

# **P, NP e NP-Completo**

## **Teoria da Complexidade**

**Marcos A. Lommez C. R.<sup>1</sup>**

<sup>1</sup>Ciencia da Computação - Pontificia Universidade Catolica de Minas Gerais  
Belo Horizonte – MG – Brazil

Dentro da teoria da complexidade existem diferentes tipos de problemas que podem ser resolvidos por diferentes tipos de algoritmos, estes algoritmos podem ser medidos por seu nivel de complexidade. O nivel de complexidade de um algoritmo é determinado pela contagem das operações que acontecem no mesmo, sendo que esta é influenciada pelo valor de entrada que é fornecido, assim sendo, para um algoritmo com entrada 'n', 'f(n)' vezes o algoritmo é repetido.

Muitos desses algoritmos são classificados como "ótimos" quando o seu grau de complexidade é o menor possível baseado em sua solução ideal, a partir daqui é importante perceber que não necessariamente sabemos ou foi descoberto a maneira de se levar este algoritmo ao melhor caso possível, não sendo possível defini-lo como "ótimo". Esta é uma área teórica onde é necessário a existência de determinados conceitos para se fazer determinadas afirmações, e isso é o que começa a criar a necessidade de uma classificação entre aqueles problemas onde o seu caso "ótimo" é conhecido ou a sua solução "ótima" também é computável.

Assim entendendo a necessidade de uma classificação definimos os algoritmos nos tipos P, NP, NP-Completo e NP-difícil. A maior pergunta aberta da Ciência da Computação atualmente é saber se todo algoritmo do tipo NP é P, ou seja, se é possível descobrir sua solução "ótima" ou a única forma de resolvê-lo é através de uma busca gulosa de todos os resultados possíveis, ou até mesmo não sendo possível resolvê-los. Os nomes dados aos tipos de algoritmos indicam se podem ser calculados em tempo polinomial (P), se o tempo é polinomial mas não-determinístico (NP), o que não deve ser erroneamente chamado de não-polinomial, alguns algoritmos acreditam-se que seu caso "ótimo" seja uma busca fatorial de testagem de todas as possibilidades, o que os torna um NP-Completo, e por fim alguns problemas são classificados como NP-difíceis, estes nem sequer precisam estar dentro das classes NP.

A grande questão dessa classificação está no ponto onde problemas do mesmo tipo podem ser transformados um no outro, um exemplo NP pode ser transformado em outro exemplo NP, e por isso foi criada essa classificação. Junto disso surgiu foi descoberto que se algum dos problemas do tipo NP for redutível ao tipo P, logo todos os problemas do tipo NP também serão. É importante dizer que não necessariamente a adaptação destes problemas é algo fácil ou que o grau de dificuldade em algum momento se iguala, apenas que é possível otimizar um problema que antes não sabia se era possível.

Definir que NP é igual a P ou diferente implicaria em grandes mudanças não apenas ao mundo teórico mas ao mundo prático. Pesquisas poderiam ser feitas sem medo de que o resultado não exista, ou no caso contrário paradas quando descoberto que não é possível ir além. Não apenas benefícios seriam trazidos como problemas também podem ser gerados como no caso da criptografia, caso todos os problemas NP fossem redutíveis

a P em algum momento seria descoberto uma maneira de quebrar chaves de segurança facilmente e levando a grandes mudanças que se tornariam necessárias no mundo da criptografia.

Ao mesmo tempo dentro do mundo academico existem diversas teses que defendem que P é diferente de NP e também que é igual a NP, entre elas existe uma grande questão ideologica sobre o ponto que provar a igualdade entre P e NP também é provar que as respostas não foram descobertas por que os estudiosos anteriores a descoberta não eram capazes de fazer isso, colocando em duvida a "grandiosidade" de seus nomes, uma briga de ego que acaba impedindo a verdade que a ciencia deve sempre buscar. Entre as relações citadas agora é possível explicar melhor sobre a definição de NP-difícil que é o ponto onde um problema tem essa classificação quando um NP-completo pode se transformado em num problema NP-difícil e o mesmo não precisa ser um problema de decisão.

Por fim é importante dizer que esta é uma discussão teorica muito importante para o mundo atual e influencia não apenas a computação mas muitas outras areas, a partir do momento em que a computação existe para servir e ajudar a resolver problemas de todos os tipos de forma eficiente e pratica, por isso é importante conhecer a complexidade de suas soluções e se é ou não possível continuar otimizando para se obter um melhor resultado. O grau de dificuldade para se aprender a logica de algoritmo não necessariamente é proporcional a sua classificação e provar a sua resposta também não necessariamente é proporcional ao seu tempo de execução, sendo possível sim existirem respostas que demoram mais tempo para serem provadas do que a sua produção.