

INDÚSTRIA 4.0

Processamento de Big Data



Aula #2 - Algoritmos: Busca e Ordenação

EDUARDO CUNHA DE ALMEIDA

Agenda

- Busca Sequencial e Binária
- Ordenação: Seleção e QuickSort
- Listas e SkipList
- Árvores Balanceadas
- Hash
- Compressão

Agenda

- **Busca Sequencial e Binária**
- Ordenação: Seleção e QuickSort
- Listas e SkipList
- Árvores Balanceadas
- Hash
- Compressão

Busca sequencial: busca(42)

- Considere o conjunto de dados abaixo (não ordenado):

Table 1

Posição	0	1	2	3	4	5	6
Valor	16						



Sabemos também que o conjunto tem 7 elementos (0->6) !!!!

ex.: Buscar 42 requer várias comparações

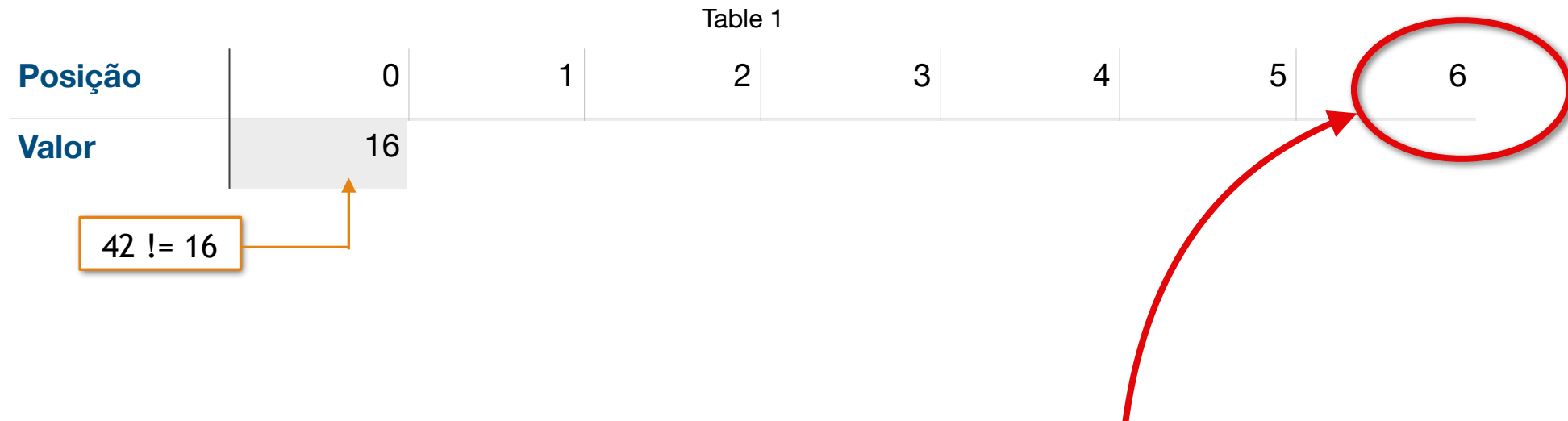
Buscar o menor valor requer “n-1” comparações (ineficiente)

Busca sequencial: busca(42)

- Considere o conjunto de dados abaixo (não ordenado):

Table 1

Posição	0	1	2	3	4	5	6
Valor	16						



Sabemos também que o conjunto tem 7 elementos (0->6) !!!!

ex.: Buscar 42 requer várias comparações

Buscar o menor valor requer “n-1” comparações (ineficiente)

Busca sequencial: busca(42)

- Considere o conjunto de dados abaixo (não ordenado):

Table 1

Posição	0	1	2	3	4	5	6
Valor	16	23					

42 != 16

42 != 23

ex.: Buscar 42 requer várias comparações

Buscar o menor valor requer “n-1” comparações (ineficiente)

Busca sequencial: busca(42)

- Considere o conjunto de dados abaixo (não ordenado):

Table 1

Posição	0	1	2	3	4	5	6
Valor	16	23	5				

42 != 16

42 != 23

42 != 5

ex.: Buscar 42 requer várias comparações

Buscar o menor valor requer “n-1” comparações (ineficiente)

Busca sequencial: busca(42)

- Considere o conjunto de dados abaixo (não ordenado):

