



ALGORITMOS E COMPLEXIDADE (ARA0174)

AULA 9

DATA: 14/04/2022



Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 1

Qual o termo de maior grau da complexidade do algoritmo apresentados pelas funções abaixo:

a) $f(n) = 6n^3 + 5n^2 + 1$

b) $f(n) = (n + 1)^2$

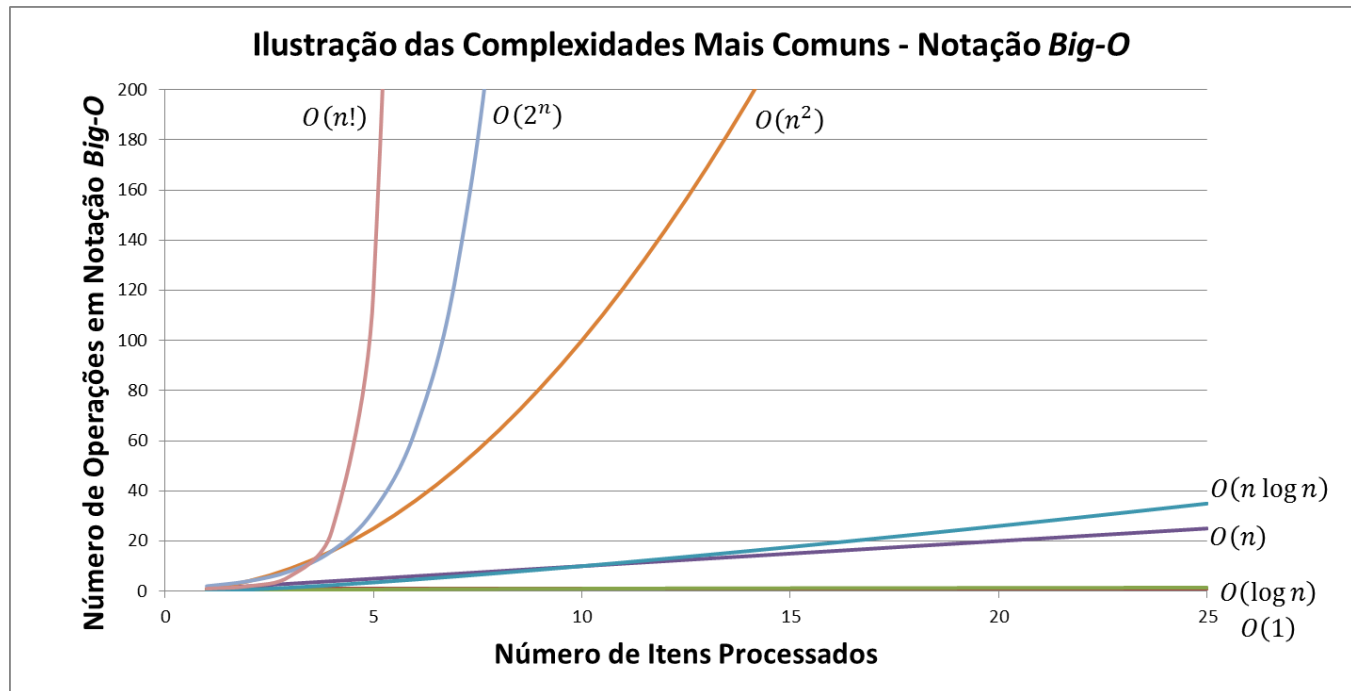
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 1

Qual o termo de maior grau da complexidade do algoritmo apresentados pelas funções abaixo:

a) $f(n) = 6n^3 + 5n^2 + 1$



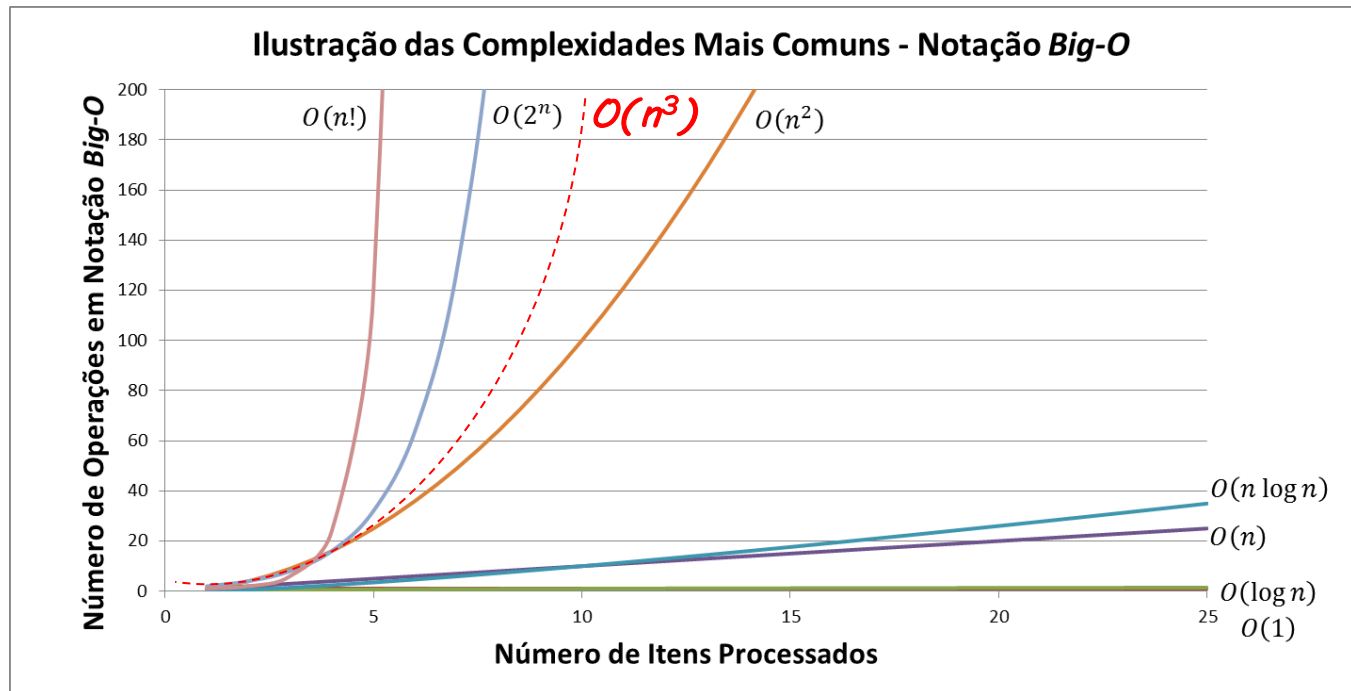
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 1

Qual o *termo de maior grau* da complexidade do algoritmo apresentados pelas funções abaixo:

