

INSTITUTO FEDERAL
SANTA CATARINA

Introdução a Complexidade de Algoritmos

Estruturas de Dados

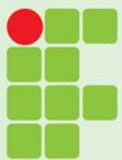
Prof. Vilson Heck Junior

105
ANOS

REDE FEDERAL
DE EDUCAÇÃO
PROFISSIONAL
E TECNOLÓGICA
1909-2014

Apresentação

- Revisão - O Algoritmo;
- A Complexidade;
- Exercício.



INSTITUTO FEDERAL
SANTA CATARINA



Complexidade de Algoritmos

REVISÃO - O ALGORITMO

105
ANOS

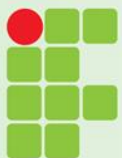
**REDE FEDERAL
DE EDUCAÇÃO
PROFISSIONAL
E TECNOLÓGICA**
1909-2014

O Algoritmo

- Os algoritmos são o cerne da computação;
- Programas de computador são criados a partir de algoritmos;
- Mas o que são algoritmos?
 - São sequências de passos bem definidos, detalhados e finitos, que quando executados realizam uma tarefa específica.

O Algoritmo

- Por ser finito, um algoritmo deve:
 - Iniciar,
 - Processar por um determinado tempo e
 - Terminar;
- Em geral, tendemos a comparar dois algoritmos distintos, que realizam uma mesma tarefa, pelo tempo necessário à sua execução.



INSTITUTO FEDERAL
SANTA CATARINA



Complexidade de Algoritmos

A COMPLEXIDADE

105
ANOS

**REDE FEDERAL
DE EDUCAÇÃO
PROFISSIONAL
E TECNOLÓGICA**
1909-2014

A Complexidade

- Diferentes computadores, muitas vezes com hardwares idênticos, podem levar tempos diferentes para processar um mesmo algoritmo;
 - Quando trocamos algum, ou mais, itens de hardware, esta diferença tende a aumentar ainda mais;
- Por tanto, avaliar o desempenho de um algoritmo com base apenas no tempo de execução deste, é ineficiente.

A Complexidade

- Surge o conceito de Complexidade de Algoritmos;
 - Que é uma fração da Complexidade Computacional;
- A Complexidade de Algoritmos estuda e define quanto eficiente é um algoritmo em relação ao número de operações (passos do algoritmo) necessárias para finalizar a tarefa.

A Complexidade

- Apesar do *hardware* influenciar no tempo de execução de um algoritmo, existe ainda um segundo parâmetro que pode também influenciar este tempo:
 - O Conjunto de Dados!

A Complexidade

- Em geral, algoritmos servem para processar conjuntos de dados com tamanho indeterminado:
 - Listas;
 - Filas;
 - Pilhas;
 - Tabelas;
 - Imagens;
 - Entre outros.

A Complexidade

- Exemplo:
 - Para somar todos os elementos de uma lista:
 - Quando a lista tiver apenas 1 elemento:
 - Uma operação de soma!
 - Quanto a lista tiver 1.000.000 de elementos:
 - 1.000.000 de somas!

A Complexidade

- Por padrão, para denotar o tamanho do conjunto de dados a ser processado, chamamos este número de dados de:

n

A Complexidade

- Criamos, então, uma “função de complexidade do algoritmo”, que irá relacionar o número de instruções utilizadas por um algoritmo – **$T(n)$** – com o tamanho do conjunto de dados – **n** :
 - **$T(n) = n$**
- Neste exemplo, para cada **n** dados, teremos **n** operações.

