Problemas P vs NP

Victor Ramos - 25/05/2023

Entendendo P e NP

Na ciência da computação, uma das maiores dúvidas existentes é relacionada ao famoso problema de equivalência de P e NP, sendo “P” um conjunto de problemas que possuem soluções simples, e “NP” um conjunto de problemas que aparentam ser extremamente difíceis de se solucionar. Então, P = NP implicaria em que os problemas aparentemente difíceis teriam soluções relativamente simples.

Essa diferenciação é feita da seguinte forma: os problemas “P” são aqueles cujo tempo de execução é proporcional a polinômios envolvendo o número de elementos que o programa está manipulando (chamado de N). Enquanto NPs são problemas que podem ter seu tempo de execução calculado por um polinômio, porém, geralmente levam um tempo exponencial para serem resolvidos, desta forma, estes problemas tornam-se um empecilho, uma vez que problemas exponenciais são extremamente lentos.

Com isso, o problema chamado “P = NP” busca a seguinte resposta: “Se um programa pode ter seu tempo calculado em tempo polinomial, ela pode ser feita em tempo polinomial?”.

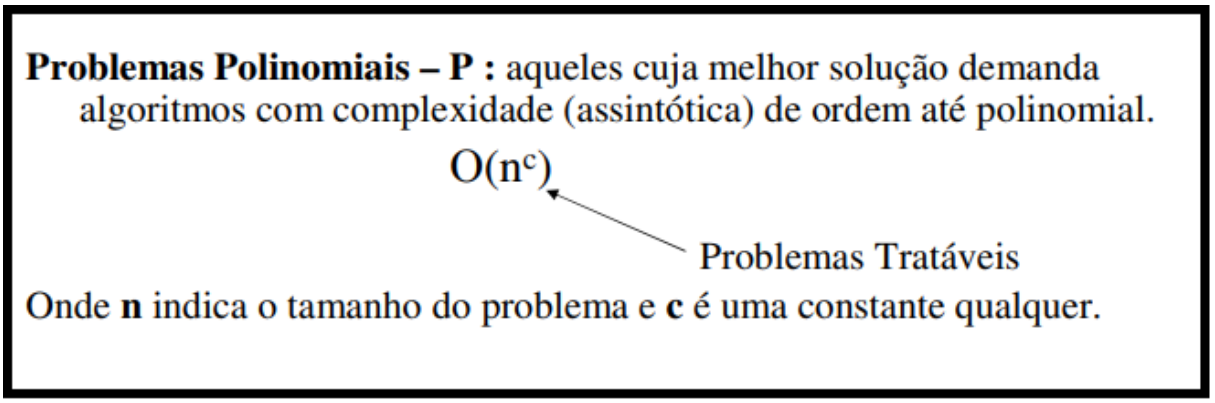
A principal motivação para esta busca é que, a maioria dos problemas NPs que exigem um tempo exponencial são os chamados NP-completos. O que significa que, a solução de um desses problemas automaticamente solucionaria todos os outros. Isto juntamente com o fato de que na vida real pode ser encontrada diversos problemas denominados “NPs”. Faz com que seja de extremo interesse para a Ciência da Computação, a solução destes problemas chamados “NP-completos”.

Apesar de que, com os estudos atuais chegou-se à conclusão de que provavelmente P não é igual a NP, o que significa que estas tais “soluções fáceis para NPs” provavelmente nunca serão encontradas. Ainda é de extrema importância o estudo destes problemas para a aprofundação da compreensão da complexidade computacional, sendo muito útil em diversas áreas, como na criptografia.

Definição de P e Exemplos

Como dito anteriormente, problemas P são aqueles cuja melhor solução demanda algoritmos com complexidade de ordem polinomial. Estes problemas são conhecidos como "fáceis" ou “trataveis”. Na computação, são aqueles que tem seu tempo de execução relativamente rápido.

Figura 1 - Ilustração de problemas P.



Fonte: Prof. Léo Pini Magalhães.

Diversos problemas são classificados como Problemas P, entre eles estão:

- "**Equação de Segundo Grau**", que visa descobrir se dados "*a*", "*b*" e "*c*", existe um número x tal que: $ax^2+bx+c=0$.

- "**Máximo Divisor Comum**", que visa descobrir se, dados "*m*", "*n*" e "*k*", existe um divisor comum de "*m*" e "*n*" que seja maior ou igual a "*k*".