a) $f(n) = 6n^3 + 5n^2 + 1$



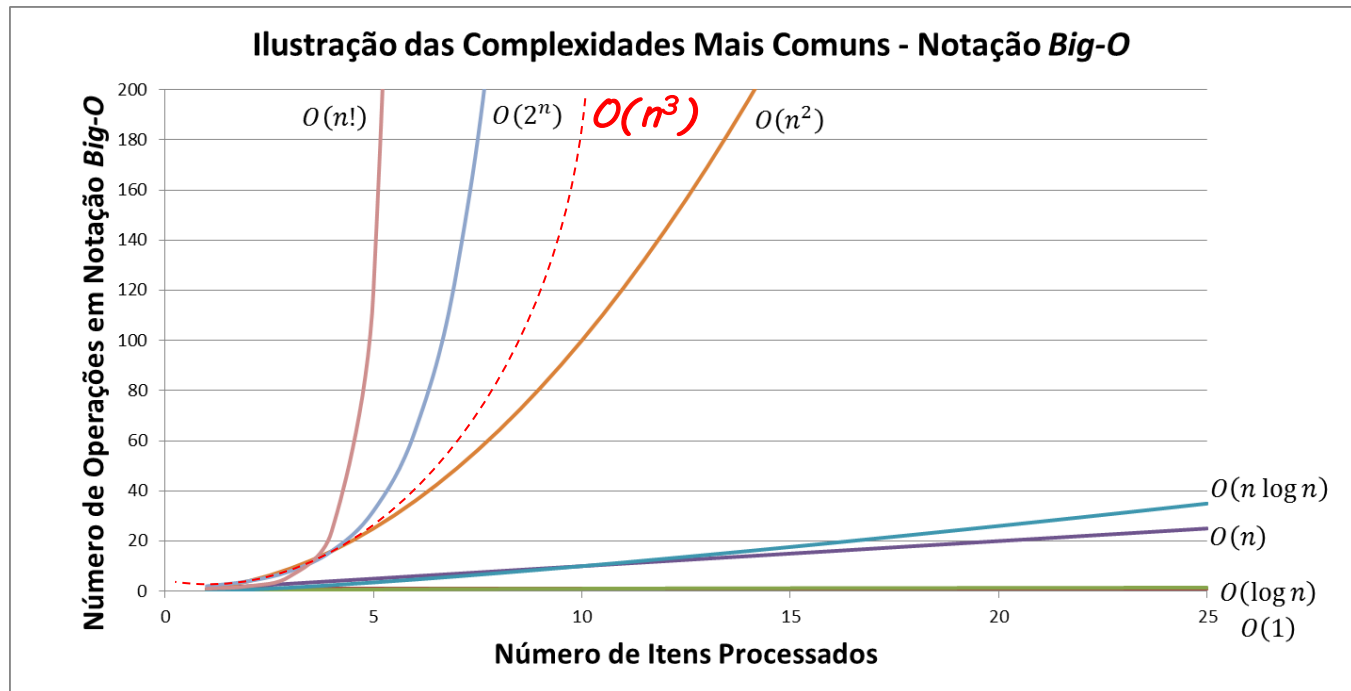
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 1

Qual o *termo de maior grau* da complexidade do algoritmo apresentados pelas funções abaixo:

$$a) f(n) = \boxed{6n^3} + 5n^2 + 1 = O(n^3)$$



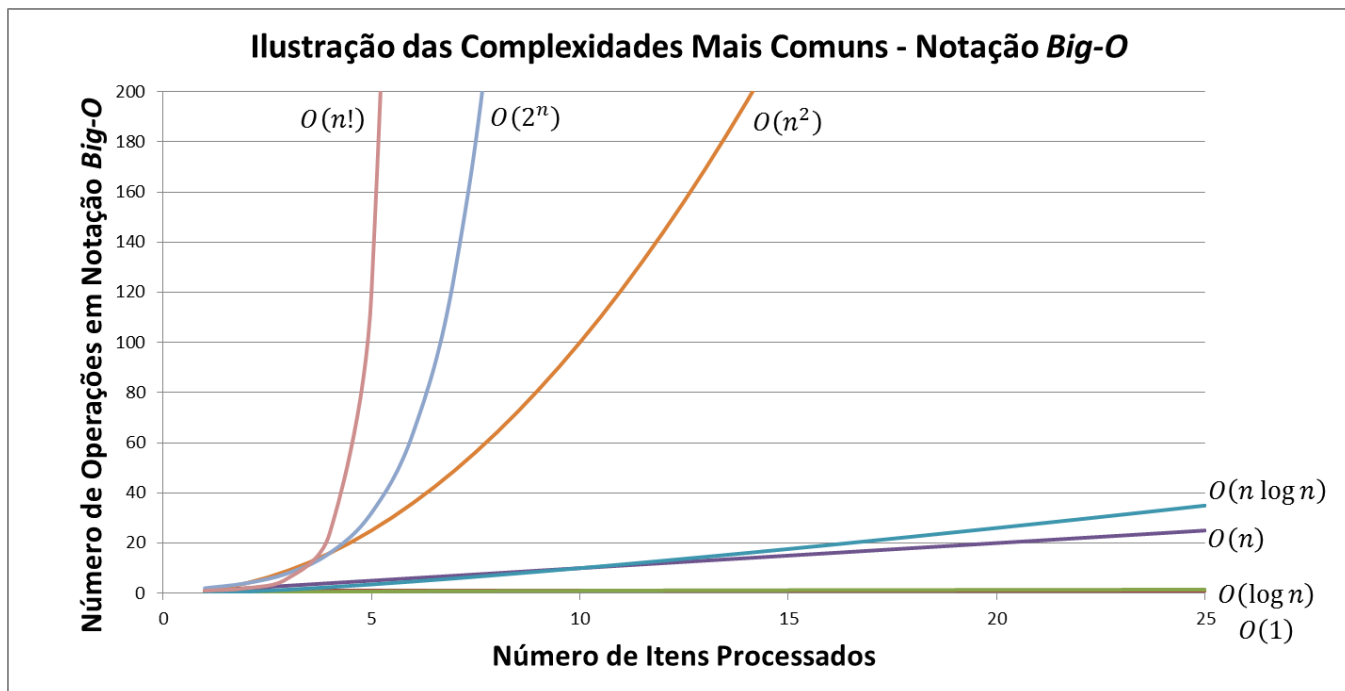
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 1

Qual o termo de maior grau da complexidade do algoritmo apresentados pelas funções abaixo:

b) $f(n) = (n + 1)^2$



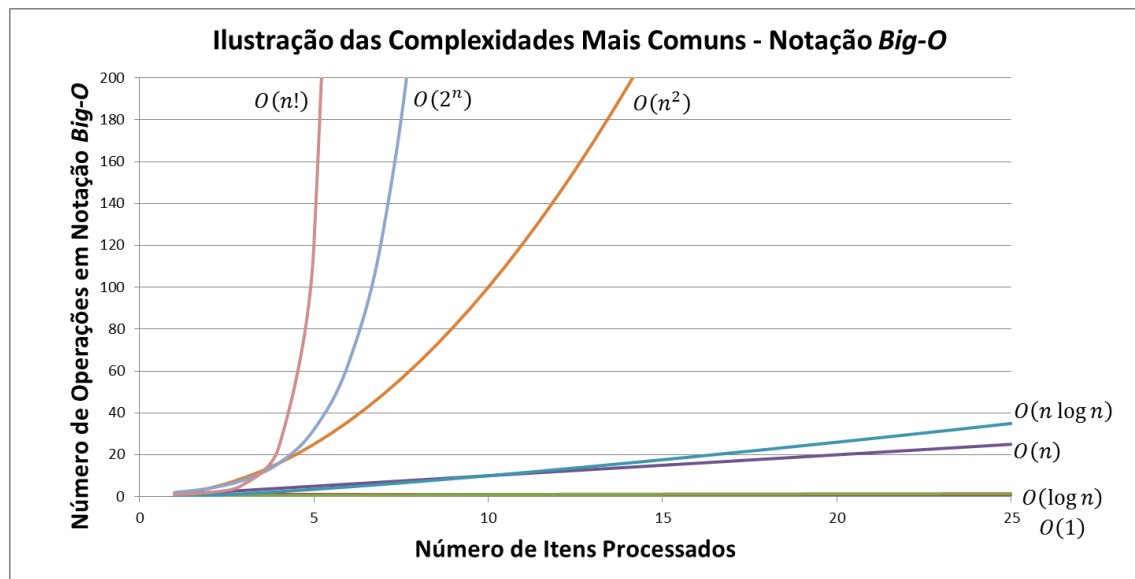
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 1

Qual o termo de maior grau da complexidade do algoritmo apresentados pelas funções abaixo:

$$\begin{aligned} b) f(n) &= (n + 1)^2 = (n + 1) \times (n + 1) \\ &= n^2 + n + n + 1 \\ &= n^2 + 2n + 1 \end{aligned}$$



