

Analise de Complexidade - P, NP e NP-Completo

Lucas Santiago de Oliveira

Novembro 2019

1 Analise de complexidade

A analise de complexidade é uma área da computação e da matemática voltada a analisar a eficiência dos algoritmos. Com o avanço da computação é importante mensurar o tempo gasto durante a execução desses algoritmos. Lembrando que um algoritmo é um conjunto de instruções finitos.

O cálculo de complexidade é dividido em três partes. Sendo a primeira, melhor caso, o número mínimo de iterações para se resolver o problema. A segunda, o pior caso, trata-se do número máximo de iterações necessários para se resolver o problema. E, por último e mais comumente usada, o caso médio, essa complexidade é definida com relação a uma distribuição de probabilidade sobre as entradas, aquilo que tende a acontecer mais vezes dada várias possibilidades de entrada.

Por exemplo, considerando o algoritmo de ordenação *quicksort*. Isso resolve o problema de ordenar uma lista de inteiros que é dada como entrada. O pior caso é quando a entrada já está ordenada ou está em ordem inversa, e o algoritmo leva tempo $\mathcal{O}(n^2)$ para este caso. Se assumirmos que todas as permutações possíveis da lista de entrada são igualmente prováveis, o tempo médio necessário para a ordenação é $\mathcal{O}(n \times \log_2 n)$. O melhor caso ocorre quando cada pivô divide a lista pela metade, levando o tempo de $\mathcal{O}(n \times \log_2 n)$.

2 P e NP

As classes P e NP são pensadas sobre uma máquina de Turing, que possui um conjunto finito de instruções. A classe de complexidade P é aquela, onde é possível ser resolvido de forma eficiente. A classe NP, por outro lado, contém muitos problemas resolvíveis, mas sem um algoritmo eficiente para resolvê-lo. Um exemplo de problema P, é a ordenação de um vetor de números inteiros. Existem vários algoritmos eficientes que podem ordená-lo como o *quicksort*, *mergesort*, *bubblesort* ou *heapsort*. Já o problema NP, é a chave criptografica RSA. Essa chave é baseada na ideia de multiplicação de dois primos muito grandes. E para quebrar essa chave e redescobrir

qual foram os dois números multiplicados é preciso fazer testes de tentativa e erro, até descobrir. Não existindo ainda, uma fórmula matemática que possa voltar a multiplicação. Nesse último caso, a multiplicação entre os dois primos seria um problema P e descobrir quais números foram multiplicados NP.

3 NP e NP-Completo

Por fim, temos o conjunto dos problemas NP-Completo que são os mais difíceis do conjunto NP. Esse conjunto é tão complexo que para ser considerado NP-Completo é necessário que o conjunto-solução desse problema também sirva para resolver outros problemas NP ou partes de outros NP-Completo. Esse conjunto de problemas existem na computação há quase meio século. Se algum dia for possível resolvê-los, será possível provar que $P = NP$ e reduzir o tempo de execução dos algoritmos NP ao tempo de execução dos algoritmos P. Entretanto, se não for possível resolvê-los, será provado que $P \neq NP$.