

ALESSANDRO SOARES DA SILVA

MATRICULA: 20231023705

## 1º LISTA DE ALGORITMO

1.1 – Para cada função  $f(n)$  e cada tempo  $t$  na tabela a seguir, determine o tamanho  $n$  de um problema que pode ser resolvido no tempo  $t$ , considerando que o algoritmo para resolver o problema demore  $f(n)$  microssegundos.

	1 segundo	1 minuto	1 hora	1 dia	1 mês	1 ano	1 século
$\log(n)$	$2^{1 \times 10^6}$	$2^{6 \times 10^7}$	$2^{3.6 \times 10^9}$	$2^{8.6 \times 10^{10}}$	$2^{2.6 \times 10^{12}}$	$2^{3.15 \times 10^{13}}$	$2^{3.15 \times 10^{15}}$
$\sqrt{n}$	$1 \times 10^{12}$	$3.6 \times 10^{15}$	$1.29 \times 10^{19}$	$7.46 \times 10^{21}$	$6.72 \times 10^{24}$	$9.95 \times 10^{26}$	$9.96 \times 10^{30}$
$n$	$1 \times 10^6$	$6 \times 10^7$	$3.6 \times 10^9$	$8.6 \times 10^{10}$	$2.6 \times 10^{12}$	$3.15 \times 10^{13}$	$3.15 \times 10^{15}$
$n \log(n)$	62805	2811900	133967668	2747894153	$7.2 \times 10^{10}$	$7.97 \times 10^{11}$	$6.85 \times 10^{13}$
$n^2$	1000	7745	60000	293938	1609968	5615692	56176151
$n^3$	100	391	1532	4420	13736	31593	146679
$2^n$	19	25	31	36	41	44	51
$n!$	9	11	12	13	15	16	17

Sendo  $\log$  na base 2.

$$f(n) = \log n \therefore \frac{\log(n)}{10^6} = 1 \text{segundo} \therefore \log(n) = 1 \times 10^6 \therefore 2^{1 \times 10^6}$$

$$f(n) = \sqrt{n} \therefore \frac{n^{1/2}}{10^6} = 1 \text{segundo} \therefore n^{1/2} = 10^6 \therefore n = 10^{12}$$

$$f(n) = n \log n \therefore p \setminus t = 1 \text{minuto}$$

$$n \log n = 6 \times 10^7 \therefore n \frac{\log_{10} n}{\log_{10} 2} = 6 \times 10^7 \therefore n \log_{10} n = \log_{10} 2 \times 6 \times 10^7 \therefore \text{sendo } \log_{10} 2 \approx 0.301$$

$$\text{fazendo } a = \log_{10} n; n = 10^a \Rightarrow na = \log_{10} 2 \times 6 \times 10^7$$

$$\log_{10} n + \log_{10} a = \log_{10} 0.301 + \log_{10} 6 + \log_{10} 10^7 \Rightarrow a + \log_{10} a = 7.257$$

$$a \quad a + \log_{10} a$$

$$7 \quad 7.845$$

$$6.5 \quad 7.313$$

$$6.4 \quad 7.206$$

$$6.44 \quad 7.249$$

$$6.449 \quad 7.258$$

$$\Rightarrow a = 6.449; n = 10^a = 2811900$$

$$f(n)=n^2 \quad \therefore \quad n^2=10^6 \quad \therefore \quad n^{2/2}=10^{6/2} \quad \therefore \quad n=10^3$$

$$f(n)=n^3 \quad \therefore \quad n^3=10^6 \quad \therefore \quad n^{3/3}=10^{6/3} \quad \therefore \quad n=10^2$$

$$f(n)=2^n \quad \therefore \quad 2^n=10^6 \quad \therefore \quad \log(2^n)=\log(10^6) \quad \therefore \quad n\log(2)=\log(10^6) \quad \therefore \quad n=\frac{\log_{10}10^6}{\log_{10}2}$$

$$n \approx 19$$

2.3 – O fragmento de código a seguir implementa a regra de Horner para avaliar um polinômio

$$P(x) = \sum_{k=0}^n a_k x^k$$

$$P(x) = a_0 + x \left( a_1 + x \left( a_2 + \dots + x \left( a_{n-1} + x a_n \right) \dots \right) \right),$$

dados os coeficientes  $a_0, a_1, \dots, a_n$  e um valor para  $x$ .

```

1      y = 0
2      i = n
3      while i >= 0
4          do y = ai + x*y
5          i = i - 1

```

a. Qual é o tempo de execução desse fragmento de código em termos da notação  $\Theta$  para a regra de Horner?

