

Estrutura de Dados e Algoritmos

Análise de Complexidade

- **Algoritmo:** conjunto claramente especificado de instruções a seguir para resolver um problema;

- **Análise de algoritmos:**
 - Provar que um algoritmo está correto;
 - Determinar recursos exigidos por um algoritmo (tempo, espaço, etc.);
 - Comparar os recursos exigidos por diferentes algoritmos que resolvem o mesmo problema (um algoritmo mais eficiente exige menos recursos para resolver o mesmo problema)
 - Prever o crescimento dos recursos exigidos por um algoritmo à medida que o tamanho dos dados de entrada cresce;

- Complexidade espacial de um programa ou algoritmo: espaço de memória que necessita para executar até ao fim
 - $S(n)$ - espaço de memória exigido em função do tamanho (n) da entrada;
- Complexidade temporal de um programa ou algoritmo: tempo que demora a executar (tempo de execução)
 - $T(n)$ - tempo de execução em função do tamanho (n) da entrada;

- Complexidade \uparrow versus Eficiência \downarrow ;
- Por vezes estima-se a complexidade para o "melhor caso" (pouco útil), o "pior caso" (mais útil) e o "caso médio" (igualmente útil);

- A Complexidade Computacional é um ramo da Matemática Computacional que estuda a eficiência dos algoritmos.
- Para medir a eficiência de um algoritmo freqüentemente usamos um tempo teórico que o programa leva para encontrar uma resposta em função dos dados de entrada.

- PROBLEMA DO CAIXEIRO VIAJANTE:
 - “Suponha que um caixeiro viajante tenha de visitar n cidades diferentes, iniciando e encerrando sua viagem na primeira cidade. Suponha, também, que não importa a ordem com que as cidades são visitadas e que de cada uma delas pode-se ir diretamente a qualquer outra. O problema do caixeiro viajante consiste em descobrir a rota que torna mínima a viagem total”.

- O problema do caixeiro viajante é um problema de otimização combinatória.
 - Como transforma-lo num problema de enumeração?
 - Como determinar todas as rotas do caixeiro?
 - Como saber qual delas é a menor?
- SOLUÇÃO: São $(n-1)!$ Rotas
 - É um trabalho fácil para a máquina?

n	Rotas por segundo	$(n - 1)!$	Cálculo total
5	250 milhões	24	insignificante
10	110 milhões	362 880	0.003 seg
15	71 milhões	87 bilhões	20 min
20	53 milhões	1.2×10^{17}	73 anos
25	42 milhões	6.2×10^{23}	470 milhões de anos