva: Se existe $n \geq 2$ tal que a equação não tem solução em $\mathbb{Z}_n$, então ela não tem solução inteira.

———

Um inteiro x é divisível por n se, e somente se, $x \equiv 0 \pmod{n}$.

Então, para mostrar que uma expressão é divisível por algum valor, podemos mapear a expressão em $\mathbb{Z}_n$ e mostrar que ela é igual zero.

Notação assintótica para funções

Introdução

Podemos ter vários algoritmos diferentes para resolver um mesmo problema. Então, qual algoritmo é melhor?

Podemos ter vários algoritmos diferentes para resolver um mesmo problema. Então, qual algoritmo é melhor?

Normalmente, queremos o algoritmo que executa menos operações.

Escrevemos o número de operações como uma função de n, o tamanho da entrada. Por exemplo,

$$f(n) = 2n^2 + 100$$

No entanto, normalmente, não ligamos tanto para as constantes.

Isso nos leva agrupar funções que tem o mesmo comportamento para *n grande*...

Detalhe técnico

Em toda nossa discussão sobre comportamento assintótico de funções, vamos considerar apenas funções com domínio $\mathbb{N}$ e que são assintoticamente positivas.

Limitante superior assintótico, ou Big-Oh

Primeira forma de agrupar funções:

$$O(f(n)) = \{g(n) : g(n) \text{ cresce mais lentamente que } f(n)\}$$

Primeira forma de agrupar funções:

$$O(f(n)) = \{g(n) : g(n) \text{ cresce mais lentamente que } f(n)\}$$

Definição formal:

$$O(f(n)) = \{g(n) : \exists (n_0, c) \in \mathbb{N} \times \mathbb{R}_{>0} \; (\forall n \geq n_0 \; g(n) \leq c \cdot f(n))\}$$

Exemplos

1. Para cada $f(n)$, mostre que $f(n) \in O(n^2)$.
   1.1 $f(n) = 4n^2$
   1.2 $f(n) = 4n^2 + 2n$
   1.3 $f(n) = 4n^2 + 2n + 100$

Exemplos

1. Para cada $f(n)$, mostre que $f(n) \in O(n^2)$.
   1.1 $f(n) = 4n^2$
   1.2 $f(n) = 4n^2 + 2n$
   1.3 $f(n) = 4n^2 + 2n + 100$
2. Prove ou refute que $0,0000001 \cdot n^3 \in O(n^2)$.

Exemplos

1. Para cada $f(n)$, mostre que $f(n) \in O(n^2)$.
   1.1 $f(n) = 4n^2$
   1.2 $f(n) = 4n^2 + 2n$
   1.3 $f(n) = 4n^2 + 2n + 100$
2. Prove ou refute que $0,0000001 \cdot n^3 \in O(n^2)$.
3. Prove ou refute que $\log_{10}(n) \in O(\log_2(n))$.

Limitante inferior assintótico, ou Omega

$$\Omega(f(n)) = \{g(n) : g(n) \text{ cresce mais rapidamente que } f(n)\}$$

Limitante inferior assintótico, ou Omega

$$\Omega(f(n)) = \{g(n) : g(n) \text{ cresce mais rapidamente que } f(n)\}$$

Definição formal:

$$\Omega(f(n)) = \{g(n) : \exists(n_0, c) \in \mathbb{N} \times \mathbb{R}_{>0} \ (\forall n \geq n_0 \ g(n) \geq c \cdot f(n))\}$$

Exemplos

Para cada $f(n)$ e $g(n)$, prove ou refute que $f(n) \in \Omega(g(n))$.

1. $f(n) = 4n^2 + 2n + 100$ e $g(n) = n^2$
2. $f(n) = n^2/4 - 2n - 100$ e $g(n) = n^2$
3. $f(n) = 2^n$ e $g(n) = n^{1000}$
4. $f(n) = \sqrt{n}$ e $g(n) = \log n$.
5. $f(n) = n^\epsilon$, onde $\epsilon \in]0, 1[$, e $g(n) = \log n$.
6. $f(n) = n$ e $g(n) = n \log n$.

Perguntas, observações, comentários?