**R = Considerando que o restante das operações são realizadas em tempo contante  $\Theta(1)$  e o loop está sendo executado  $n$  vezes, podemos concluir que o tempo de execução é  $\Theta(n)$**

b. Escreva pseudocódigo para implementar o algoritmo ingênuo de avaliação polinomial que calcula cada termo do polinômio desde o início. Qual é o tempo de execução desse algoritmo? Como ele se comporta com a regra de Horner?

```

In [1]: def polinomio(poly, n, x):
        for j in range(0, n):
            result = poly[j]
            for i in range(1, n):
                result = result*x + poly[i]
            return result

In [2]: poly = [2, 0, 3, 1]
        x = 2
        n = len(poly)

        print("Value of polynomial is ", polinomio(poly, n, x))

Value of polynomial is 23

```

Este código tem tempo de execução  $\Theta(n^2)$ , porque precisa realizar **for** indentado. Isso é mais lento que a regra de Horner.

c. Prove que a expressão a seguir é um loop invariante para o loop while das linhas 3 a 5.

No início de cada iteração do laço for nas linhas 2-3,

$$y = \sum_{k=0}^{n-(i+1)} a_{k+i+1} x^k$$

Interprete um somatório sem termos como igual a 0. Seguindo a estrutura do invariante de laço apresentado nesse capítulo, use esse invariante de laço para mostrar que, no término,  $\sum_{k=0}^n a_k x^k$ .

R = Inicialmente  $i = n$ , então, o limite superior da soma é **-1**, logo, a soma é avaliada como 0, que é o valor de y.

$$y = \sum_{k=0}^{n-n-1} a_{k+n+1} x^k \quad y = \sum_{k=0}^{-1} 0 x^k$$

Na manutenção:..  $i = n-i$

$$P = \sum_{k=0}^{i-1} a_{k+i+1} x^k \Rightarrow \text{Temos dentro do loop } P = A[i] + XP$$

$$P = A[i] + X \sum_{k=0}^{n-(i+1)} a_{k+i+1} x^k \Rightarrow P = A[i] + X \sum_{k=0+1}^{n-1-1+1} a_{k+i+1} x^{k-1} \Rightarrow P = A[i] X^0 + X \sum_{k=1}^{n-1} a_{k+i} x^{k-1} \Rightarrow$$

$$P = \sum_{k=0}^{n-1} a_{k+i} x^k$$

Na Terminação:.. O código termina quando  $i = -1$

$$P = \sum_{k=0}^{n-i-1} a_{k+i+1} x^k \Rightarrow P = \sum_{k=0}^{n-(1)-1} a_{k+-1+1} x^k \Rightarrow P = \sum_{k=0}^n a_k x^k$$

d. Conclua demonstrando que o fragmento de código dado avalia corretamente um polinômio caracterizado pelos coeficientes  $a_0, a_1, \dots, a_n$ .

Na Terminação:.. O código termina quando  $i = -1$

$$P = \sum_{k=0}^{n-i-1} a_{k+i+1} x^k \Rightarrow P = \sum_{k=0}^{n-(1)-1} a_{k+-1+1} x^k \Rightarrow P = \sum_{k=0}^n a_k x^k$$

### 3.1 – Seja

$$p(n) = \sum_{i=0}^d a_i n^i,$$

onde  $a_d > 0$ , um polinômio de grau d em n, e seja k uma constante. Use as definições das notações assintóticas para provar as propriedades a seguir:

a. Se  $k \geq d$ , então  $p(n) = O(n^k)$ .

R =  $\exists c \in \mathbb{R}^*, p(n) \leq g(n)$  para todo  $n \geq m \in \mathbb{N}^*$ , suponha que:

$$p(n) \leq c f(n) \Rightarrow \frac{p(n)}{f(n)} \leq c \frac{f(n)}{f(n)} \quad \frac{n^{(k)}}{n^k} = 1 \Rightarrow 1 \leq c \text{ ou } \frac{n^{(k-1)}}{n^k} \Rightarrow \frac{1}{n} \leq c$$

b. Se  $k \leq d$ , então  $p(n) = \Omega(n^k)$ .

R =  $\exists c \in \mathbb{R}^*, cg(n) \leq p(n)$  para todo  $n \geq m \in \mathbb{N}^*$ , suponha que:

$$cf(n) \leq p(n) \Rightarrow c \frac{f(n)}{f(n)} \leq \frac{p(n)}{f(n)} \quad \frac{n^{(k)}}{n^k} = 1 \Rightarrow c \leq 1 \text{ ou } \frac{n^{(k+1)}}{n^k} \Rightarrow c \leq n$$

c. Se  $k = d$ , então  $p(n) = \Theta(n^k)$ .

