
Algoritmos e Estruturas de Dados

Aula 4 :: Busca e Ordenação

Filipe Cordeiro (filipe.rolim@ufrpe.br)



UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

Crédito slides: Prof. Francisco Simões (francisco.simoes@ufrpe.br)

Busca vs Ordenação

Busca

- Buscar um item em um banco de dados

	Chen	3	A	991-878-4944	308 Blair
	Rohde	2	A	232-343-5555	343 Forbes
	Gazsi	4	B	766-093-9873	101 Brown
item →	Furia	1	A	766-093-9873	101 Brown
	Kanaga	3	B	898-122-9643	22 Brown
	Andrews	3	A	664-480-0023	097 Little
key →	Battle	4	C	874-088-1212	121 Whitman

Ordenação

- Organizar os N itens or ordem crescente

Andrews	3	A	664-480-0023	097 Little
Battle	4	C	874-088-1212	121 Whitman
Chen	3	A	991-878-4944	308 Blair
Furia	1	A	766-093-9873	101 Brown
Gazsi	4	B	766-093-9873	101 Brown
Kanaga	3	B	898-122-9643	22 Brown
Rohde	2	A	232-343-5555	343 Forbes

Busca

25	4	19	10	99	84	13	2	30	50
----	---	----	----	----	----	----	---	----	----

Para encontrar um elemento, precisamos percorrer nossa lista de elementos para informar se ele está ou não na lista (e consequentemente informar sua posição)

P1. O elemento 13 está na lista?

R. Sim, na posição 6

Método mais natural: **Busca linear**

Percorre a lista do início ao fim e verifica cada elemento

Isto é eficiente?

Busca

25	4	19	10	99	84	13	2	30	50
----	---	----	----	----	----	----	---	----	----

Para encontrar um elemento, precisamos percorrer nossa lista de elementos para informar se ele está ou não na lista (e consequentemente informar sua posição)

P1. O elemento 13 está na lista?

R. Sim, na posição 6

Método mais natural: **Busca linear**

Percorre a lista do início ao fim e verifica cada elemento

Isto é eficiente?

E se tivéssemos uma lista ordenada?

Busca binária!

Busca linear vs busca binária

25	4	19	10	99	84	13	2	30	50
25	4	19	10	99	84	13	2	30	50
25	4	19	10	99	84	13	2	30	50
25	4	19	10	99	84	13	2	30	50
25	4	19	10	99	84	13	2	30	50
25	4	19	10	99	84	13	2	30	50
25	4	19	10	99	84	13	2	30	50
25	4	19	10	99	84	13	2	30	50
25	4	19	10	99	84	13	2	30	50