A Complexidade

- As Complexidades mais comuns são:

$$\log_2 n$$

$$n$$

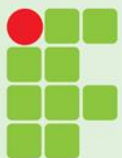
$$n \log_2 n$$

$$n^2$$

$$n^3$$

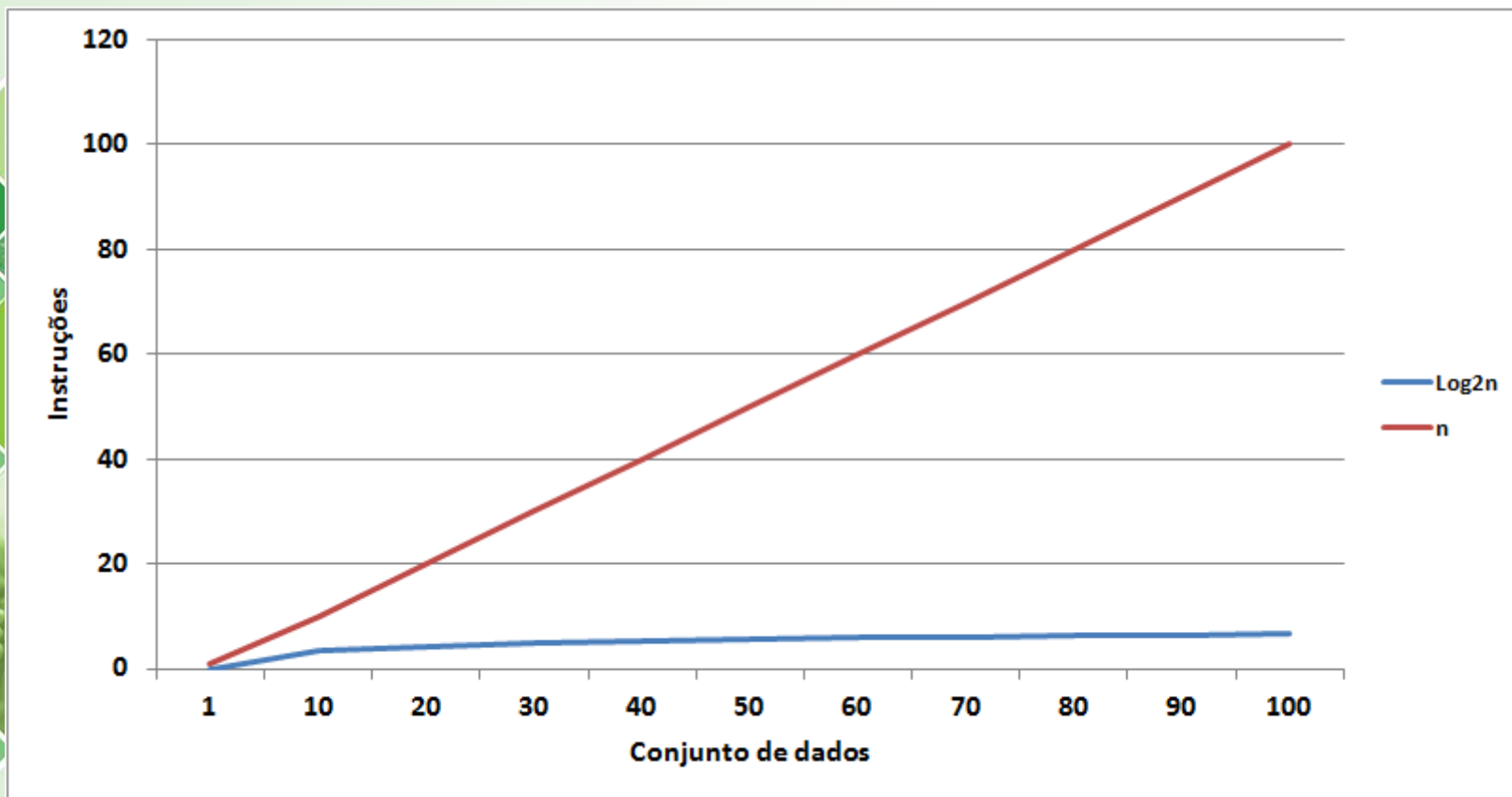
$$2^n$$

