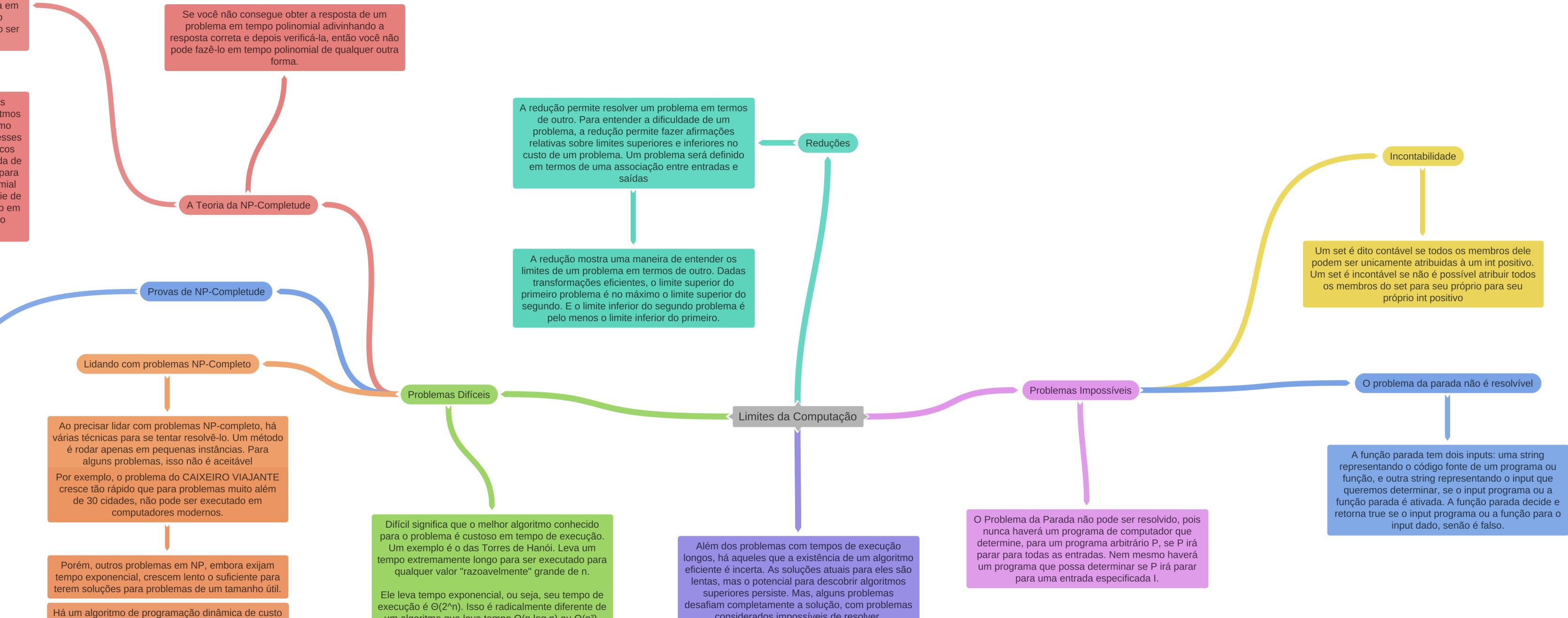
coggle

made for free at coggle.it

A ideia de "adivinhar" a resposta de problema, ou verificar todas as soluções possíveis para ver qual está correta, é chamada de não determinismo. E qualquer problema com um algoritmo que roda em uma máquina não determinística em tempo polinomial recebe um nome especial: Ele é dito ser um problema em NP.

Conhecemos algoritmos não determinísticos eficientes, mas não sabemos se existem algoritmos determinísticos eficientes para eles. Ao mesmo tempo, não foi capaz de provar que nenhum desses problemas não possui algoritmos determinísticos eficientes. Essa classe de problemas é chamada de NP-completo. Se alguém encontrar a solução para qualquer um deles que rode em tempo polinomial em um computador comum, então por uma série de reduções, todos os outros problemas que estão em NP também podem ser resolvidos em tempo polinomial em um computador comum!

Para provar que um problema é NP-completos, é preciso provar que um problema H é NP-completo. Logo, para mostrar que qualquer problema X é NPdifícil, é preciso reduzir H para X. Ao fazer essas provas, não se pode inverter essa redução! Se reduzirmos o problema candidato X para o problema conhecido e difícil H, estamos usando H para resolver X. Mas, quando reduzimos H para X, estamos usando X para resolver H. Logo os dois são difíceis.



considerados impossíveis de resolver.

um algoritmo que leva tempo $\Theta(n \log n)$ ou $\Theta(n^2)$.

Θ(nK) para n objetos sendo encaixados em uma mochila de tamanho K.Apesar disso, o Problema da Mochila é NP-completo. Ele é tratável se os números estiverem na casa dos milhares. Algoritmos assim são chamados de algoritmos de tempo pseudo-polynomial.