Processamento da Informação

Fabrício Olivetti de França

02 de Fevereiro de 2019

UFABC

CMCC
Centro de Matemática, Computação e Cognição
Center for Mathematics, Computation and Cognition

Topics

1. Laços de Repetição

Laços de Repetição

Revisitando o MDC

Nosso algoritmo de Euclides para o cálculo do MDC foi definido como:

```
def mdc(a, b):
    if a==0:
        return b
    elif b==0:
        return a
    else:
        return mdc(b, a%b)
```

Revisitando o MDC

O que estamos falando nesse código é:

- Se a ou b for zero, retorne o que não for zero.
- Senão, repita o procedimento substituindo o valor de `a` por `b` e de `b` por `a % b`.

Revisitando o MDC

Em outras palavras:

- Repita enquanto a e b diferentes de zero:
  - `a` recebe o valor de `b` e `b` recebe o valor de `a % b`

Enquanto

A instrução `while` faz justamente isso que queremos:

```
while condicao:
    instrucao
```

“Enquanto a condição for verdadeira, repita a instrução”

Revisitando o MDC

```
def mdc(a, b):
    while a != 0 and b != 0:
        a = b
        b = a%b
    return max(a, b)
```

Tem um erro nesse código! Qual é?

Revisitando o MDC

```
def mdc(a, b):
    while a != 0 and b != 0:
        r = a%b
        a = b
        b = r
    return max(a, b)
```

Revisitando o MDC

Podemos melhorar!

```
def mdc(a, b):
    while b != 0:
        r = a%b
        a = b
        b = r
    return a
```

Exercício

Defina a função `soma(n)` que retorna a somatória dos n primeiros números naturais.

Exercício

A multiplicação Etíope de dois números naturais `x, y` funciona da seguinte forma:

1) Iniciamos com o resultado contendo valor 0
2) Se x igual a zero, retorna o resultado
3) Se x for ímpar, adiciona y ao resultado
4) Divide x pela metade e dobra o valor de y
5) Retorna ao passo 2

Exercício

x	y	r
19	34	0
9	68	34
4	136	102
2	272	102
1	544	102
0	1088	646

Exercício

1) Reescreva a descrição utilizando a ideia de repetição *enquanto*
2) Escreva a definição da função `etiope(x, y)`

Curiosidade

Multiplicação e divisão de inteiros por potências de 2 é barato para o computador! Basta andar n bits para esquerda ou direita!

Mesmo assim o uso do operador $*$ é mais rápido que a multiplicação etíope.

Exercício

Converta um número natural *x* em um número binário *b*. Represente um número binário como uma String, lembrem-se que o operador `+` concatena Strings.

A instrução `str(x)` converte um número para uma String. Ex.: `str(123) == "123"`.

Exercício

O algoritmo aprendido em NI funciona da seguinte forma:

- Enquanto o número não se tornar 0
  - Calcule o resto da divisão por 2 e anote no seu número binário
  - Divida o número por 2

Exercício

x	b
13	0
6	1
3	01
1	101
0	1101

Exercício

Determine se um número é primo. Vamos utilizar nossa função `divisivel(x,n)`:

```
def divisivel(x, n):
    return x%n == 0
```

Dicas

- Todo número é divisível por 1 e por ele mesmo.
- Um número primo só é divisível por 1 e por ele mesmo.
- Se eu encontrar um número divisível que não seja nenhum desses dois, posso assumir que não é primo.

Exercício

Dada a definição da função `soma_digitos` que soma os dígitos de um número:

```
def soma_digitos(x):
    s = 0
    while x != 0:
        s = s + ultimo_digito(x)
        x = extrai_ultimo_digito(x)
    return s
```

Defina `ultimo_digito, extrai_ultimo_digito`.