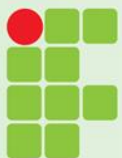
$$3^n$$



INSTITUTO FEDERAL
SANTA CATARINA

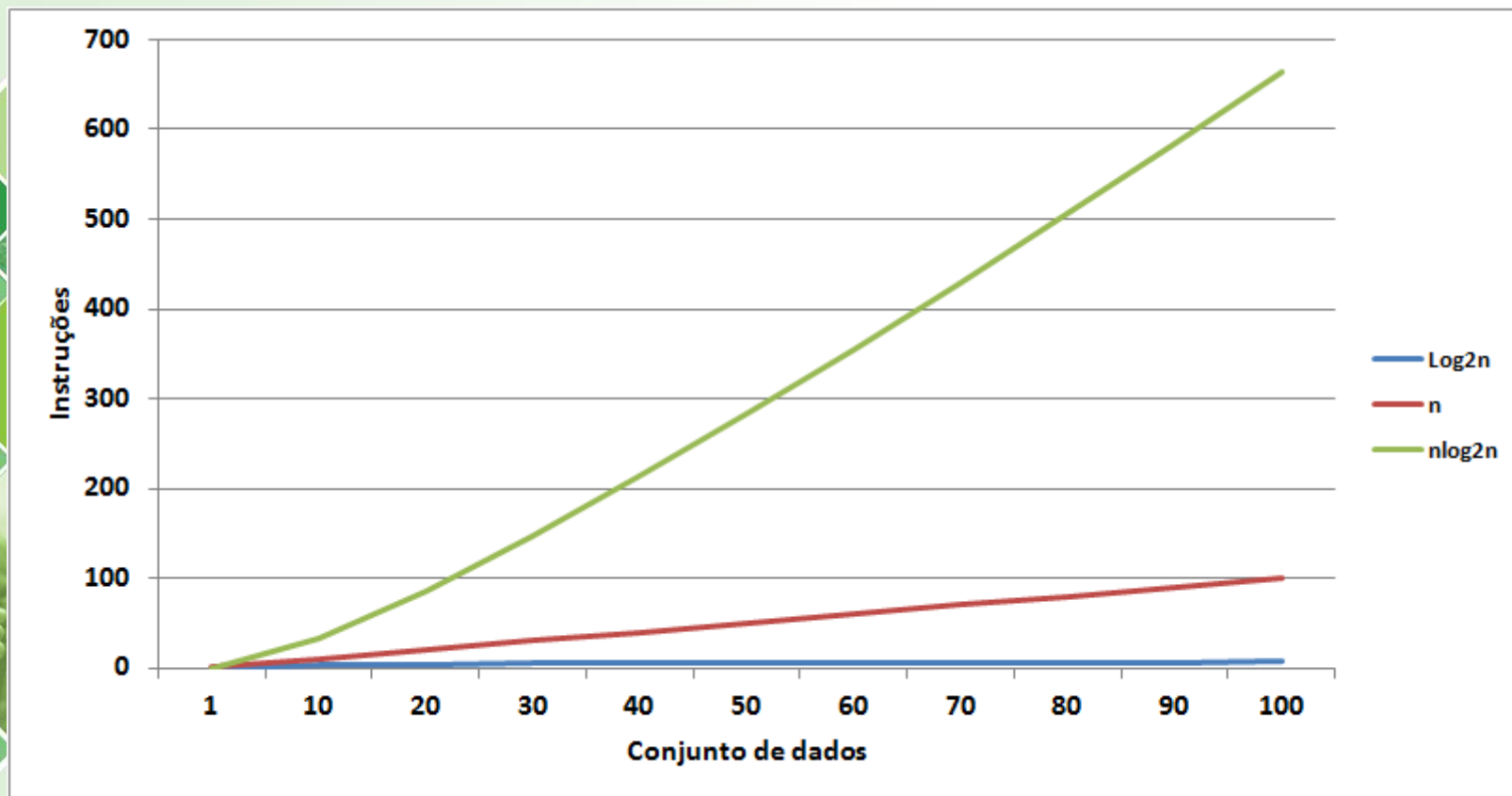
A Complexidade

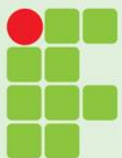




INSTITUTO FEDERAL
SANTA CATARINA