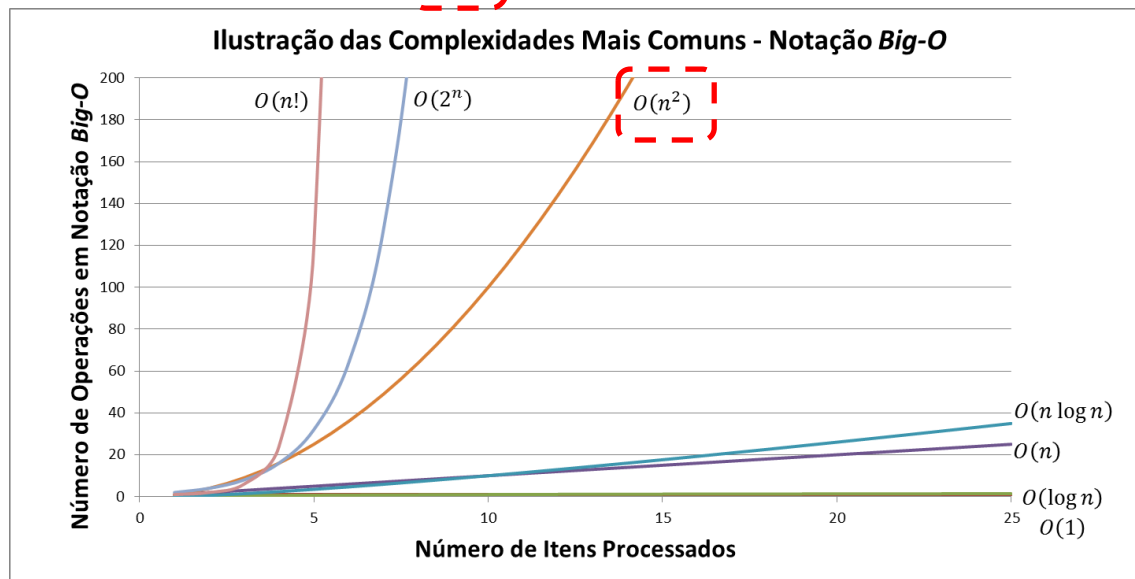
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 1

Qual *o termo de maior grau* da complexidade do algoritmo apresentados pelas funções abaixo:

$$\begin{aligned} b) f(n) &= (n + 1)^2 = (n + 1) \times (n + 1) \\ &= n^2 + n + n + 1 \\ &= \boxed{n^2} + 2n + 1 \end{aligned}$$



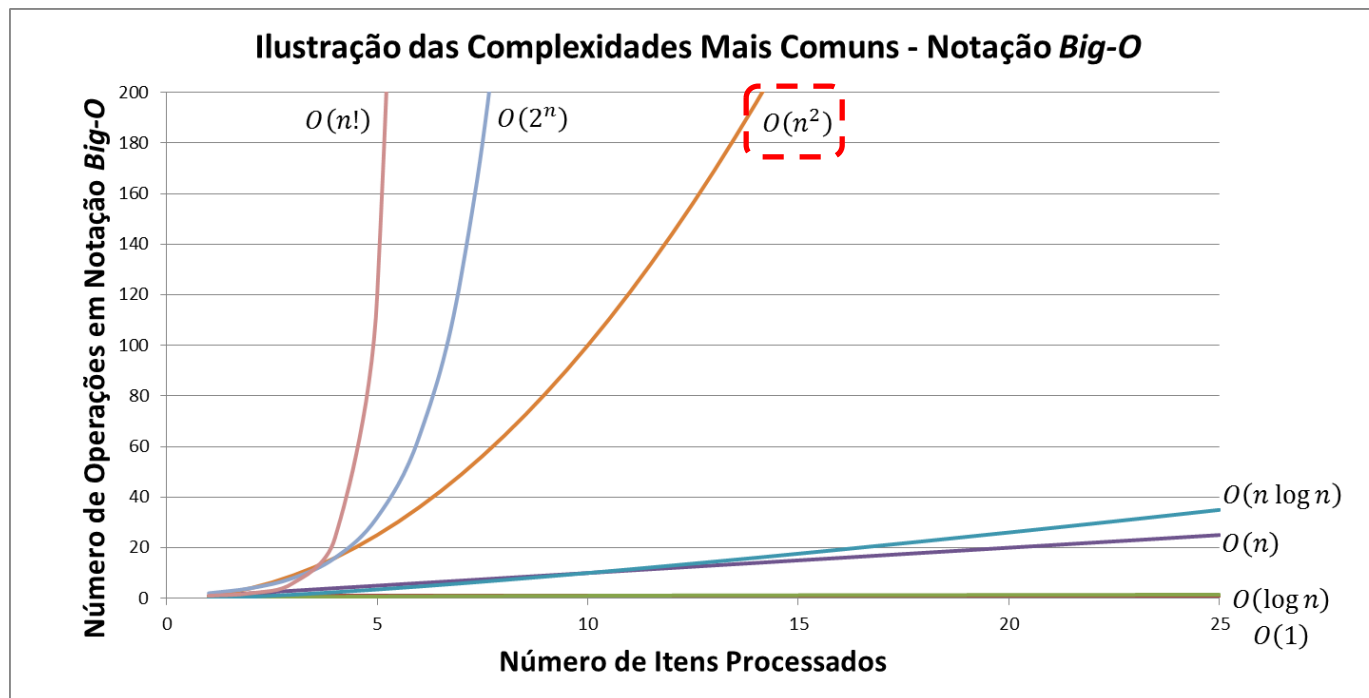
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 1

Qual *o termo de maior grau* da complexidade do algoritmo apresentados pelas funções abaixo:

b) $f(n) = (n + 1)^2 = O(n^2)$



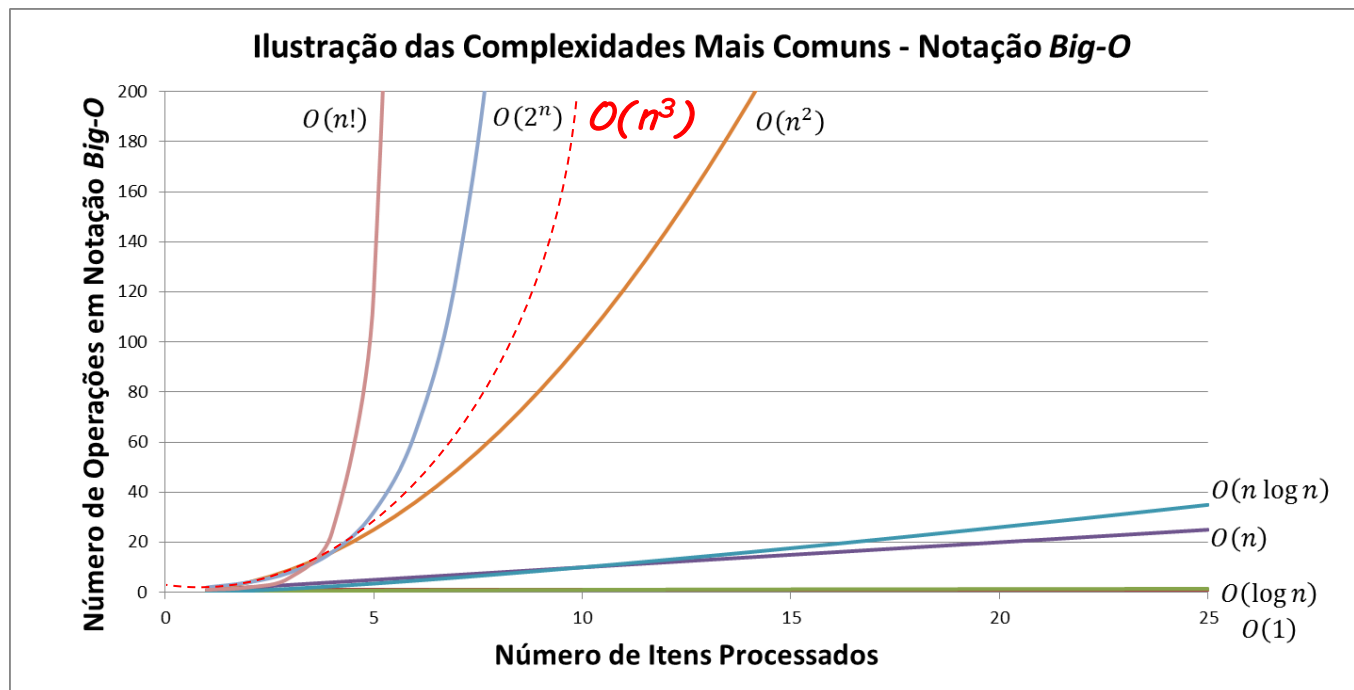
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize uma análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

