Teoria da Computação

Guilherme Henrique de Souza Nakahata

Universidade Estadual do Paraná - Unespar

03 de Julho de 2024

Histórico da Teoria da Computação

- **Alan Turing** (1912-1954)
  - Matemático e cientista da computação britânico.
  - Considerado o pai da computação teórica.
  - Contribuições:
    - Máquina de Turing;
    - Teste de Turing;
    - Decifração de códigos na Segunda Guerra Mundial;
- **Kurt Gödel** (1906-1978)
  - Matemático e lógico austro-húngaro.
  - Conhecido pelo Teorema da Incompletude.
  - Contribuições:
    - Provou que em qualquer sistema formal consistente, existem proposições verdadeiras que não podem ser provadas dentro do sistema.
- **Emil Post** (1897-1954)
  - Matemático e lógico polonês-americano.
  - Contribuições:
    - Teoria da computabilidade;
    - Sistemas formais.

Computabilidade

- Classificação de problemas:
  - Indecidíveis: Problemas para os quais não existe algoritmo que os resolvam.
  - Intratáveis: Problemas para os quais não existe algoritmo que os resolvam em tempo polinomial.
  - Tratáveis: Problemas para os quais existe algoritmo que os resolvam em tempo polinomial.

- Problemas de decisão: Respondem **sim** ou **não**;
  - Dado um grafo G(V,A), orientado e ponderado nas arestas, dois vértices u, v $\in$ V e um inteiro não negativo K, existe um caminho em G entre u e v com comprimento menor ou igual a k?

- Problemas de localização: Encontra uma solução que satisfaça o problema;
  - Dado um grafo G(V,A), orientado e ponderado nas arestas, dois vértices u, v $\in$ V, o problema consiste em encontrar um caminho entre u e v.

- Problemas de otimização: Dentre as soluções possíveis encontrar a melhor;
  - Dado um grafo G(V,A), orientado e ponderado nas arestas, dois vértices u, v $\in$ V, o problema consiste em encontrar um caminho de menor comprimento possível.

Computabilidade

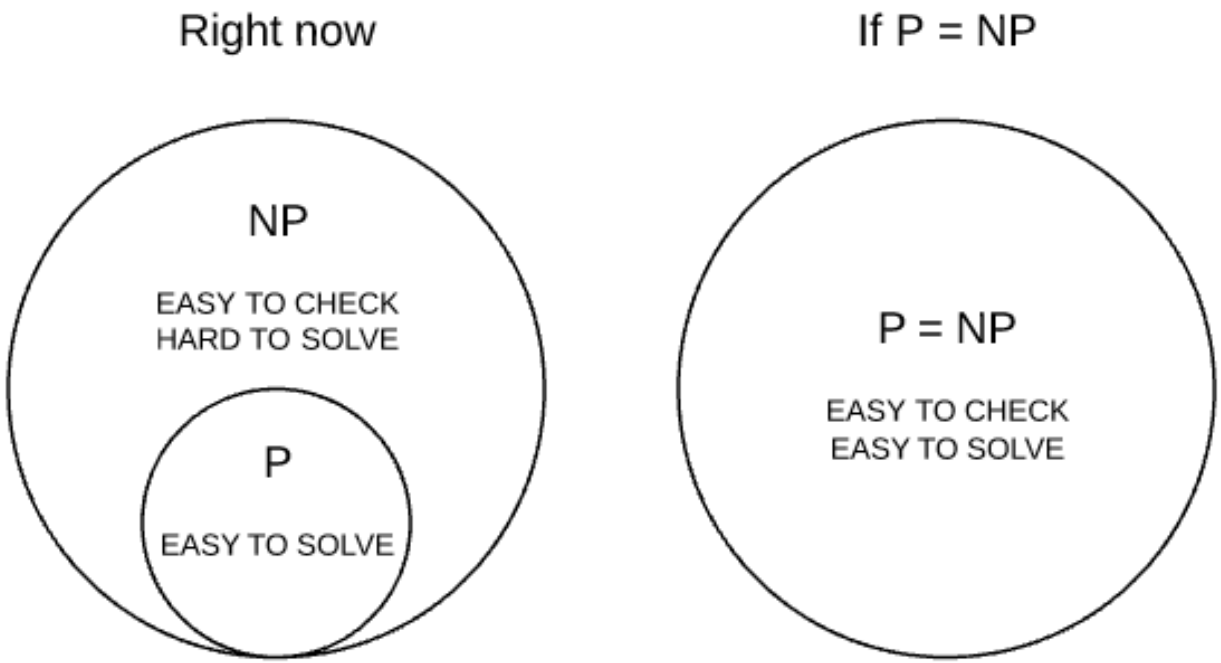
- Classe **P**: É a classe dos problemas de decisão que podem ser resolvidos por algoritmos (determinísticos) de tempo polinomial.
- Exemplos de problemas:
  - Ordenação;
  - Caminho mais curto em grafo;
  - Árvore geradora de peso mínimo;

- Classe **NP**: é a classe dos problemas de decisão para os quais são conhecidos algoritmos não-determinísticos polinomiais;
- O algoritmo gera uma solução candidata ao problema e verifica a resposta "sim" em tempo polinomial;
- São construídos:
  - Choice - advinha uma solução;
  - Check - verifica (em tempo polinomial) se uma proposta de solução leva ou não a uma resposta "sim"
- Exemplo: Problema do Caminho Hamiltoniano em Grafos Não Orientados;