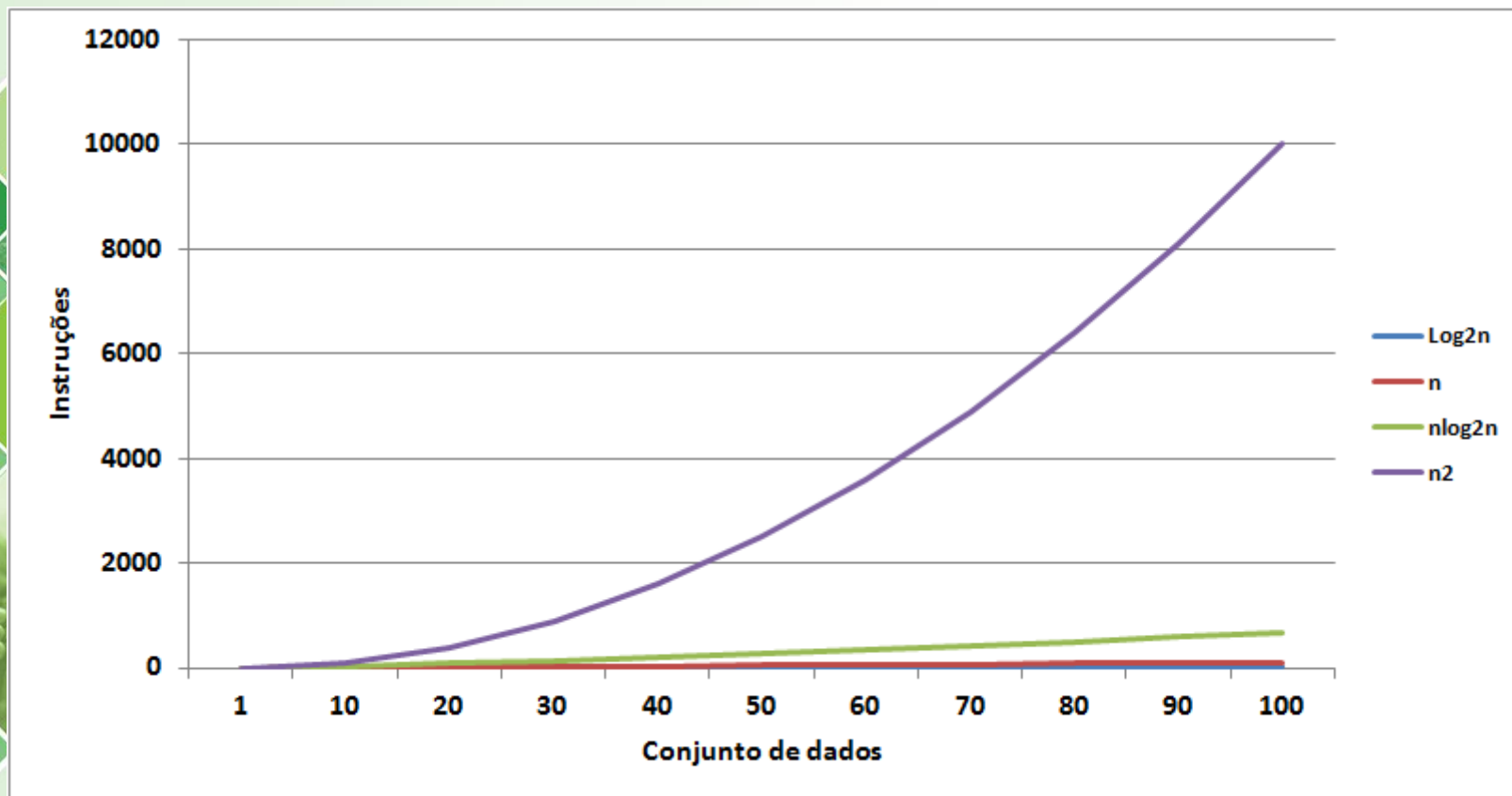
A Complexidade

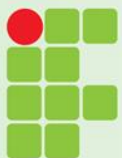




INSTITUTO FEDERAL
SANTA CATARINA

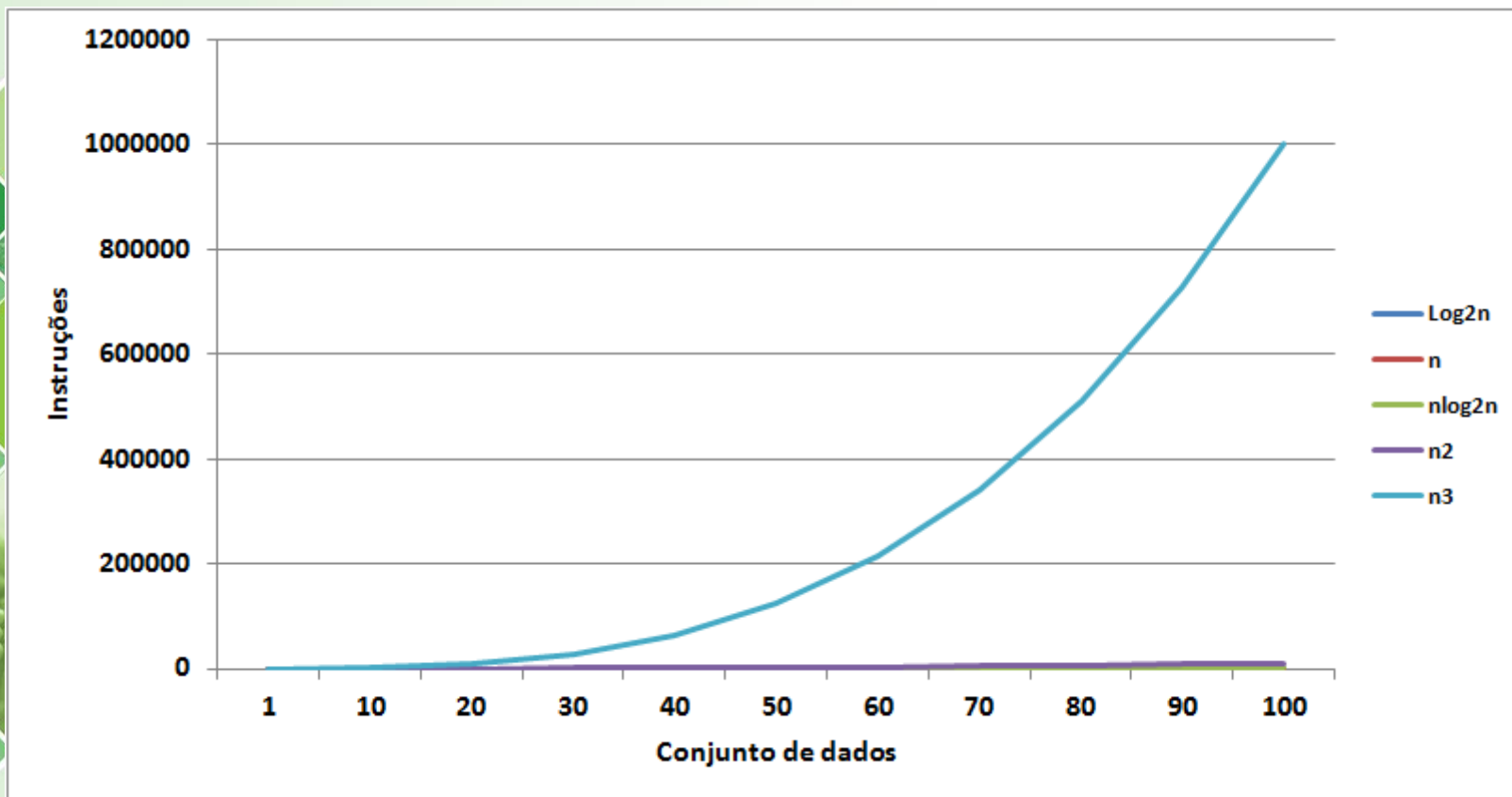
A Complexidade

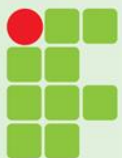




