

**BACHARELADO EM** **CIÊNCIA DA COMPUTAÇÃO**

**Thiago Bruchmann Carnaiba**

**Complexidade de Algoritmos**

Presidente Epitácio – SP

2022

1. **Problemas tratáveis e intratáveis**

A complexidade de um algoritmo é uma métrica de quanto tempo um algoritmo levará para solucionar um problema. A métrica mais comum é usar a notação Big O (Big Order function), a notação Big O se preocupa com o pior cenário.

* Worst-case or upper bound: Big-O: O(n)
* Best-case or lower bound: Big-Omega: Ω(n)
* Average-case: Big-Theta: Θ(n)
* Non-tight upper bound: o(n)
* Non-tight lower bound: ω(n)

Problemas para os quais existem soluções com tempo de execução **polinomial** são considerados tratáveis, enquanto aqueles que comprovadamente não podem ser solucionados por **algoritmos polinomiais** são denominados intratáveis. Os problemas tratáveis constituem a classe **P** ao passo que os problemas que somente podem ser resolvidos com o emprego de algoritmos enumerativos (algoritmos com complexidade assintótica de ordem exponencial) constituem a classe **NP**.

Problema **P:** O(nc) - Onde **n** indica o tamanho do problema e **c** é uma constante qualquer.

Problema **NP**: O(cn) - Se já provado que melhor solução é **Ο(cn)**.

Exemplos:

Problema **P**

algoritmos de: busca linear Ο(n) e de busca binária Ο(log2n) em sequências ordenadas de números.

Problema **NP**

3-cores: colorir com 3 cores diferentes os vértices de um grafo, de forma que 2 vértices compartilhando a mesma aresta não tem a mesma cor.

Problema **NP-completo**: Em termos práticos, um problema NP-completo comum pode ser resolvido usando algoritmos de enumeração implícitos como programação dinâmica. Nesse caso, a complexidade de tempo do algoritmo não é polinomial no tamanho dos dados de entrada, mas é polinomial no tamanho desses dados. Quando a complexidade de um algoritmo é polinomial no tamanho dos dados, mas não no comprimento da entrada, referem-se a ele como um algoritmo pseudo-polinomial. Um problema NP-completo solucionável por um algoritmo pseudo-polinomial é chamado de NP-completo comum. Caso contrário, o problema é NP-difíceis.

Problema **NP-difíceis**: é uma classe de problemas que são, informalmente, "Pelo menos tão difíceis quanto os problemas mais difíceis em NP". Um exemplo de um problema NP-difícil é o problema de decisão da soma de subconjuntos, que é o seguinte: dado um conjunto de números inteiros, pode algum subconjunto não-vazio deste somar zero? Isso é uma questão do tipo sim/não, e acontece de ser NP-completo.

1. **Referências**

Dineshpathak. **Algorithmic Complexity**. [*S. l.*]. Disponível em: <https://devopedia.org/algorithmic-complexity#qst-ans-0>. Acesso em: 24 jun. 2022.

**The Complexity of Problems**. Disponível em: <https://link.springer.com/content/pdf/bbm%3A978-1-4614-5152-5%2F1.pdf>. Acesso em: 24 jun. 2022.

Nasar, Audrey A. (2016) **The history of Algorithmic complexity** The Mathematics Enthusiast: Vol. 13 : No. 3 , Article 4. DOI: <https://doi.org/10.54870/1551-3440.1375>. Disponível em: <https://scholarworks.umt.edu/cgi/viewcontent.cgi?article=1375&context=tme>.

Acesso em: 24 jun. 2022.