a) $n^2 = O(n^3)$



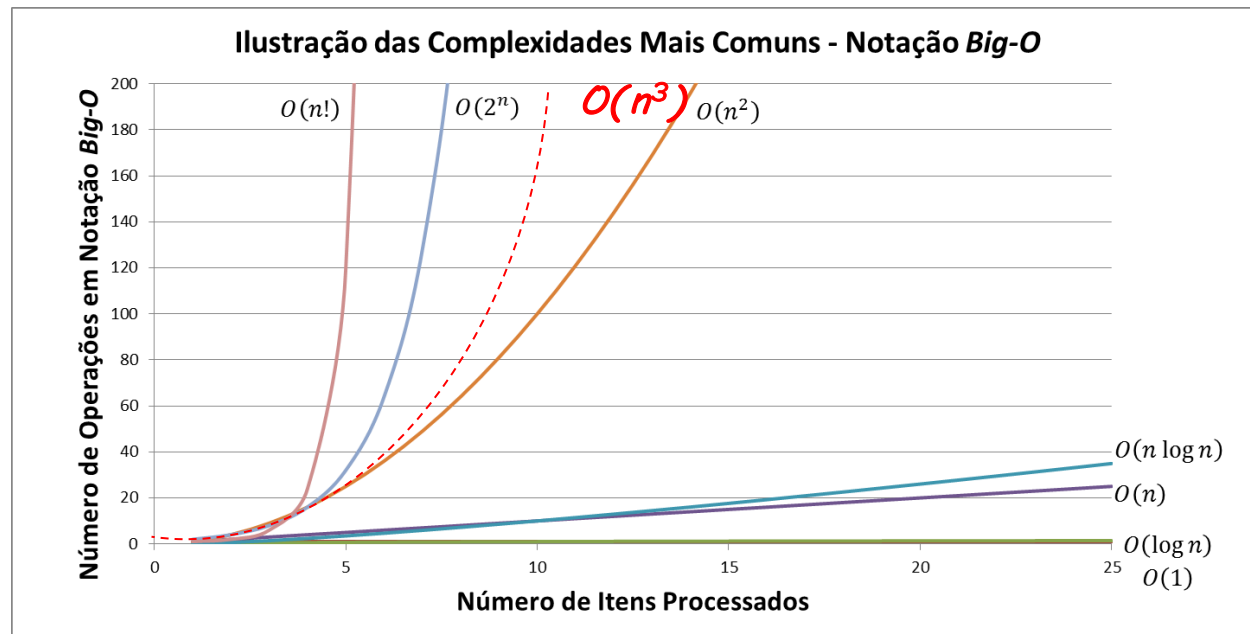
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

a) $n^2 = O(n^3)$ = **É verdadeira, pois n^3 é um limite superior para n^2 .**



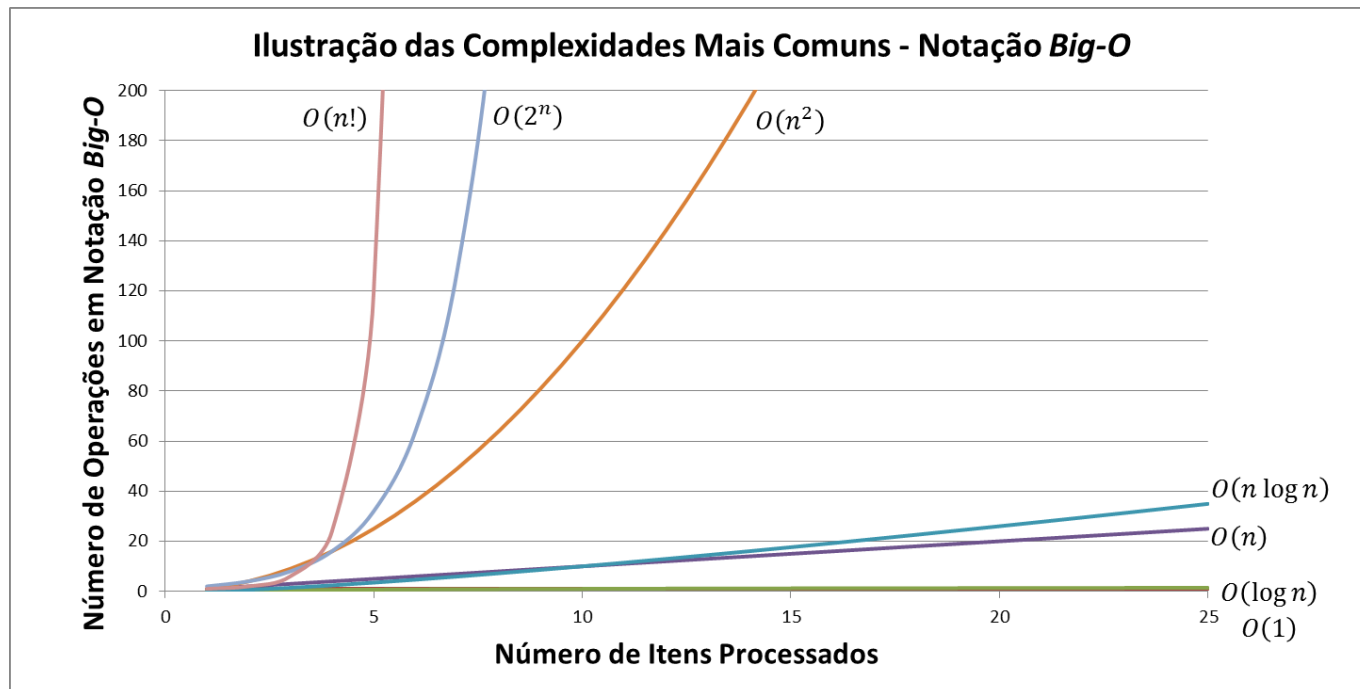
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade **é o limite superior do valor** que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

b) $2 * n + 1 = O(n^2)$



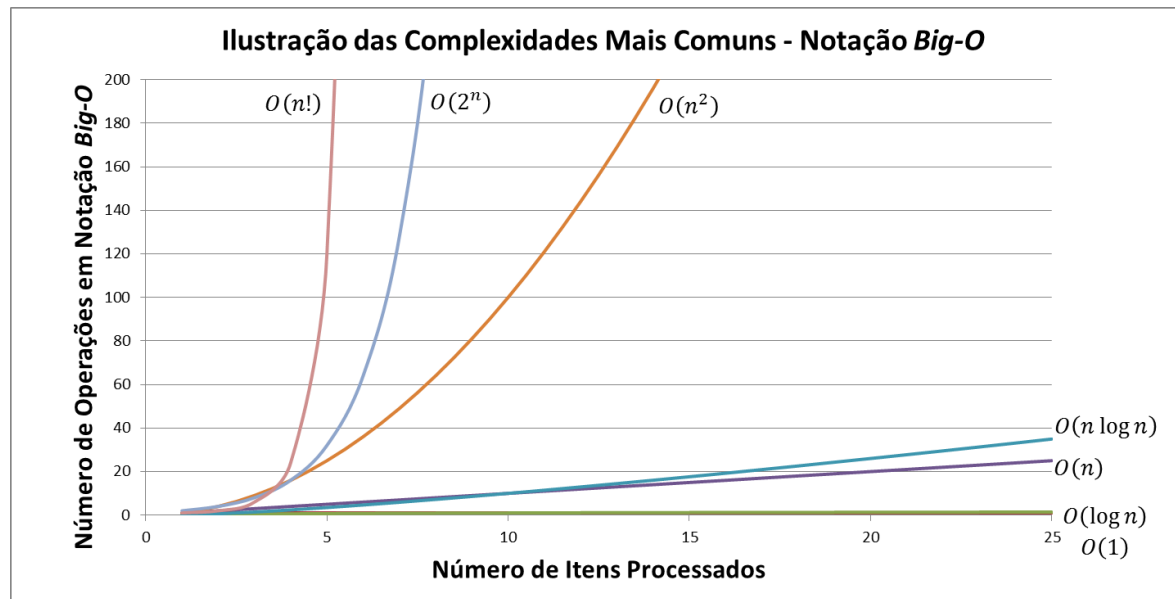
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

b) $2 * n + 1 = O(n^2)$ = **É verdadeira, pois n^2 é assintoticamente maior que $2 * n + 1$.**