INSTITUTO FEDERAL
SANTA CATARINA

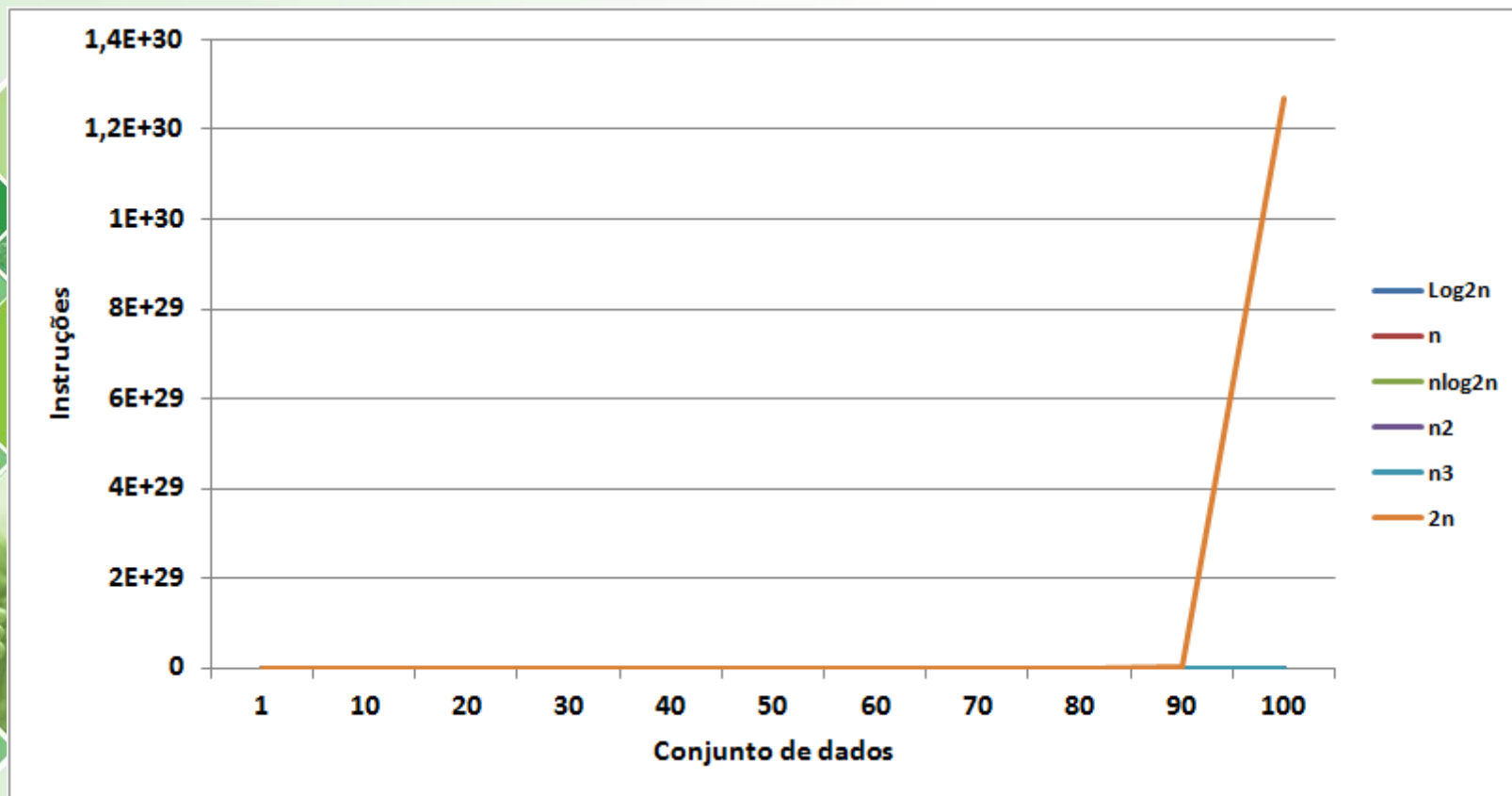
A Complexidade

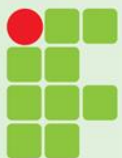




INSTITUTO FEDERAL
SANTA CATARINA

A Complexidade





INSTITUTO FEDERAL
SANTA CATARINA

A Complexidade



A Complexidade

- Pergunta 01:
 - Qual é o $T(n)$ para a pesquisa de um elemento em uma lista?
- Pergunta 02:
 - Qual é o $T(n)$ para inserir um elemento em uma lista?

