Complexidade Assintótica

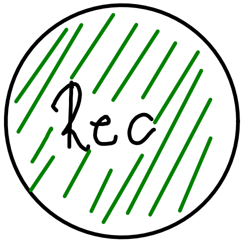
Complexidade Assintótica

MT

Não RE

RE

- Linguagem Recursiva
  - Problema decidível
  - Possui solução algorítmica



  - Solução é tratável computacionalmente?

P e NP

Complexidade Assintótica

- O fato de um algoritmo resolver um dado problema não significa que seja aceitável na prática
- Exemplo:
  - Caixeiro Viajante
    - Um **caixeiro viajante** deseja visitar n cidades de tal forma que sua viagem inicie e termine em uma mesma cidade, e cada cidade deve ser visitada uma única vez
    - Supondo que sempre há uma estrada entre duas cidades quaisquer, o problema é encontrar a menor rota para a viagem.

Complexidade Assintótica

$n = 4$

– A figura ilustra o exemplo para quatro cidades *c1*, *c2*, *c3*, *c4*, em que os números nos arcos indicam a distância entre duas cidades.

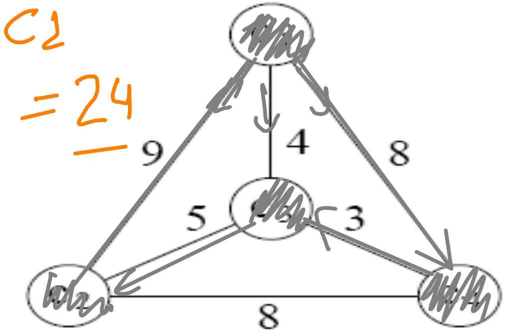
– O percurso <*c1*, *c3*, *c4*, *c2*, *c1*> é uma solução para o problema, cujo percurso total tem distância 24.

$C_1 \to C_3 \to C_4 \to C_2 \to C_1$

$4 + 3 + 8 + 9 = 24$

$C_1 \to C_4 \to C_3 \to C_2 \to C_1$

$8 + 3 + 5 + 9 = 25$



$C_1 \to C_2$, $C_1 \to C_3$, $C_1 \to C_4$

$(n-1) = 3$ | $(n-2) = 2$ | $(n-3) = 1$

$3 \cdot 2 \cdot 1 = 6$

$n = 4$

$(n-1)!$

$n = 10$

$\underbrace{9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}$

↳ não tratável

Complexidade Assintótica

- Um algoritmo simples seria verificar todas as rotas e escolher a menor delas.

- Há (n - 1)! rotas possíveis e a distância total percorrida em cada rota envolve n adições, logo o número total de adições é n!

- Suponha agora 50 cidades: o número de adições seria
$50! \approx 10^{64}$
- Em um computador que executa 10^9 adições por segundo, o tempo total para resolver o problema com 50 cidades seria maior do que 10^{45} séculos só para executar as adições