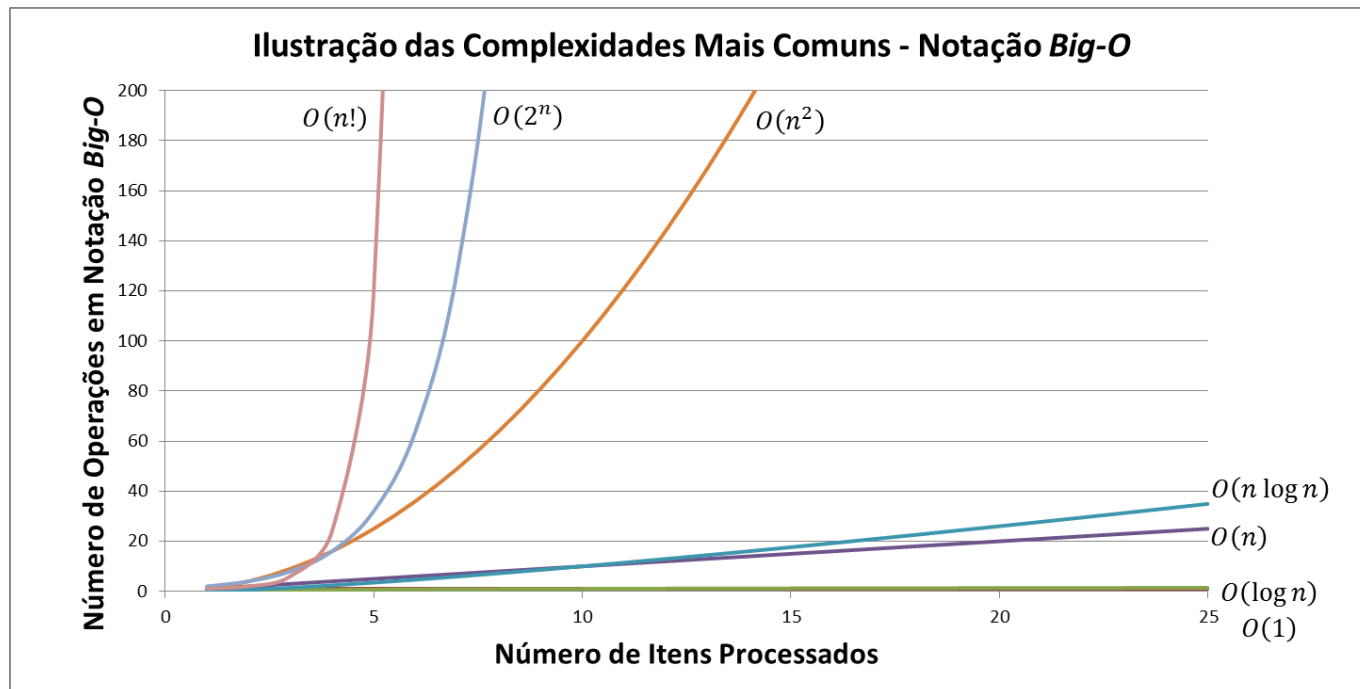
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade **é o limite superior do valor** que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

c) $n^3 = O(n^2)$



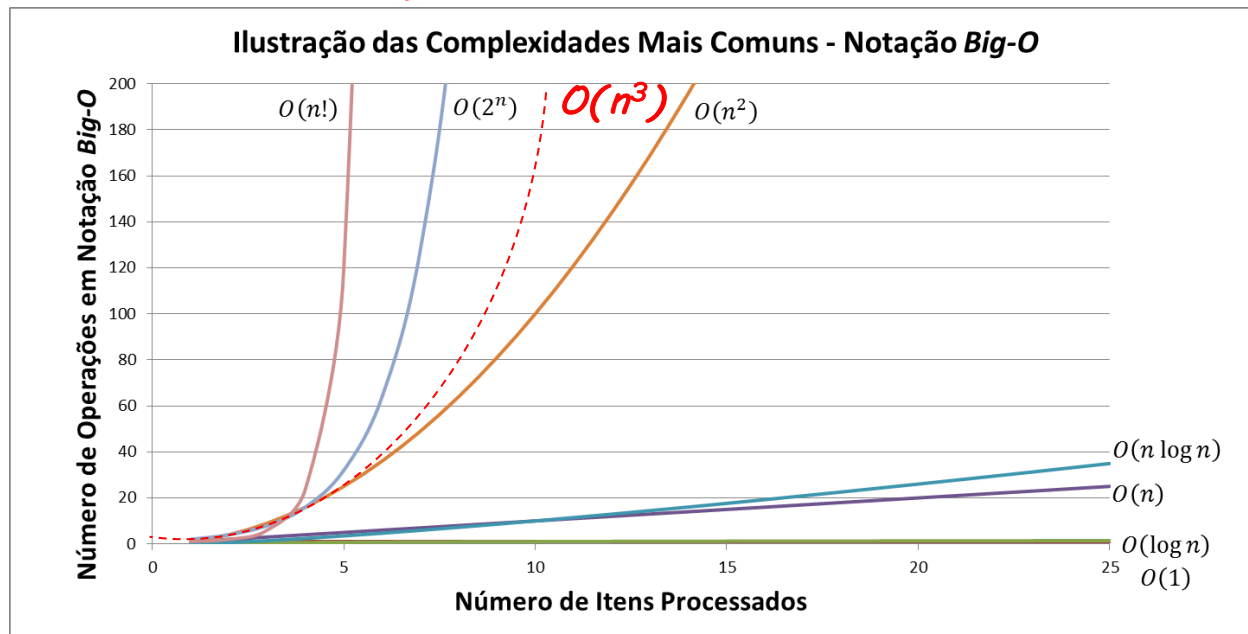
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

c) $n^3 = O(n^2)$ = É falsa, pois n^2 não é um limite superior para n^3 . Na prática, n^3 é assintoticamente maior que n^2 .



