

Complexidade de Algoritmos

Algoritmos e Programação de Computadores

Guilherme N. Ramos
gnramos@unb.br

2016/1



Complexidade de Algoritmos

Objetivos do projeto de um algoritmo

- 1 resolver corretamente a tarefa; e
- 2 ser eficiente.

"Otimização prematura é a raiz de todos os males."

Donald Knuth

Quanto tempo para ler $2 \cdot 10^{46}$ tweets?

Supondo 5 segundos por *tweet*, e 16 horas de leitura por dia... seriam necessários $4.8 \cdot 10^{39}$ anos. Mas apenas $3.4 \cdot 10^{27}$ milênios considerando que metade das pessoas capazes de ler no mundo ajudem...

"No extremo norte de uma terra chamada Svithjod, há uma pedra. Ela tem 100 milhas de altura, e 100 milhas de largura. Uma vez a cada 1000 anos, um passarinho vai a pedra afiar seu bico. Quando a pedra tiver sido completamente desgastada, então um único dia da eternidade terá passado."

Hendrik Willem Van Loon

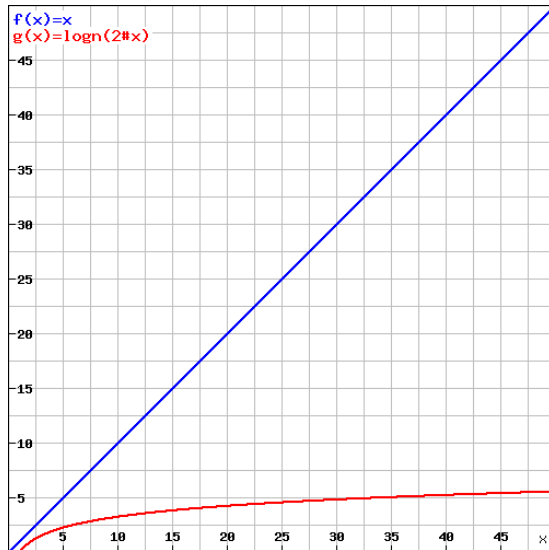
Se 1 "dia eterno" dura 10^{32} anos, então seriam necessários apenas 50 "minutos eternos"... ou 130 "milênios eternos" se estiver só.

Complexidade de Algoritmos

Quanto tempo demora a execução do programa? Basta executar e medir...

```
1 int i, n;
2
3 scanf("%d", &n); /* 1 es */
4
5 if(n < 0) /* 1 cmp */
6     printf("Valor inválido!"); /* 1 es */
7 else {
8     printf("Valor válido!"); /* 1 es */
9     printf("Mas vou mudá-lo..."); /* 1 es */
10    printf("... para facilitar a demonstração."); /* 1 es */
11 }
12
13 n = 4; /* 1 atr */
14 for(i = 0; i < n; ++i) /* n+1 atr, n+1 cmp, n ad */
15     printf("i = %d\n", i); /* n es */
```

Complexidade de Algoritmos



Complexidade de Algoritmos

Esta análise depende de:

- velocidade do computador;
- especificidades da linguagem; e
- tamanho da entrada.

"O sistema de análise matemática [...] constitui o maior avanço técnico do pensamento exato."

John von Neumann

Complexidade de Algoritmos

Notação Assintótica (Grande-O/Big-O Notation)

Notação matemática usada para analisar o comportamento de funções

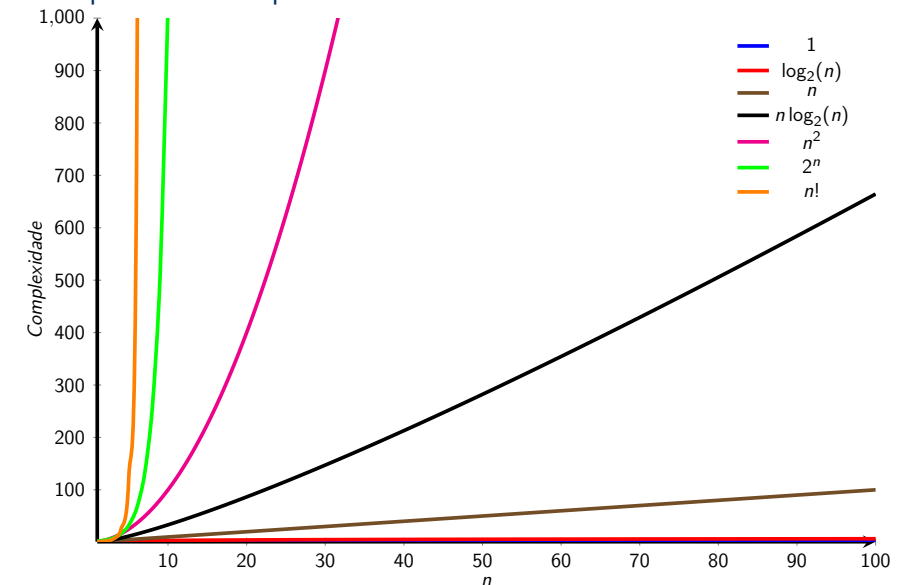
- utilizada para descrever o uso de recursos computacionais

Permite:

- prever o comportamento do algoritmo; e
- determinar qual algoritmo utilizar.

Complexidade de Algoritmos

Principais classes de problemas



Complexidade de Algoritmos

Casos de estudo da função de custo:

$\Omega(n)$ Melhor: definida pelo menor número de passos executados para qualquer instância de tamanho n

$O(n)$ Pior: definida pelo maior número de passos executados para qualquer instância de tamanho n

Complexidade Assintótica

- Para pequenos valores de n , a maioria dos algoritmos não representa problemas.
- Estuda-se a complexidade somente para **grandes valores de n** .
- O comportamento assintótico representa o limite do comportamento do custo quando n cresce.

Ordenação

Busca binária é mais eficiente que a sequencial, mas exige que o vetor esteja ordenado (após a aplicação de um algoritmo). Qual a melhor estratégia para buscar um elemento? E vários elementos?

O custo pode ser *amortizado*...

Ordenação

A busca binária separa o vetor e descarta uma das partes, diminuindo o esforço necessário...

Dividir para Conquistar

- 1 Dividir o problema em versões menores;
- 2 resolver estas versões recursivamente; e
- 3 combinar os resultados de forma a obter a solução completa.

Ordenação

Busca binária é mais eficiente que a sequencial, mas exige que o vetor esteja ordenado (após a aplicação de um algoritmo) e tem ordem logarítmica... Seria melhor se houvesse uma forma mais eficiente³...

³Cujo custo também pode ser *amortizado*...