Complexidade Assintótica

2 EV
1 EV

$O(n)$
$\Omega(1)$

- Classes de complexidade

v = | 8 | 1 | 9 | 3 | 4 | 5 | 6 | 7 | 2 |

Sim

$n = 9$

| 3 | 6 | 8 | 9 |

$n = 4$

exponencial

fatorial

$n!$

polinomial

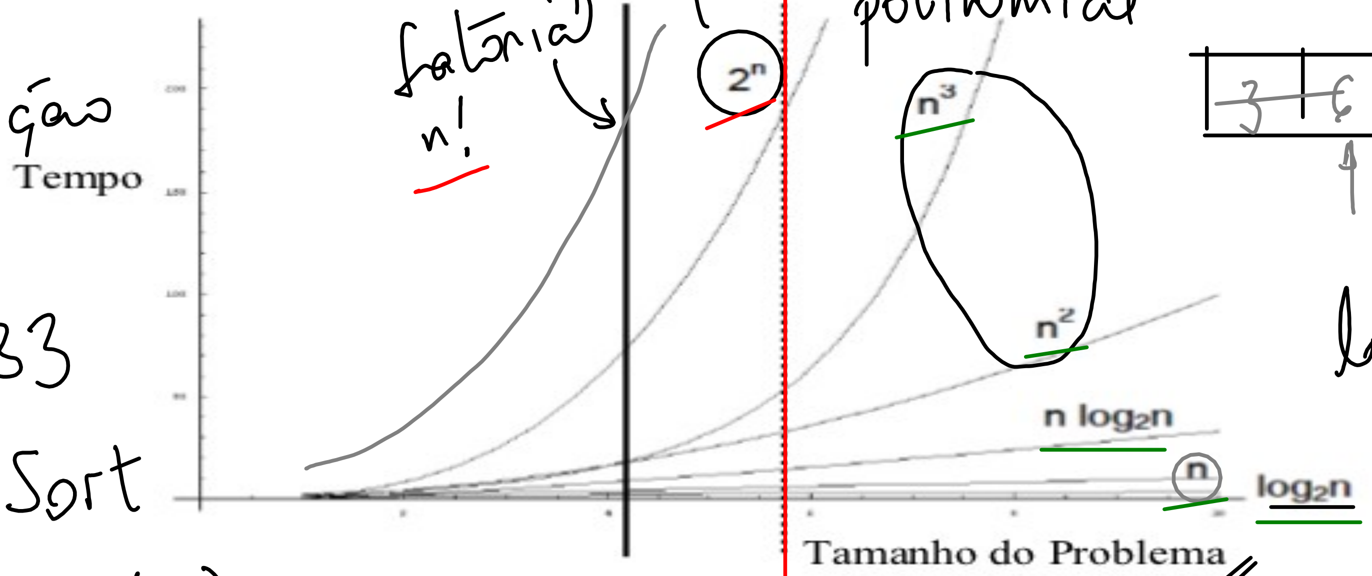
$n \log n$
↳ Ordenação

$n = 10$

$10 \cdot \log_2 10 = 33$

$n^2 \to$ BubbleSort

$O(n^2)$ e $\Omega(n^2) \Rightarrow \Theta(n^2)$



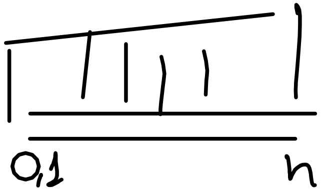
$\log_2 n = 2+1$

$2^2 = n$

$n = 4$

$n = 1000$ $\log_2 1000 \cong 10$

Complexidade Assintótica



$2 \cdot 2 \cdot 2 \cdots$

2^n

- Classes de Complexidade

n

	10	100	10^3	10^4	10^5	10^6
$\log_2 n$	3	6	9	13	16	19
n	10	100	1000	10^4	10^5	10^6
$n \log_2 n$	30	664	9965	10^5	10^6	10^7
n^2	100	10^4	10^6	10^8	10^{10}	10^{12}
n^3	10^3	10^6	10^9	10^{12}	10^{15}	10^{18}
2^n	10^3	10^{30}	10^{300}	10^{300}	10^{3000}	10^{300000}

tratável

n = 10

1 ano = 365 × 24 × 60 × 60 ≈ 3 × 10^7 segundos
1 século ≈ 3 × 10^9 segundos
1 milénio ≈ 3 × 10^{10} segundos

```
for (i=0; i<n; i++){
    for (j=0; j<n; j++){
        op;
        if (COND){break;}
} }
```