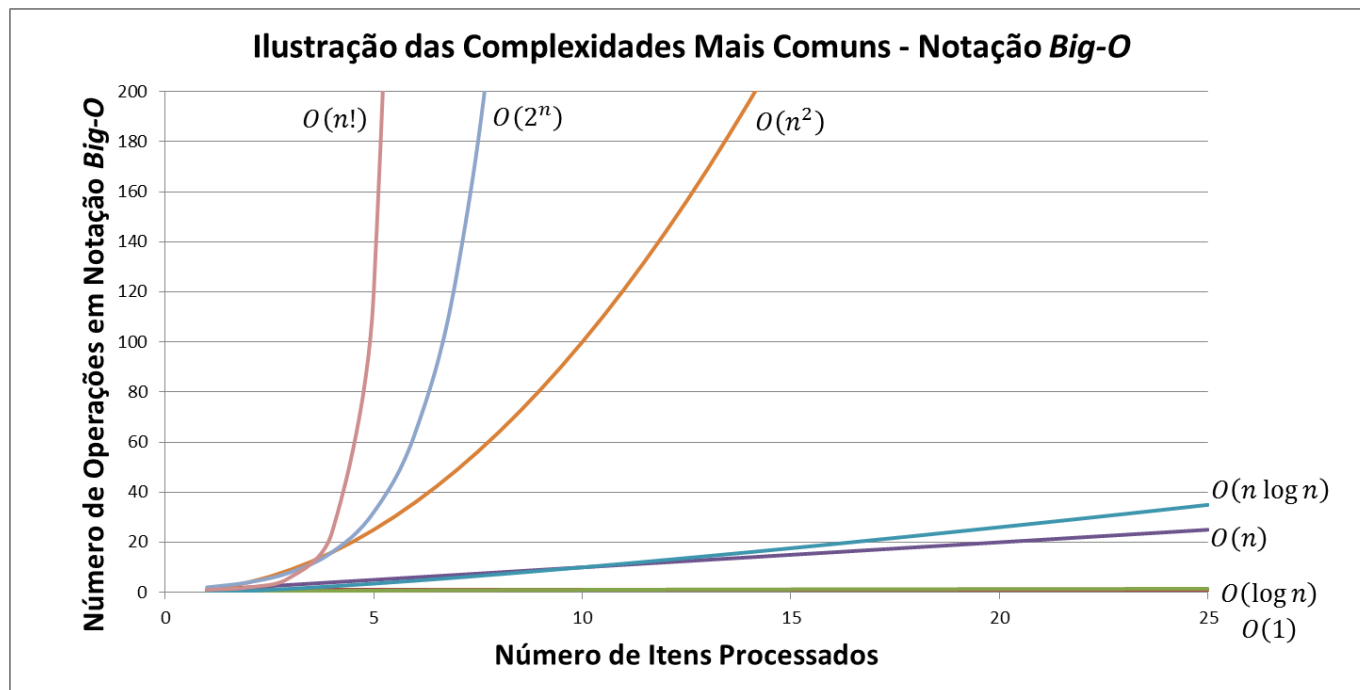
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

d) $3*n + 5*n\log(n) = O(n)$



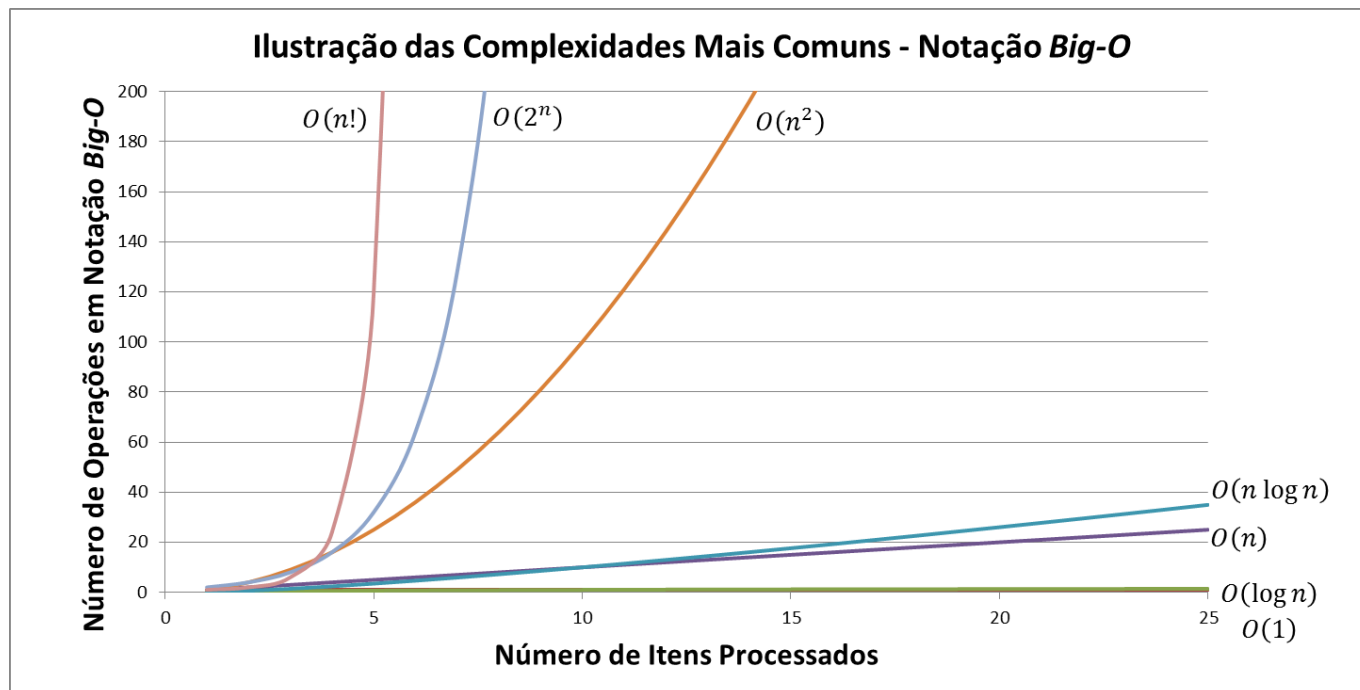
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

d) $3*n + 5*n\log(n) = O(n)$



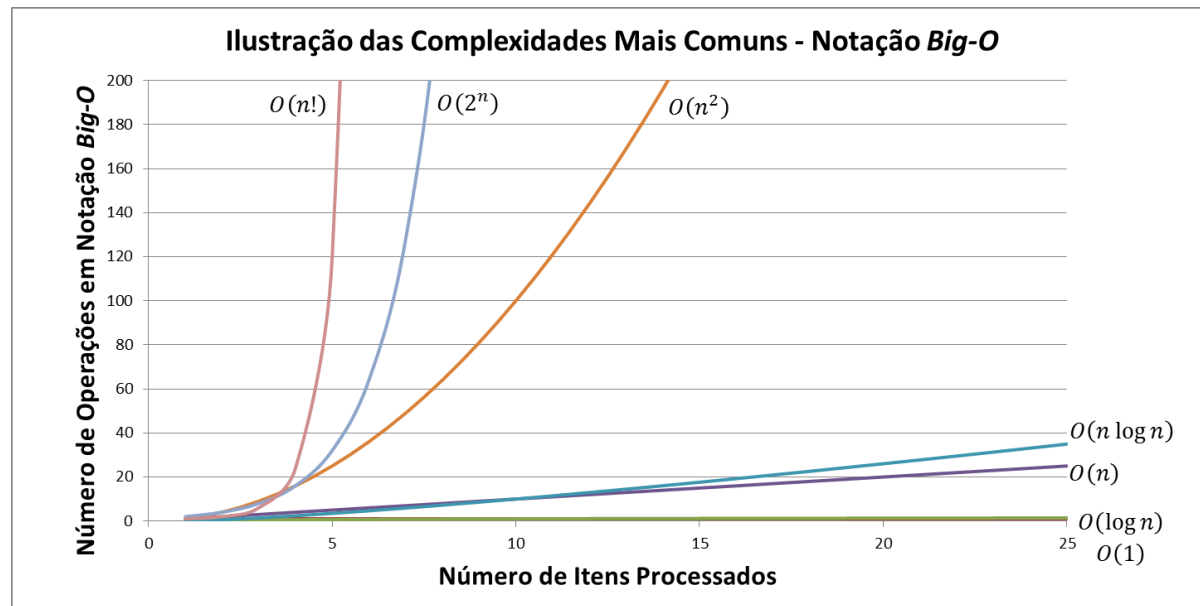
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

d) $3*n + 5*n\log(n) = O(n)$ = **É falsa, pois $n\log n$ é assintoticamente maior que n .**