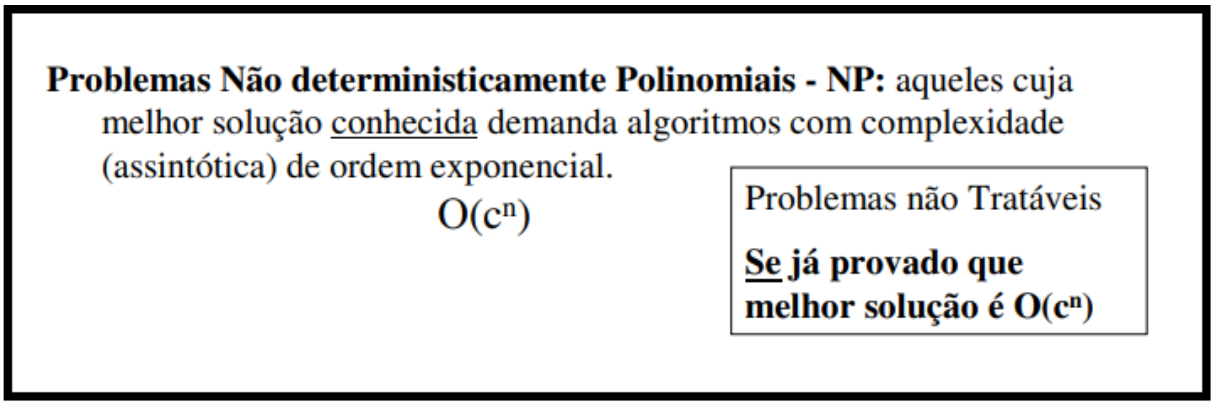
Entre diversos outros.

Definição de NP e Exemplos

Problemas NP são aqueles que, diferente dos problemas P, são aqueles cuja melhor solução demanda algoritmos com complexidade de ordem exponencial. Estes problemas são conhecidos como intratáveis, isto não significa que não existe solução para estes problemas, apenas que a melhor solução encontrada é lenta e ineficiente. Na computação, este tipo de solução pode ser extremamente ruim, uma vez que podem existir milhares de "entradas" no programa, fazendo com que seu tempo de execução escale de forma extremamente grande.

Vale sempre lembrar que, "NP" não significa "não polinomial", mas sim "nondeterministic polynomial".

Figura 2 - Ilustração de problemas NP.



Fonte: Prof. Léo Pini Magalhães.

Alguns dos problemas NP mais conhecidos são:

- "**Traveling Salesman Problem**", que visa determinar o menor caminho para visitar várias cidades e retornar a sua origem.

- "**The Graph Isomorphism Problem**", que visa determinar se dois grafos podem ser feitos de forma idêntica.

Diferença entre NP e NP-Completos

Um problema é "NP-Hard" se um algoritmo para sua solução pode ser modificado para servir de solução para qualquer problema NP (é importante destacar que nem todos os problemas "NP-Hard" são também problemas "NP"). Porém, quando um problema pode ser verificado como NP e NP-Hard, ele é chamado de NP-Completo. Desta forma, essa afirmação mostra que, caso um algoritmo de tempo polinomial fosse encontrado para qualquer problema NP-Completo, ele automaticamente serviria de solução para todos os outros problemas "NP".

Referências

HARDESTY, Larry. Explained P vs. NP. 2009. Disponível em: <https://news.mit.edu/2009/explainer-pnp>. Acesso em 25 de fevereiro de 2023.

HOSCH, William. P versus NP problem. 2023. Disponível em: <https://www.britannica.com/science/P-versus-NP-problem>. Acesso em 25 de fevereiro de 2023.

WEISSTEIN, Eric. NP-Problem. 2023. Disponível em: <https://mathworld.wolfram.com/NP-Problem.html>. Acesso em 25 de fevereiro de 2023.

FEOFILOFF, Paulo. Complexidade e Problemas NP-completos. 2021. Disponível em: <https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/NPcompleto.html>. Acesso em 25 de fevereiro de 2023.

MAGALHÃES, Léo. Problemas P, NP e NPC. 2011. Disponível em: <https://www.dca.fee.unicamp.br/~marco/cursos/ea869_11_1/slides/ea869-mar2011-cap1-1-algoritmoTR77a79.pdf>. Acesso em 25 de fevereiro de 2023.

The Editors of Encyclopaedia Britannica. Traveling salesman problem. 2023. Disponível em: <https://www.britannica.com/science/traveling-salesman-problem>. Acesso em 25 de fevereiro de 2023.

GROHE, Martin & SCHWEITZER, Pascal. The Graph Isomorphism. 2023. Disponível em: <https://cacm.acm.org/magazines/2020/11/248220-the-graph-isomorphism-problem/abstract>. Acesso em 25 de fevereiro de 2023.