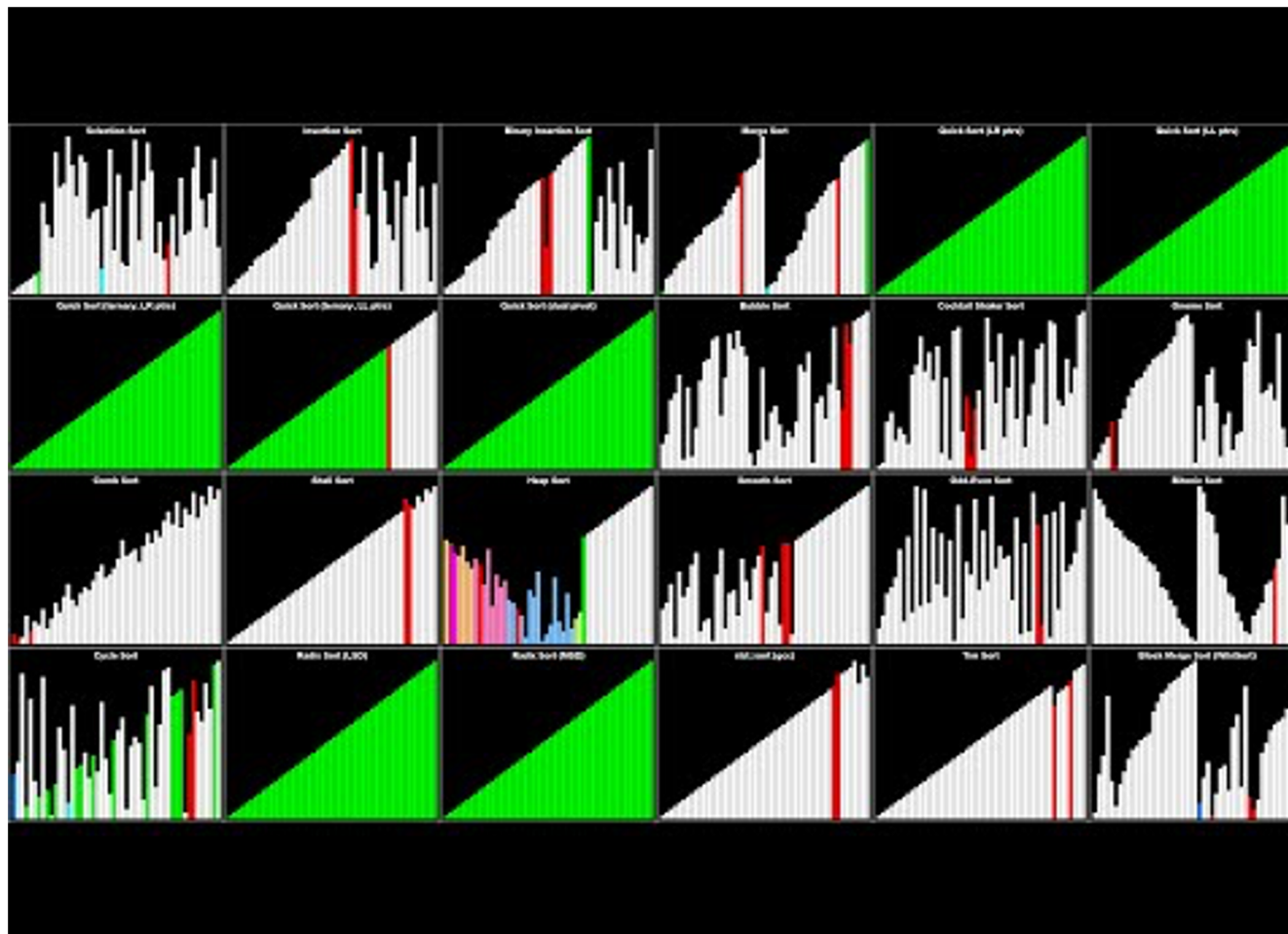
25	4	19	10	99	84	13	2	30	50
Ordена									
2	4	10	13	19	25	30	50	84	99
2	4	10	13	19	25	30	50	84	99
2	4	10	13	19	25	30	50	84	99
2	4	10	13	19	25	30	50	84	99
2	4	10	13	19	25	30	50	84	99

Atividade

- Implemente os algoritmos de Busca Binária e Busca Sequencial.

Como ordenar?