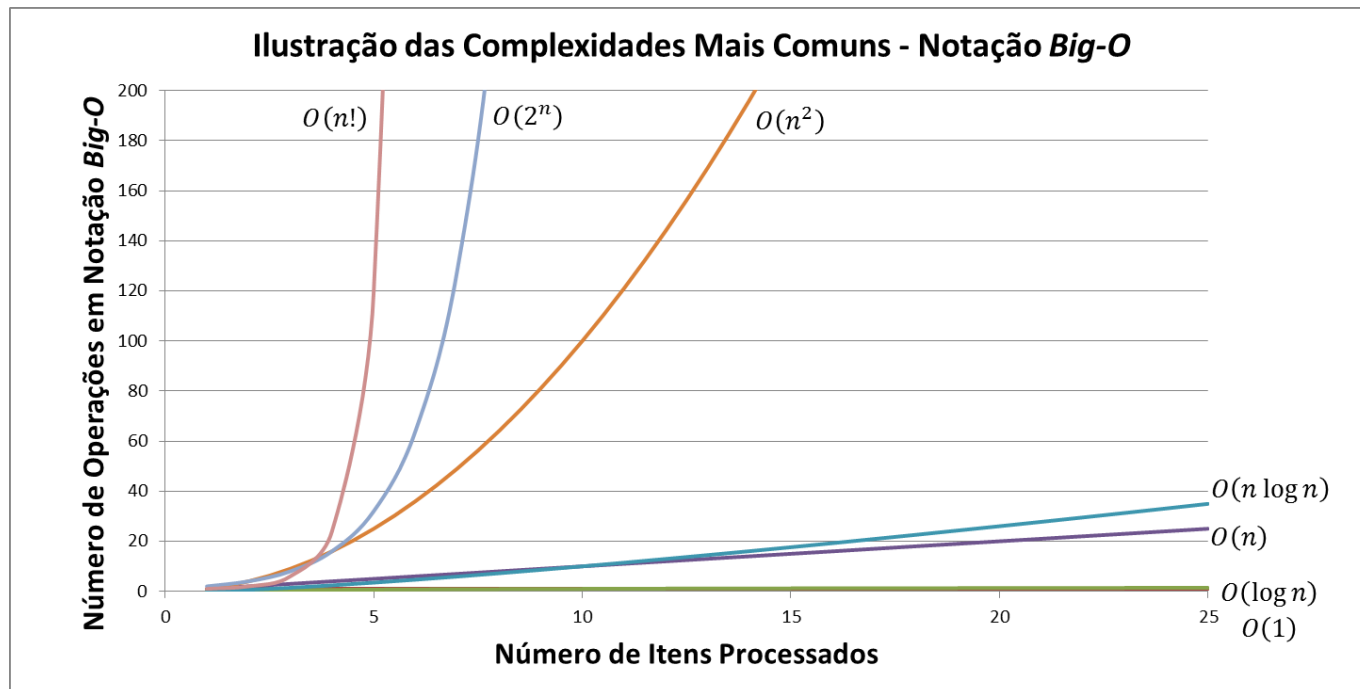
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

e) $\log(n) + \sqrt{n} = O(n)$



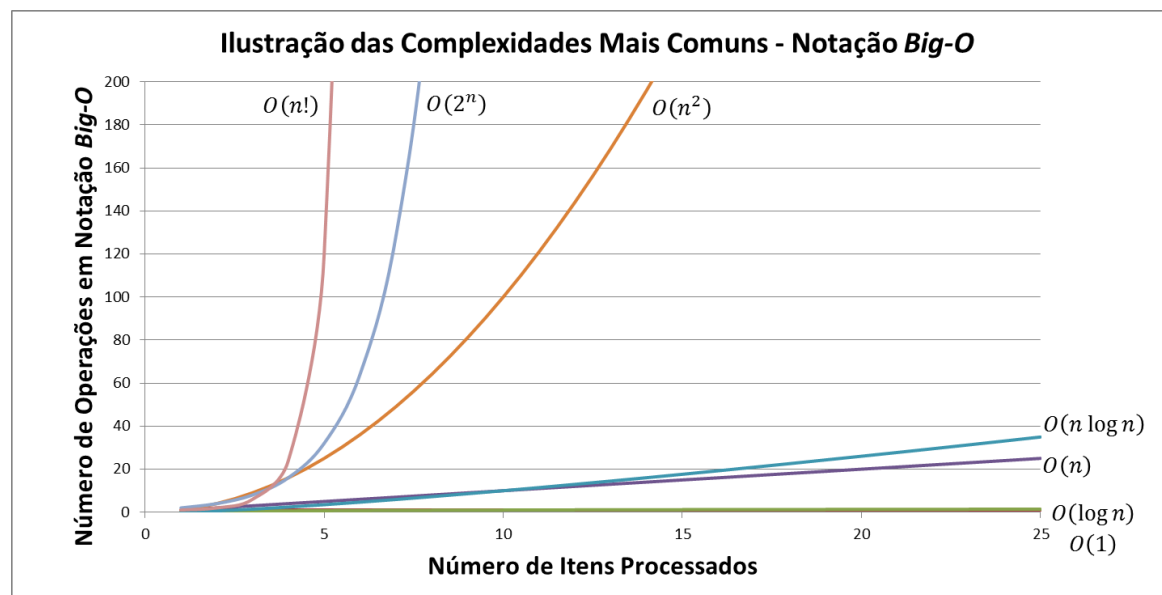
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

e) $\log(n) + \sqrt{n} = O(n)$ = É verdadeira, pois n é um limite superior para \sqrt{n} e para $\log(n)$.



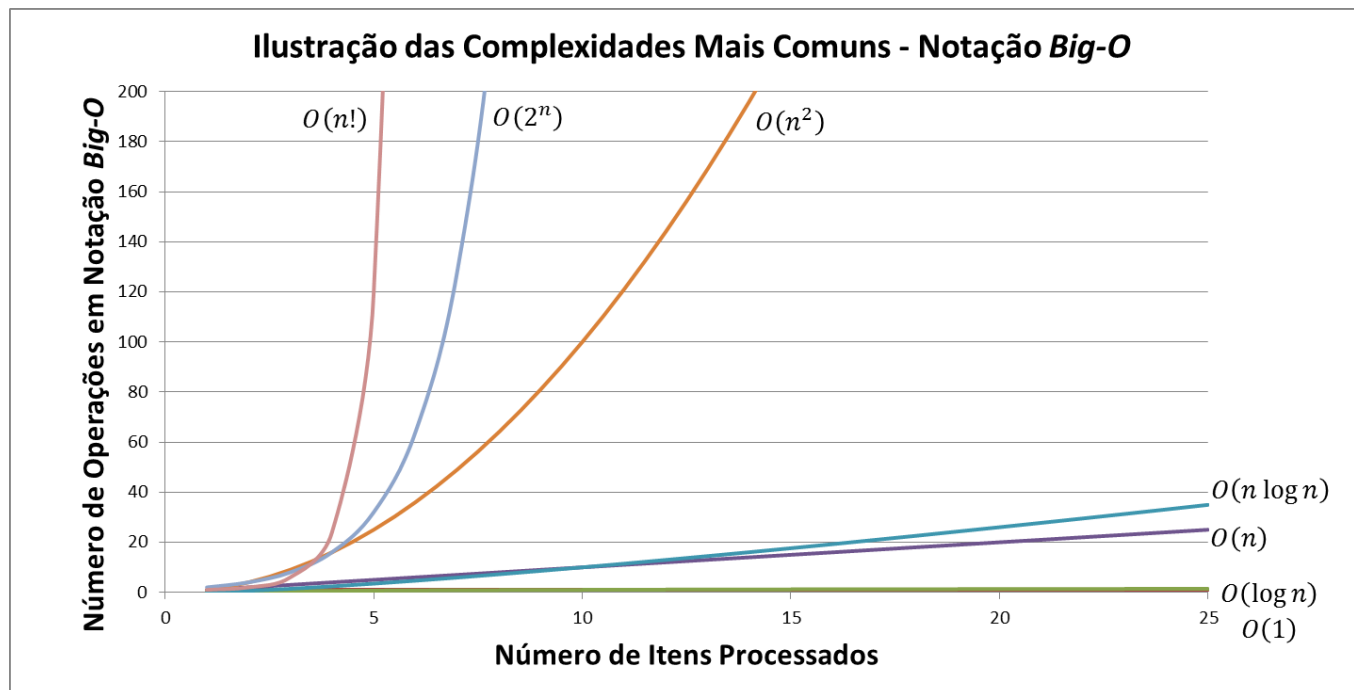
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

