**Classificação de Problemas Computacionais (Problemas NP)**

**Equipe:**

**Clésio de Araújo Gonçalves**

**Lucas Marques de Sousa Silva**

1. **P**

Diz-se que um problema pertence à classe P se existe um algoritmo que o pode resolver em tempo **polinomial**. Um problema computacional é polinomial se existe um algoritmo polinomial para o problema. Problemas desse tipo são considerados **tratáveis ou fáceis**.

Alguns exemplos de problemas polinomiais:

* **Equação inteira do segundo grau:** Dados números inteiros a, b e c, encontrar um número inteiro x tal que ax² + bx + c = 0 (ou constatar que tal x não existe).
* **Divisor comum grande:** Dados números naturais m, n e k, encontrar um divisor comum de m e n que seja maior que k (ou constatar que tal divisor não existe).
* **Subsequência crescente longa:** Dada uma sequência s1, … , sn de números naturais e um número natural k, encontrar uma subsequência crescente de s1, … , sn que tenha comprimento maior que k (ou constatar que tal subsequência não existe).
* **Caminho curto:** Dados vértices r e s de um grafo e um número d, encontrar um caminho de r a s no grafo que tenha comprimento menor que d (ou constatar que tal caminho não existe).
* Os correspondentes **problemas de decisão** (problemas cujas instâncias têm solução binária: “sim” ou “não”) também são polinomiais.

1. **NP**

Diz-se que um problema pertence à classe NP se ele pode ser resolvido por algum algoritmo (sem alegações sobre seu tempo de execução) e qualquer solução pode ser **verificada dentro do tempo polinomial por algum outro algoritmo**.

São conhecidos algoritmos não-determinísticos polinomiais, ou seja, o algoritmo gera uma solução candidata ao problema e verifica sua viabilidade em tempo polinomial.

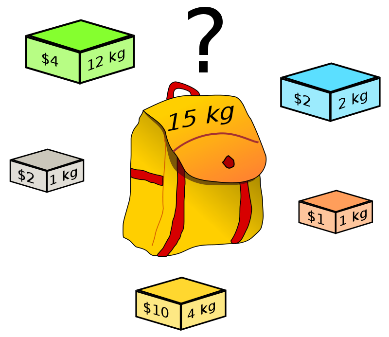
Exemplos de problemas da classe NP:

* **problema da soma do subconjunto:** dado um conjunto de inteiros, existe algum conjunto de um ou mais elementos desse conjunto cuja soma é zero?
* o **conjunto dos problemas de decisão que são polinomialmente verificáveis**
* **problema da fatoração** - é fácil verificar se um dado número natural p é divisor de n
* **problema do campo minado** - é possível verificar em tempo polinomial s e uma dada distribuição das minas é consistente com a configuração dada
* o **problema do caixeiro viajante** que tenta determinar a menor rota para percorrer uma série de cidades (visitando uma única vez cada uma delas), retornando à cidade de origem
* **coloração de Grafos** - é um caso especial de rotulagem de grafos, ou seja, uma atribuição de rótulos tradicionalmente chamados "cores" a elementos de um grafo sujeita a certas restrições.

1. **NP – COMPLETO**

NP-completo é um subconjunto de NP, o conjunto de todos os problemas de decisão cujas soluções podem ser verificadas em tempo polinomial; NP pode ser equivalentemente definida como o conjunto de problemas de decisão que podem ser solucionados em tempo polinomial em uma Máquina de Turing não determinística. Um problema p em NP também está em NPC Se e somente se todos os outros problemas em NP podem ser transformados em p em tempo polinomial.

O problema da mochila é um problema de optimização combinatória. O nome dá-se devido ao modelo de uma situação em que é necessário preencher uma mochila com objetos de diferentes pesos e valores. O objetivo é que se preencha a mochila com o maior valor possível, não ultrapassando o peso máximo. O problema da mochila é um dos 21 problemas NP-completos de Richard Karp, exposto em 1972. A formulação do problema é extremamente simples, porém sua solução é mais complexa. Este problema é a base do primeiro algoritmo de chave pública (chaves assimétricas).

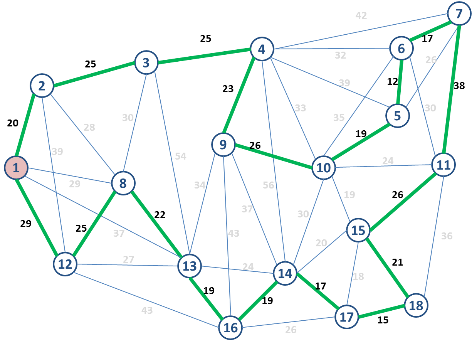


Representação do problema da mochila

1. **NP – HARD**

NP-difícil (ou NP-hard, ou NP-complexo) na teoria da complexidade computacional, é uma classe de problemas que são, informalmente, "Pelo menos tão difíceis quanto os problemas mais difíceis em NP". Um problema H é NP-difícil se e somente se existe um problema NP-completo L que é Turing-redutível em tempo polinomial para H, em outras palavras, L pode ser resolvido em tempo polinomial por uma Máquina de Turing não determinística com um oráculo para H. Informalmente, podemos pensar em um algoritmo que pode chamar tal Máquina de Turing Não-Determinística como uma sub-rotina para resolver H, e resolver L em tempo polinomial, se a chamada da sub-rotina leva apenas um passo para computar. Problemas NP-difíceis podem ser de qualquer tipo: problemas de decisão, problemas de pesquisa ou problemas de otimização.

Um exemplo de um problema NP-difícil é o problema de decisão da soma de subconjuntos, que é o seguinte: dado um conjunto de números inteiros, pode algum subconjunto não-vazio deste somar zero? Isso é uma questão do tipo sim/não, e acontece de ser NP-completo. Outro exemplo de um problema NP-difícil é o problema de otimização de se encontrar a rota de menor custo através de todos os nós de um grafo ponderado. Isto é conhecido como o problema do caixeiro viajante, no qual é representado pela imagem abaixo.



Problema do caixeiro viajante