

**P vs NP**

Antes de entender a diferença entre problemas P e NP é necessário entender o que são algoritmos polinomiais e se um problema pode ser facilmente tratado. Algoritmos polinomiais são algoritmos que retornam uma solução de um determinado problema ao receber uma instância do mesmo, de forma que a solução seja considerada razoavelmente rápida. Porém, para uma solução ser considerada razoavelmente rápida ela precisa se encaixar na categoria de tempo polinomial, onde o tempo gasto para resolver o pior caso possível do problema deve ser determinado por uma função polinomial do tamanho de instâncias do problema, ou seja um problema de tamanho  $N$  é resolvido em um tempo dado por uma função de  $N$ .

Desta forma podemos determinar o que são problemas P e NP, começando pelos problemas P ou problemas polinomiais. Estes são os problemas que possuem uma solução que é um algoritmo polinomial e pode ser considerado um problema fácil de ser resolvido, por outro lado existem os problemas não polinomiais que são problemas que nunca irão possuir um algoritmo polinomial que seja capaz de resolvê-los, assim a melhor opção para solucionar uma de suas instâncias é testar individualmente cada uma das possíveis soluções. É importante notar que problemas não polinomiais não são podem ser atribuídos a abreviação NP, pois ela é dedicada aos problemas que são categorizados como Nondeterministic Polynomial.

Exemplos de problemas P: raízes de uma equação de segundo grau e mdc entre dois números. Os problemas NP por outro lado são problemas os quais uma solução sugerida pode ser verificada em tempo polinomial, além disso a sugestão de solução não precisa seguir nenhuma regra particular, ou seja não existe só uma melhor solução para um problema. Devido a esse conceito é possível ainda categorizar um problema NP-Completo. Este tipo de problema é encontrado quando um problema NP existe e todos os outros problemas NPs podem ser reduzidos em tempo polinomial até o problema NP em específico, assim existindo ainda mais uma implicação que se for possível encontrar um algoritmo eficiente para qualquer NP-Completo a solução valerá para todos os problemas desta classe.

Exemplo de problema NP: “O problema do caixeiro viajante” que levanta a pergunta “Dado um conjunto de cidades e a distância entre cada par de cidades, é possível encontrar a menor rota que visita cada cidade uma única vez e retorna ao ponto inicial”.

Referências:

PROBLEMAS NP-completos. [S. l.], 17 jan. 2021. Disponível em: [https://www.ime.usp.br/~pf/analise\\_de\\_algoritmos/aulas/NPcompleto.html#polynomial-algorithm](https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/NPcompleto.html#polynomial-algorithm). Acesso em: 23 fev. 2023.

NP-COMPLETE problem. [S. l.], 10 out. 2022. Disponível em: <https://www.britannica.com/science/NP-complete-problem>. Acesso em: 23 fev. 2023.

TRAVELLING Salesman Problem using Dynamic Programming. [S. l.], 6 fev. 2023. Disponível em: <https://www.geeksforgeeks.org/travelling-salesman-problem-using-dynamic-programming/>. Acesso em: 23 fev. 2023.