f) $\log n^2 = O(\log n)$



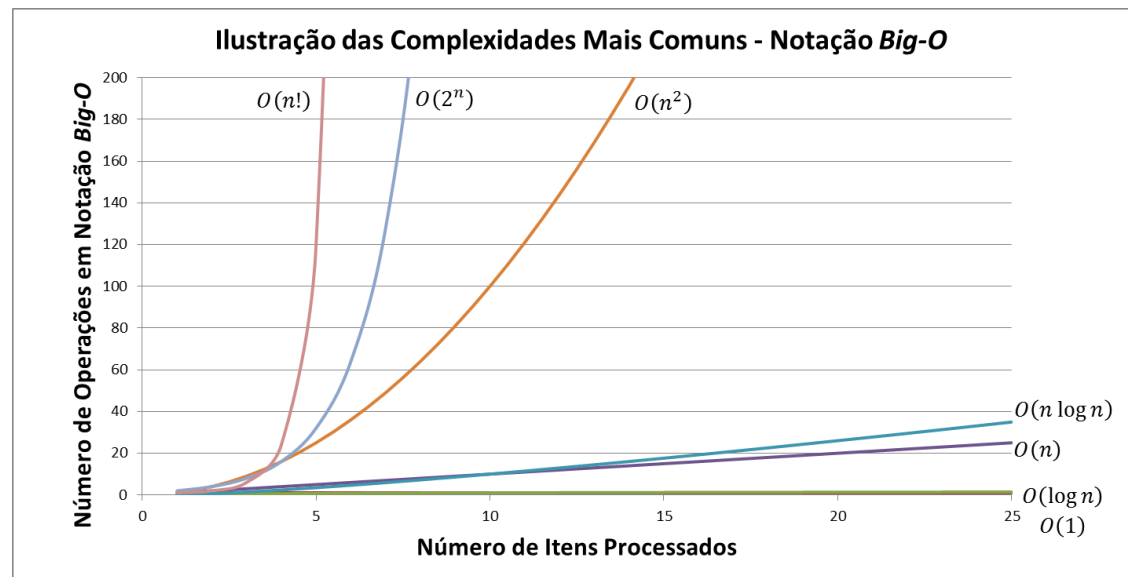
Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

f) $\log n^2 = O(\log n)$ = **É verdade, pois: $\log n^2$ é igual a $2(\log n) = O(\log n)$.**



Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade *é o limite superior do valor* que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

g) Se $f(n) = O(g(n))$ então pode dizer que:
 $g(n) = O(f(n))$

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

g) Se $f(n) = O(g(n))$ então pode dizer que:
 $g(n) = O(f(n))$

É falso. Quando afirmamos que $f(n) = O(g(n))$ significa que $g(n)$ representa um limite superior para $f(n)$. Portanto nem sempre temos $g(n) = O(f(n))$.

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 2

Realize um análise em relação as seguintes igualdades (se o que está determinado na direita da igualdade é o limite superior do valor que se encontra a esquerda) e diga se é verdadeiro (é o limite superior) ou falso (não é o limite superior).

g) Se $f(n) = O(g(n))$ então pode dizer que:
 $g(n) = O(f(n))$

Observe o exemplo:

$$f(n) = n^2 \text{ e } g(n) = n^3$$

Dessa forma, teríamos $n^2 = O(n^3)$, que é VERDADE.

Porém, $n^3 = O(n^2)$, seria FALSO.

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 3

Um algoritmo tem complexidade:

$$O(6m^4 + 10mn^3 + 2n^2 + 90m)$$

Escreva uma maneira simplificada de representar a complexidade desse algoritmo.

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 3

Um algoritmo tem complexidade:

$$O(6m^4 + 10mn^3 + 2n^2 + 90m)$$

Escreva uma maneira simplificada de representar a complexidade desse algoritmo.

RESPOSTA:

$$~~O(6m^4 + 10mn^3 + 2n^2 + 90m)~~$$

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 3

Um algoritmo tem complexidade:

$$O(6m^4 + 10mn^3 + 2n^2 + 90m)$$

Escreva uma maneira simplificada de representar a complexidade desse algoritmo.

RESPOSTA:

$$O(m^4 + mn^3 + n^2 + m)$$

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 3

Um algoritmo tem complexidade:

$$O(6m^4 + 10mn^3 + 2n^2 + 90m)$$

Escreva uma maneira simplificada de representar a complexidade desse algoritmo.

RESPOSTA:

$$O(m^4 + mn^3 + n^2 + m)$$

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 3

Um algoritmo tem complexidade:

$$O(6m^4 + 10mn^3 + 2n^2 + 90m)$$

Escreva uma maneira simplificada de representar a complexidade desse algoritmo.

RESPOSTA:

$$O(m^4 + mn^3)$$

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 4

Construa um código que trabalhe com uma função recursiva que tenha como objetivo incrementar a variável valor de um em um até o limite estabelecido na condição “if” ($\text{valor} < 10$), e retornando o resultado final desta operação. Sendo a variável valor iniciando em 0.

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 4

```
#include <iostream>
using namespace std;

int incrementa(int valor);

int main(){
    int acumular = 0;
    incrementa(acumular);
    return 0;
}

int incrementa(int valor){
    if (valor < 10){
        valor++; // vou andar com a variável valor de 1 em 1.
        return incrementa(valor);
    } else {
        cout << "O valor final acumulado: " << valor;
    }
}
```


Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 4

Outra forma de se fazer!

```
#include <iostream>
using namespace std;

void incrementa(int &valor);

int main(){
    int acumula = 0;
    incrementa(acumula);
    cout << "O valor acumulado: " << acumula;
    return 0;
}

void incrementa(int &valor){
    if (valor < 10){
        valor++; // vou andar com a variável valor de 1 em 1.
        incrementa(valor);
    }
}
```

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 5

A seguir temos uma função recursiva em pseudocódigo. Converta para C++, estruturando com a função principal e colocando o protótipo da função, e identifique quem são o(s) caso(s) base e o caso geral.

```
int fazAlgo(int n)
início
se n=1 então
    retorne (1);
senão se n = 0 então
    retorne(0);
senão
    retorne(fazAlgo(n-2));
fim
```

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 5

```
#include <iostream>
using namespace std;

int fazAlgo(int n);

int main(){
    int n = 0;
    fazAlgo(n);
    cout << fazAlgo(n);
    return 0;
}
```

```
int fazAlgo(int n) {
    if (n==1) {
        return (1);
    } else {
        if (n==0) {
            return(0);
        } else {
            return(fazAlgo(n-2));
        }
    }
}
```

Caso base

Caso Geral

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 6

Em determinada funcionalidade de um software, é necessário realizar a ordenação de um conjunto formado por muitos números inteiros. Que algoritmo de ordenação oferece melhor complexidade de tempo (Big O) no pior caso: (Merge sort ?, Insertion sort ?, Bubble sort ? ou Selection sort ?). Explique por que.

Aula 9 - ALGORITMOS E COMPLEXIDADE

EXERCÍCIOS

EXERCÍCIO 6

Em determinada funcionalidade de um software, é necessário realizar a ordenação de um conjunto formado por muitos números inteiros. Que algoritmo de ordenação oferece melhor complexidade de tempo (Big O) no pior caso: (Merge sort ?, Insertion sort ?, Bubble sort ? ou Selection sort ?). Explique por que.

Merge sort. Independentemente se os elementos estão ordenados OU não, a complexidade é sempre a mesma.

Melhor caso: $O(n \log n)$ - Pior caso: $O(n \log n)$

- As outras ordenações citadas tem complexidade é $O(n^2)$.*