A Complexidade

- Temos respostas diferentes para as perguntas anteriores:
 - Pergunta 01 – 3 cenários existentes:
 - O primeiro elemento ser o procurado (melhor caso);
 - O elemento central ser o procurado (caso médio);
 - O último elemento ser o procurado (pior caso);
 - Pergunta 02 – os 3 cenários são iguais:
 - Uma instrução para criar o próximo elemento;
 - Encadear o elemento!

A Complexidade

- Cenários possíveis:
 - Melhor Caso:
 - Descreve o **menor** número de instruções possíveis de ocorrer para um determinado algoritmo (conforme os dados utilizados);
 - No exemplo de pesquisa em uma lista, o melhor cenário existe quando o primeiro elemento é o procurado;
 - Neste caso, $T(n) = 1$, utiliza-se a notação:

$$\Omega(n) = 1$$

A Complexidade

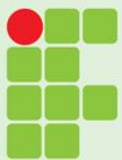
- Cenários possíveis:
 - Caso Médio:
 - Descreve o cenário mais comum, o cenário **médio**, de número de instruções;
 - No exemplo da lista, na média os valores procurados estarão no meio da lista. Algumas vezes estarão no início, porém outra vez estarão no fim, isto torna nosso valor médio sendo $T(n) = n / 2$, utilizando a notação:

$$\theta(n) = \frac{n}{2}$$

A Complexidade

- Cenários possíveis:
 - Pior Caso:
 - Descreve o **maior** número de instruções possíveis de ocorrer para um determinado algoritmo (conforme os dados utilizados);
 - No exemplo da lista, o valor pesquisado se encontraria no fim e, por tanto, precisaríamos de $T(n) = n$, com a notação:

$$O(n) = n$$



A Complexidade

$$\log_2 n$$

$$n$$

$$n \log_2 n$$

$$n^2$$

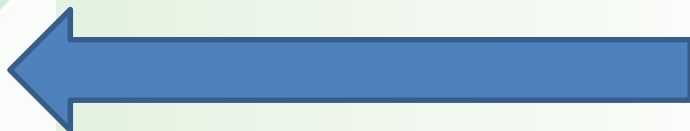
$$n^3$$

$$2^n$$

$$3^n$$

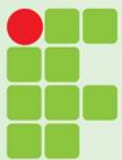
- Representações comuns:

- O cenário utilizado para representar um algoritmo é o de pior caso;
- Apesar de podermos quantificar as instruções necessárias para cada algoritmo, para o pior caso, o mais comum é *arredondarmos* para uma das complexidade mais comuns:



A Complexidade

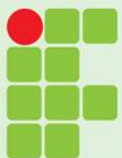
- Ao medir o número de instruções necessárias para cada algoritmo, devem ser consideradas estruturas, tais como de decisão ou de repetição;
- Em geral, a estratégia de um algoritmo é responsável pela sua eficiência, mas vale lembrar que estruturas de repetição aninhadas são as maiores responsáveis pelo aumento de instruções;



INSTITUTO FEDERAL
SANTA CATARINA

A Complexidade

Complexidade	Tempo para executar N = 100, em um Core i7 980-X Extreme Edition
$\log_2 n$	133 ns
n	2 μs
$n \log_2 n$	13 μs
n^2	200 μs
n^3	20 ms
2^n	815.104.552.615.888 anos!
3^n	331.389.866.725.830.000 000.000.000.000.000 anos!



INSTITUTO FEDERAL
SANTA CATARINA

Exercício

- Identifique quais são as complexidades computacionais do seu algoritmo de ordenação de dados. Lembre, você deverá explicar estas complexidades aplicadas em seu algoritmo:
 - Cenário de pior caso;
 - Cenário de caso médio;
 - Cenário de melhor caso;