

Teoria da Complexidade

Projeto e Análise de Algoritmo

Daniel Capanema

Pontifícia Universidade Católica de Minas Gerais

Introdução

- Problemas intratáveis ou difíceis são comuns na natureza e nas áreas do conhecimento.
- Problemas “fáceis”: resolvidos por algoritmos polinomiais.
- Problemas “difíceis”: somente possuem algoritmos exponenciais para resolvê-los.
- A complexidade de tempo da maioria dos problemas é polinomial ou exponencial.

Introdução

- **Polinomial:** função de complexidade é $O(p(n))$, onde $p(n)$ é um polinômio
- Ex.: algoritmos com pesquisa binária ($O(\log n)$), pesquisa sequencial ($O(n)$), ordenação por inserção ($O(n^2)$), e multiplicação de matrizes ($O(n^3)$)
- **Exponencial:** função de complexidade é $O(c^n)$, $c > 1$.
- Ex.: problema do caixeiro viajante (PCV) ($O(n!)$)
- Mesmo problemas de tamanho pequeno a moderado não podem ser resolvidos por algoritmos não-polinomiais

Problemas NP Completo

- A teoria de complexidade a ser apresentada não mostra como obter algoritmos polinomiais para problemas que demandam algoritmos exponenciais, nem afirma que não existem
- É possível mostrar que os problemas para os quais não há algoritmo polinomial conhecido são computacionalmente relacionados.
- Formam a classe conhecida como NP
- Propriedade: um problema da classe NP poderá ser resolvido em tempo polinomial se e somente se todos os outros problemas em NP também puderem
- Este fato é um indício forte de que dificilmente alguém será capaz de encontrar um algoritmo eficiente para um problema da classe NP