$O(n^2)$ e $\Omega(n)$

COND = Falso   Cond = Verd

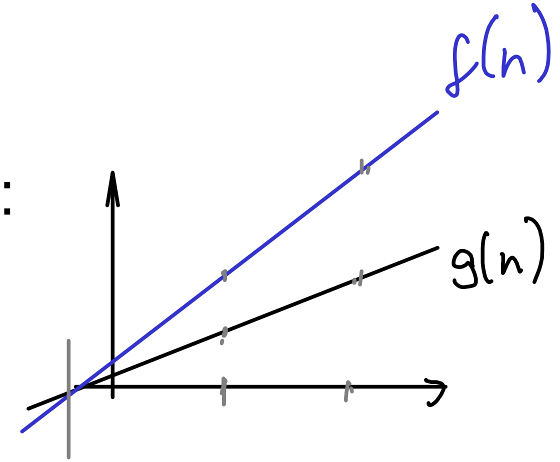
Complexidade Assintótica

- Custo Assintótico
  - O custo assintótico de uma função f(n) representa o limite do comportamento de custo quando n cresce
  - Notação O → pior caso
  - Notação Ω → melhor caso
  - Notação Θ

Complexidade Assintótica

- Notação O
  - Trazida da matemática por Knuth (1968):
  - g(n) = O(f(n))
  - Lê-se:
    - g(n) é de ordem no máximo f(n)
    - f(n) domina assintoticamente g(n)
    - (f(n) é um limite assintótico superior para g(n))

log n
n
n log n
⋮



Formalmente:

$$g(n) = O(f(n)),\ c > 0 \text{ e } n0 \mid 0 \leq g(n) \leq c.f(n),\ \forall\, n \geq n0$$

Complexidade Assintótica

- Se dizemos que $g(n) = O(n2)$, significa que existem constantes c e m | $g(n) <= c. n2$ para $n >= m$
- Exemplo:

  $g(0) = 1$

  $g(1) = 4$

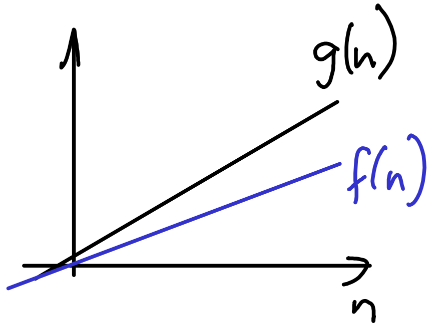
  $g(n) = (n+1)^2$

  $(n+1)^2 <= cn^2$

  $n^2 + 2n + 1 <= cn^2$

  $2n + 1 <= (c-1)n^2$

Complexidade Assintótica



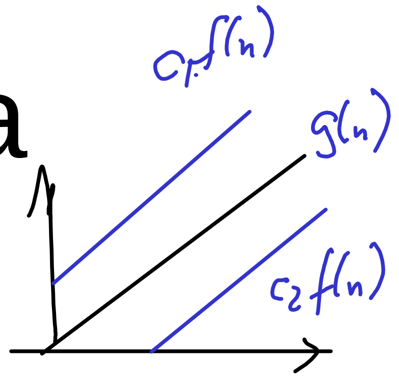
- Notação Ω
  - Ω define um limite inferior para a função, por um fator constante.
    - Escreve-se g(n) = Ω(f(n)), se existirem constantes positivas c e n0 | para n >= n0, o valor de g(n) é maior ou igual a c.f(n)
  - Neste caso, diz-se que f(n) é um limite assintótico inferior para g(n).

$\Omega(n)$ e $O(n^2)$

Formalmente:

g(n) = Ω(f(n)), c > 0 e n0 | 0 <= c.f(n) <= g(n), ∀ n >= n0

Complexidade Assintótica



- Notação Θ
  - A notação Θ limita a função por fatores constantes
    - g(n) = Θ(f(n)) se existirem constantes positivas c1 e c2 e n0 tais que para n >= n0, o valor de g(n) está sempre entre c1.f(n) e c2.f(n) inclusive.

$\Theta(f(n)) \rightarrow O(f(n))$ e $\Omega(f(n))$

Formalmente:

g(n) = Θ(f(n)), c1 > 0 e c2 > 0 e n0 |

0 <= c1.f(n) <= g(n) <= c2.f(n), ∀ n >= n0

Complexidade Assintótica

Complexidade Assintótica

Complexidade Assintótica