Computabilidade

- Grande questão da Ciência da Computação teórica (1971);
- $P = NP$?
- $NP \subset P$?
- Soluções que podem ser verificadas rapidamente podem ser encontradas rapidamente?
- Diversas implicações em áreas como criptografia, otimização, inteligência artificial;
- Prova definitiva que demonstre que P é igual a NP ou que P é diferente de NP;

Computabilidade



NP-Completo

- NP-Completo;
- Pertencer a NP;
- Ser tão difícil quanto o problema mais difícil em NP;
- Se existir um algoritmo (determinístico) polinomial para a resolução de algum problema NP-completo;
- Todos os problemas da classe NP também poderão ser resolvidos em tempo polinomial;
  - Caixeiro viajante;
  - Mochila;
  - Coloração de grafos;
  - Programação inteira.
- Redução polinomial;
  - Demonstrar que outro problema NP pode ser reduzido a ele;

NP-hard

- Um problema é **NP-hard** se é, no mínimo, tão difícil quanto os problemas mais difíceis em NP.
- Qualquer problema em NP pode ser reduzido a um problema NP-hard em tempo polinomial.
- Problemas NP-hard não precisam estar em NP, ou seja, suas soluções não precisam ser verificáveis em tempo polinomial.

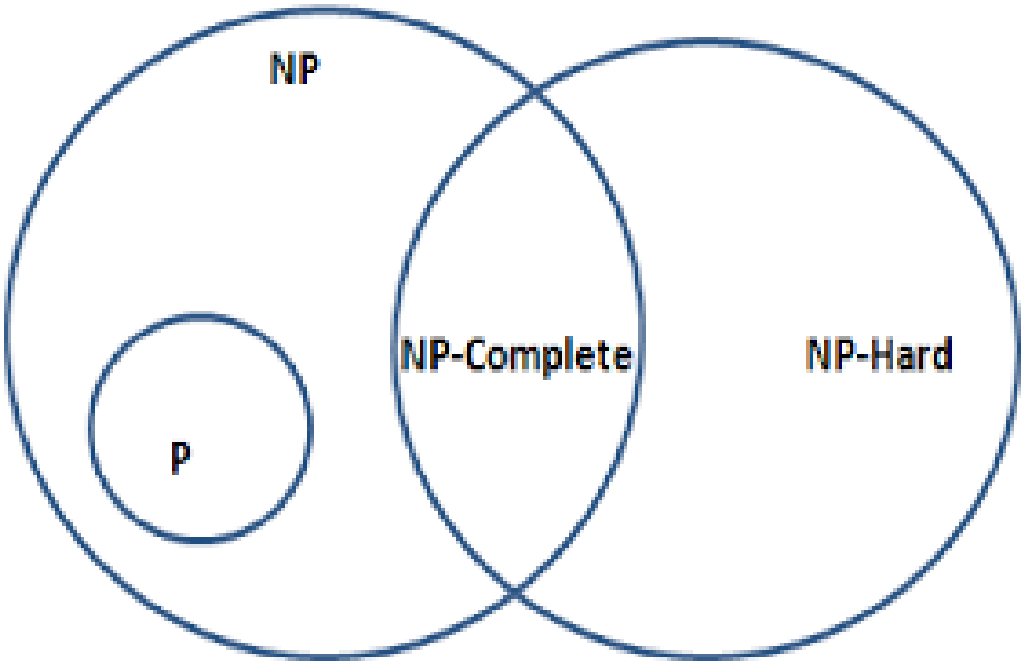
Diferença entre NP-completo e NP-hard

- **NP-completo**:
  - O problema deve estar em NP.
  - Deve ser NP-hard.
- **NP-hard**:
  - O problema é, no mínimo, tão difícil quanto qualquer problema em NP.
  - Não precisa estar em NP.

Importância na Computação

- Problemas NP-hard são críticos para entender a complexidade computacional.
- Definem os limites do que pode ser resolvido eficientemente com os recursos computacionais disponíveis.
- Soluções eficientes para problemas NP-hard têm implicações profundas em várias áreas;

Computabilidade



Atividade

- Grupos de até 3 pessoas:
  - Problema de correspondência de Post;
  - Problema do Halting (Parada);
  - Problema da Satisfação de Restrições (Constraint Satisfaction Problem - CSP);
  - Problema do Subconjunto (Subset Sum Problem);
  - Exemplo de redução de problema (Problema da mochila para outro).
- Complexidade;
- Categorias (decidíveis, indecidíveis, tratáveis e intratáveis);
- Aplicações práticas;
- Limitações;
- Reflexão ética e social.

Bibliografia Básica

- LEWIS, H. R.; PAPADIMITRIOU, C. H. **Elementos de Teoria da Computação**. 2 ed. Porto Alegre: Bookman, 2000.
- VIEIRA, N. J. **Introdução aos Fundamentos da Computação**. Editora Pioneira Thomson Learning, 2006.
- DIVERIO, T. A.; MENEZES, P. B. **Teoria da Computação: Máquinas Universais e Computabilidade**. Série Livros Didáticos Número 5, Instituto de Informática da UFRGS, Editora Sagra Luzzato, 1 ed. 1999.

Obrigado!
Dúvidas?

Guilherme Henrique de Souza Nakahata

guilhermenakahata@gmail.com

https://github.com/GuilhermeNakahata/UNESPAR-2024