

# **Algoritmos e Estruturas de Dados II**

## **Resumo sobre Teoria da Complexidade e Classes de Problemas**

**Matheus Kraissfeld B. de Lima**

<sup>1</sup>Instituto de Ciências Exatas e Informática  
Pontifícia Universidade Católica de Minas Gerais  
Caixa Postal 1.686 – 30535-901 – Belo Horizonte – MG – Brasil

{matheuskraissfeld@gmail.com.br}

### **1. Informações Gerais**

A teoria da complexidade computacional é um ramo da ciência da computação que busca classificar os problemas por meio de suas dificuldades inerentes, e relacionar essas classes de problemas. Esses problemas são inicialmente possíveis de se resolver por um computador, por meio de desmembramentos do problema em partes menores buscando a resolução por meios matemáticos. Para chegar em uma classificação do problema, leva-se em conta, além da dificuldade para resolução, os recursos que serão necessários para o mesmo como tempo, processamento, armazenamento, número de processadores, portas lógicas, e etc.

### **2. Instância do problema**

A instância de um problema computacional é um conjunto de dados que compõe um dos casos particulares do problema. É importante salientar que nem toda instância possui uma solução. O tamanho da instância é definido pela quantidade de espaço necessário para descrevê-la.

### **3. Classes de problemas**

#### **3.1. Classe P ( *Polynomial* )**

Um problema é classificado como classe P se possuir um algoritmo polinomial que é capaz de resolvê-lo. Problemas desse tipo são considerados tratáveis. Contudo, alguns problemas ainda não possuem classificação, pois não se sabe se pertencem à classe P ou se está fora de P.

Um problema é considerado *não polinomial* se não possuir um algoritmo polinomial capaz de solucioná-lo. Esses problemas são chamados de intratáveis, e um exemplo bem conhecido desse caso é o problema do caixeiro viajante. Além desses casos, existem também problemas insolúveis - chamados de indecidíveis - aos quais não se tem algoritmo algum capaz de solucioná-los (por exemplo, equação diofantina).

#### **3.2. Classe NP ( *nondeterministic polynomial* )**

Essa classe é formada pelos problemas de decisão que possuem verificadores polinomiais. Um algoritmo verificador recebe dois objetos: uma instância do problema e uma cadeia de caracteres chamada de certificado. O certificado é uma espécie de prova de que a instância é positiva. Se retornar "SIM", diz-se que o verificador aceitou o certificado. Para cada instância negativa do problema, não existe certificado que o verificador

aceite. Um certificado é denominado curto quando tem no máximo o mesmo tamanho da instância do problema. Com isso, podemos dizer que um problema pertence à classe NP se possui um certificado curto, e isso justifica a classe NP ser chamada de classe dos problemas razoáveis. Todo problema conhecido da classe NP, pertence a classe P, mas não se consegue provar que P é igual a NP. O instituto Clay de Matemática oferece um prêmio de um milhão de dólares para a solução da questão "P=NP?".

### **3.3. Classe NP-Completo**

Um problema X é NP-Completo se X está em NP e qualquer outro problema em NP pode ser polinomialmente reduzido a X, ou seja, X é NP-Completo se for tão difícil quanto qualquer outro NP. A classe NPC é o conjunto de todos os problemas completos em NP, consequentemente é a classe de todos os problemas mais difíceis de NP. Assim como o caixeiro viajante pertence a P, consequentemente pertence a NP e é um exemplo da classe NP-Completo.

### **References**

Feofiloff, P. (2015). Complexidade computacional e problemas np-completos.