

## **Resumo - N e NP**

**Rithie Natan Carvalhaes Prado**

Curso de Bacharelado em Ciências da Computação - Pontifícia Universidade Católica de Minas Gerais

(PUC MG) - Campus Coração Eucarístico

30535-901 - Belo Horizonte - MG - Brasil

rithienatan@gmail.com

A teoria da complexidade computacional é um ramo da teoria da computação em ciência da computação teórica e matemática que se concentra em classificar problemas computacionais de acordo com sua dificuldade de relacionar essas classes entre si. Neste contexto, um problema computacional é entendido como uma tarefa que pode ser resolvida por um computador (que pode ser representado por um problema matemático). Alguns exemplos que podemos citar são o problemas de P, NP e NP completo que serão citados neste texto.

O problemas de P ou problemas na qual possa ser resolvido em tempo polinomial são conjuntos de problemas na qual pode ser resolvido com um tempo de execução  $O(N^k)$  (sendo k uma constante). Pode ser resolvido por uma máquina de Turing determinística, na qual existe uma tabela com no máximo uma entrada para a cada combinação de um símbolo. Um exemplo fácil de citar é o cálculo do Máximo Divisor Comum.

O segundo problema a ser citado e mais complexo que o anterior é o problema de NP. Em geral, é um problema que não é resolvido em tempo polinomial e que são decidíveis por uma máquina de Turing não-determinística. Em ordem de fatores é compreendido que um problema de P pode ser incluído em problemas de NP, devido a sua ordem de complexidade. Um dos exemplos mais relevante é o problema do caixeiro viajante, não o objetivo é percorrer várias cidades pela melhor rota.

Por último, problema de NP-Completo que não teoria, engloba todos os outros problemas, dito que é mais complexo e difícil de resolver. Portanto, é conclusivo que se podessemos resolver estes problemas em tempo polinomial, poderíamos resolver todos os problemas de NP rapidamente. Um exemplo de problema NP-Completo é sobre o problema da soma dos subconjuntos que, dado um conjunto S de inteiros, determine se há algum conjunto não vazio de S cujos elementos somem zero. Na prática, o conhecimento de NP-Completo pode evitar desperdício de tempo na tentativa de encontrar um algoritmo que não existe.