**Pense em como seria um algoritmo
de ordenação**



Visualização: <https://www.youtube.com/watch?v=BeoCbJPuvSE>

Som de cada algoritmo: <https://www.youtube.com/watch?v=kPRA0W1kECg&t=185s>

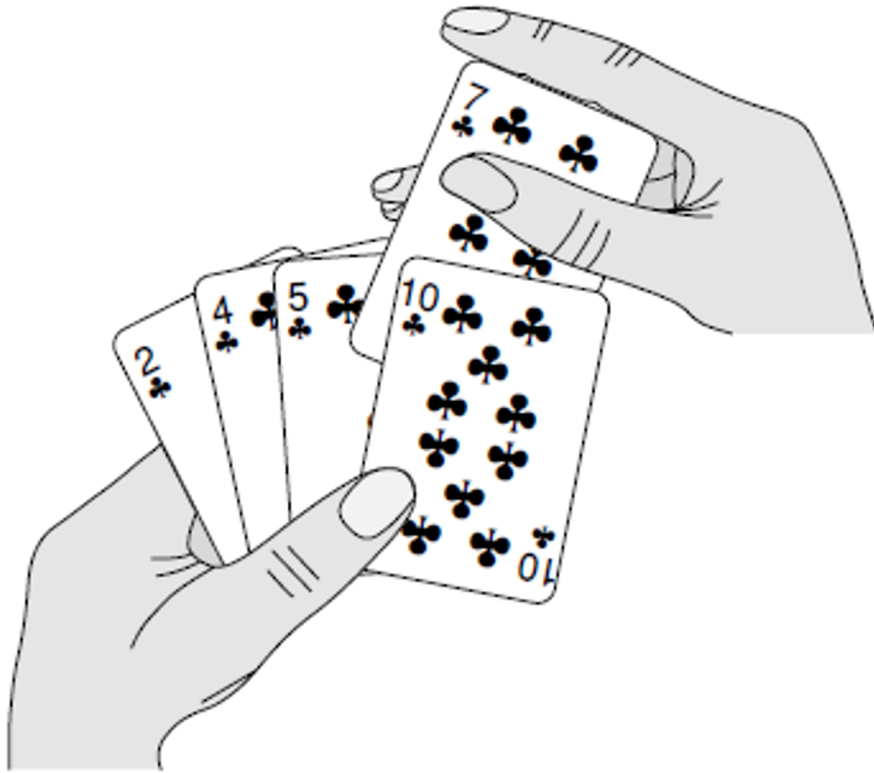
Ordenar sequência de elementos

25	4	19	10	99	84	13	2	30	50
----	---	----	----	----	----	----	---	----	----



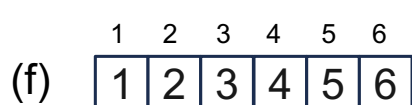
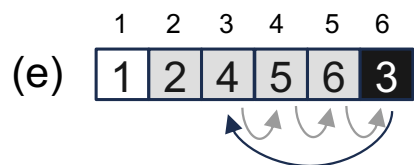
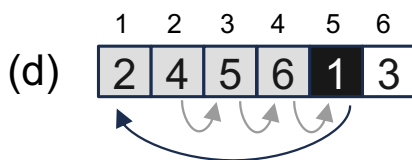
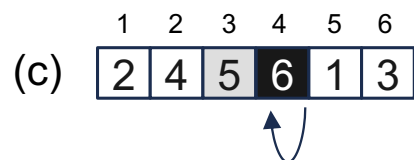
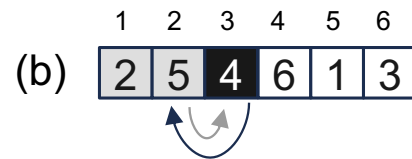
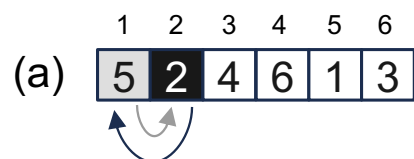
2	4	10	13	19	25	30	50	84	99
---	---	----	----	----	----	----	----	----	----

Insertion Sort (Problema)



Insertion Sort (Intuição)

1	2	3	4	5	6
5	2	4	6	1	3



Insertion Sort

(Intuição)

Atividade

- Execute o InsertionSort para ordenar o seguinte vetor
- [5 10 3 7 8 6 1 2 9 2]

INSERTION-SORT (A, n) *//* $A[1 \dots n]$

for $j \leftarrow 2$ **to** n

do $key \leftarrow A[j]$

$i \leftarrow j - 1$

while $i > 0$ and $A[i] > key$

do $A[i+1] \leftarrow A[i]$

$i \leftarrow i - 1$

$A[i+1] = key$

Insertion Sort (Pseudocódigo)

Tipos de Análise

Pior caso (mais comum)

$T(n)$ = tempo máximo que o algoritmo demora para executar

Caso médio

$T(n)$ = tempo médio esperado considerando todas as possíveis entradas (Baseado em estatística)

Melhor caso

$T(n)$ = tempo mínimo que demora para executar (para alguma entrada)

Insertion Sort (Complexidade)

Qual o tempo que o Insertion Sort demora para ordenar?

Velocidade depende de máquina, processos, memória...

Como analisar independentemente?

Análise Assintótica ou Funcional!

Análise acerca da velocidade de crescimento da função $T(n)$ quando n tende a infinito $n \rightarrow \infty$

Insertion Sort (Complexidade)

Qual a complexidade do Insertion Sort?

$O(n^2)$

Ele é um algoritmo rápido?

Moderado, para entradas pequenas

Não, para entradas grandes

Insertion Sort (Complexidade)

Qual é o pior tipo de entrada
que pode ser passada para o
Insertion Sort?

Insertion Sort (Para pensar)

Atividades

1. Implemente a função `insertionSort` que ordena um vetor
2. Implemente a função `buscaBinaria` que realiza a busca binária considerando um vetor ordenado
3. Execute diversas buscas aleatórias num vetor ($tam > 1000$) e compare sua eficiência:
 - a. Utilizando sempre a busca linear
 - b. Primeiro ordenando e sempre utilizando `buscaBinaria` para buscar

Atividade

Estude o material do prof. Paulo Feafiloff sobre ordenação

<https://www.ime.usp.br/~pf/algoritmos/aulas/ordena.html>