A Teoria da Complexidade

João Henrique Carvalho

Inicialmente a resposta para a pergunta P = NP pode parecer trivial, uma vez que só é necessário “testar” um determinado problema pertencente a classe P, e outro de classe NP e comparar os resultados com relação à complexidade matemática, porém para responder essa pergunta de maneira satisfatória e entender porque esse é o maior problema em aberto da computação, será necessário expandir um pouco o nosso conhecimento sobre como essa pergunta surgiu.

A princípio, talvez seja necessário definir o significado da palavra algoritmo para a matemática e a computação. Segundo o dicionário da Oxford Languages[1], um algoritmo é um “conjunto de regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de etapas”. Ou seja, um algoritmo que resolverá um determinado problema, irá possuir instruções específicas sobre como resolver tal problema, e a “resposta” do problema será encontrada após uma quantia de tempo. Podemos então classificar e agrupar os diversos algoritmos que escrevemos, usando como parâmetro o tempo que o algoritmo leva para resolver o problema em comparação com o tamanho do problema em si, esse parâmetro é chamado de Tempo de Execução.

Convencionamos a chamar o tamanho de um problema de n e a complexidade de um algoritmo é representada com o uso de uma notação chamada Big-Oh Notation[2] (ou notação do grande O), que representa uma aproximação da quantidade de etapas necessárias para resolver o problema,uma maneira mais prática de "visualizar” a diferença entre as diferentes classes de da notação é entender o que acontece com o tempo de execução alterarmos o tamanho do problema n por exemplo: considere um problema que pertence a classe $O(n)$ ao dobrar o tamanho da entrada, este algoritmo irá necessitar do dobro do tempo para resolver o problema, enquanto isso, para um algoritmo $O(n^2)$, ao dobrar o tamanho da entrada, necessitamos do quádruplo do tempo para resolver o problema, podemos então dizer que quanto “menor” a notação O para um algoritmo, mas eficiente ele é para a resolução do problema.

Retomando a discussão sobre P = NP, temos que P é o conjunto dos problemas que pode ser resolvido em tempo polinomial, ou seja, $O(n)$, $O(n^2)$, $O(n^5)$, etc. Informalmente, dizemos que tais problemas podem ser resolvidos em uma quantia "razoável" de tempo. Em contrapartida, NP é o conjunto de problemas que podem ser resolvidos em tempo polinomial, ou seja, necessitam de uma quantidade de tempo superior quando comparados ao tempo polinomial, por exemplo, um problema que só pode ser resolvido por um algoritmo $O(2^n)$, ou seja, em tempo exponencial.

Ainda assim, dentro do conjunto NP, existem problemas que não podem ser resolvidos em tempo polinomial, porém, verificar se uma determinada resposta está correta ou não é trivial. Dentro do conjunto NP, ainda existe um grupo de problemas em que até a verificação de um resultado é complexa demais para ser realizada em tempo polinomial, esse grupo de problemas é conhecido como NP-completo, informalmente, dizemos que esses problemas são pelo menos tão difíceis quanto qualquer outro problema pertencente a NP.

Como um problema NP-completo é pelo menos tão difícil quanto qualquer outro problema NP, podemos assumir que, caso seja possível resolver um problema NP-completo em tempo polinomial, também será possível resolver todos os problemas NP-completos em tempo polinomial, por que esses são igualmente difíceis de serem resolvidos, então, se uma maneira mais fácil para resolver um problema for encontrada, sabemos que também existem soluções mais fáceis para todos os outros problemas.

Finalmente podemos analisar a pergunta, P é igual a NP? Muitos cientistas acreditam que não, e muito esforço já foi feito para provar que $P \neq NP$ mas nenhuma prova conclusiva foi encontrada, deixando a questão ainda em aberto. Porém um dos aspectos mais interessantes dessa pergunta é entender o que poderia ser feito caso $P = NP$, e entender as consequências dessa descoberta. Caso seja provado que a igualdade seja verdadeira, todos os problemas que necessitam de uma certa dificuldade para serem resolvidos, poderiam ser resolvidos com a mesma trivialidade de serem verificados, como por exemplo a criptografia, que se baseia em verificar se um número é primo ou não.

Talvez a maneira mais explicativa de compreender as consequências de $P = NP$ seja através da frase de Scott Aaronson “Se $P = NP$, então o mundo seria um lugar profundamente diferente do que supomos que ele seja. Não haveria nenhum valor especial em "saltos criativos," nenhuma lacuna fundamental entre resolver um problema e reconhecer a solução, uma vez que ele é encontrado. Todos que pudessem apreciar uma sinfonia seriam Mozart,...”[3].

Referencias

[1] Dicionario Oxford Languages, significado oferecido através do Google;
[2] Slides vistos em sala;
[3] Scott Aaronson