R = Apartir das questões anteriores temos que  $p(n)=O(n^k)$  e  $p(n)=\Omega(n^k)$ . Então temos que  $p(n) = \Theta(n^k)$ , se e somente se  $p(n)=\Omega(n^k)$  e  $p(n)=O(n^k)$ .

d. Se  $k > d$ , então  $p(n) = o(n^k)$ .

R =  $\exists c \in \mathbb{R}^*, p(n) \leq g(n)$  para todo  $n \geq m \in \mathbb{N}^*$ , suponha que:

$$p(n) \leq c f(n) \Rightarrow \frac{p(n)}{f(n)} \leq c \frac{f(n)}{f(n)} \quad \frac{n^{(k-1)}}{n^k} \Rightarrow \frac{1}{n} < c$$

a medida que  $n$  tende ao infinito, temos que para qualquer constante  $c$  o limite  $0 < f(n) < cg(n)$  é válido.

e. Se  $k < d$ , então  $p(n) = \omega(n^k)$ .

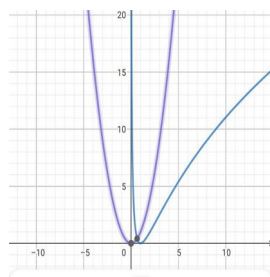
R =  $\exists c \in \mathbb{R}^*, cg(n) \leq p(n)$  para todo  $n \geq m \in \mathbb{N}^*$ , suponha que:

$$cf(n) \leq p(n) \Rightarrow c \frac{f(n)}{f(n)} \leq \frac{p(n)}{f(n)} \quad \frac{n^{(k+1)}}{n^k} \Rightarrow c < n$$

a medida que  $n$  tende ao infinito, temos que para qualquer constante  $c$  o limite  $0 < cg(n) < f(n)$  é válido

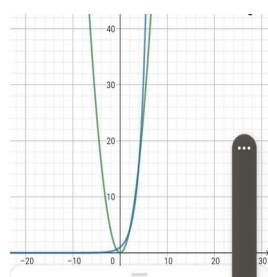
3.2 – Indique, para cada par de expressões (A,B) na tabela a seguir, se A é O, o,  $\Omega$ ,  $\omega$  ou  $\Theta$  de B. Considere que  $k \geq 1, \epsilon > 0$  e  $c > 1$  são constantes. Sua resposta deve estar na forma de tabela, com “sim” ou “não” escrito em cada retângulo.

A	B	O	o	$\Omega$	$\omega$	$\Theta$
$lg^k n$	$n^\epsilon$	Sim	Sim	Não	Não	Não
$n^k$	$c^n$	Sim	Sim	Não	Não	Não
$\sqrt{n}$	$n^{sen n}$	Não	Não	Não	Não	Não
$2^n$	$2^{n/2}$	Não	Não	Sim	Sim	Não
$n^{lg c}$	$c^{lg n}$	Sim	Não	Sim	Não	Sim
$lg(n!)$	$lg(n^n)$	Sim	Não	Sim	Não	Sim



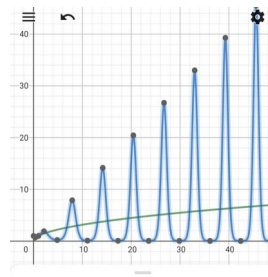
$f(x) = (\log_2(x))^2$

$g(x) = x^2$



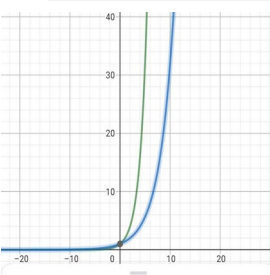
$f(x) = x^2$

$g(x) = 2^x$



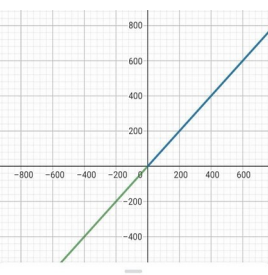
$f(x) = \sqrt{x}$

$g(x) = x^{sen(x)}$



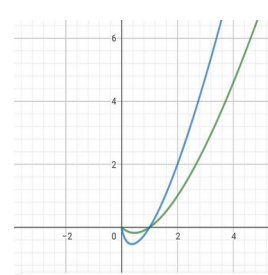
$f(x) = 2^x$

$g(x) = 2^{lg(x)}$



$f(x) = x^{sen(2)}$

$g(x) = 2^{lg(x)}$



$f(x) = \log_2(x!)$

$g(x) = \log_2(x^x)$