Table 1

Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4

42 != 16

42 != 23

42 != 5

42 = 42

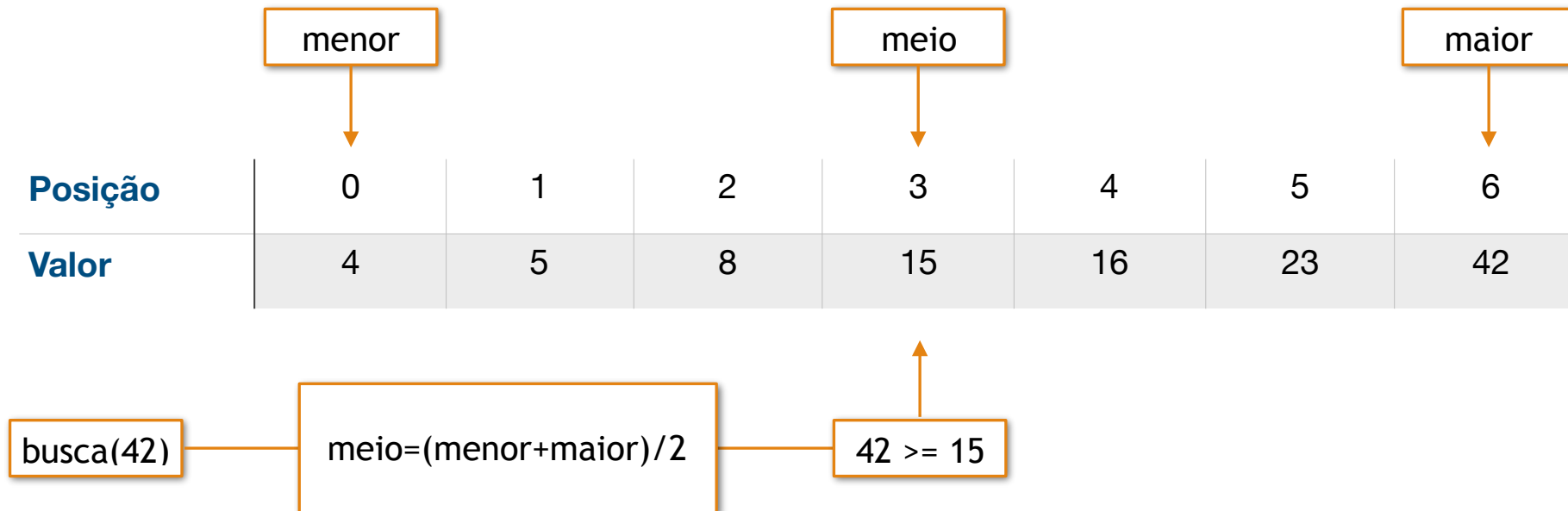
ex.: Buscar 42 requer várias comparações

Buscar o menor valor requer “n-1” comparações (ineficiente)

Como evitar a busca sequencial?

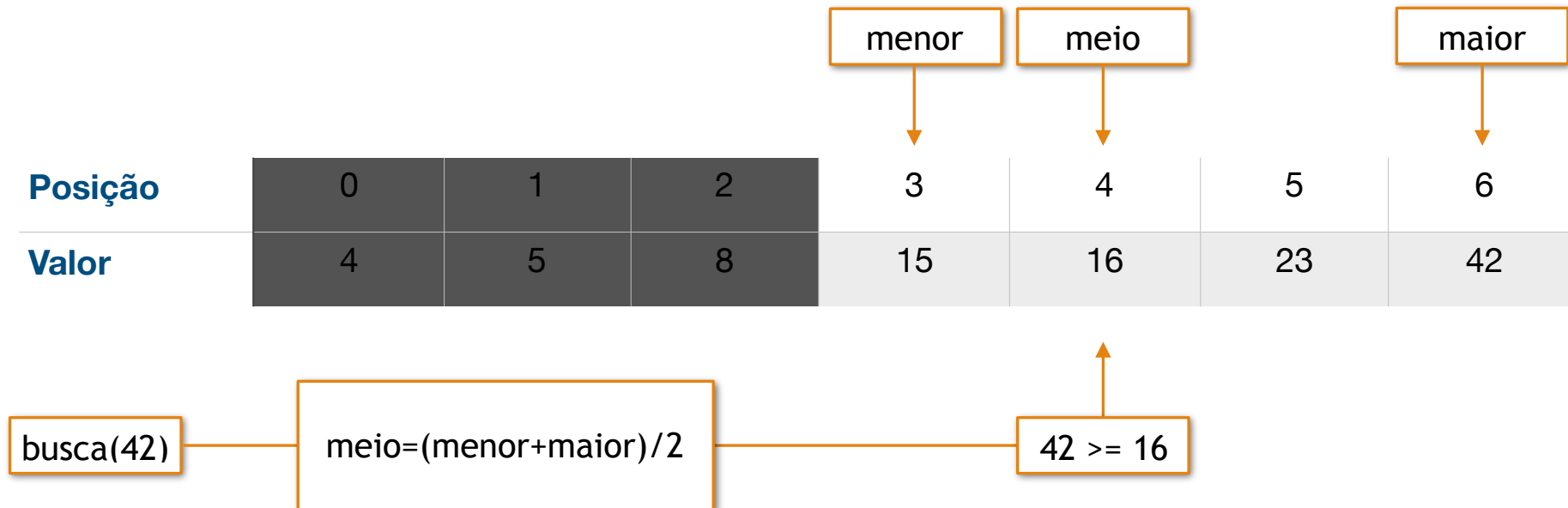
Busca binária

- Considere o conjunto de dados abaixo (ordenado):



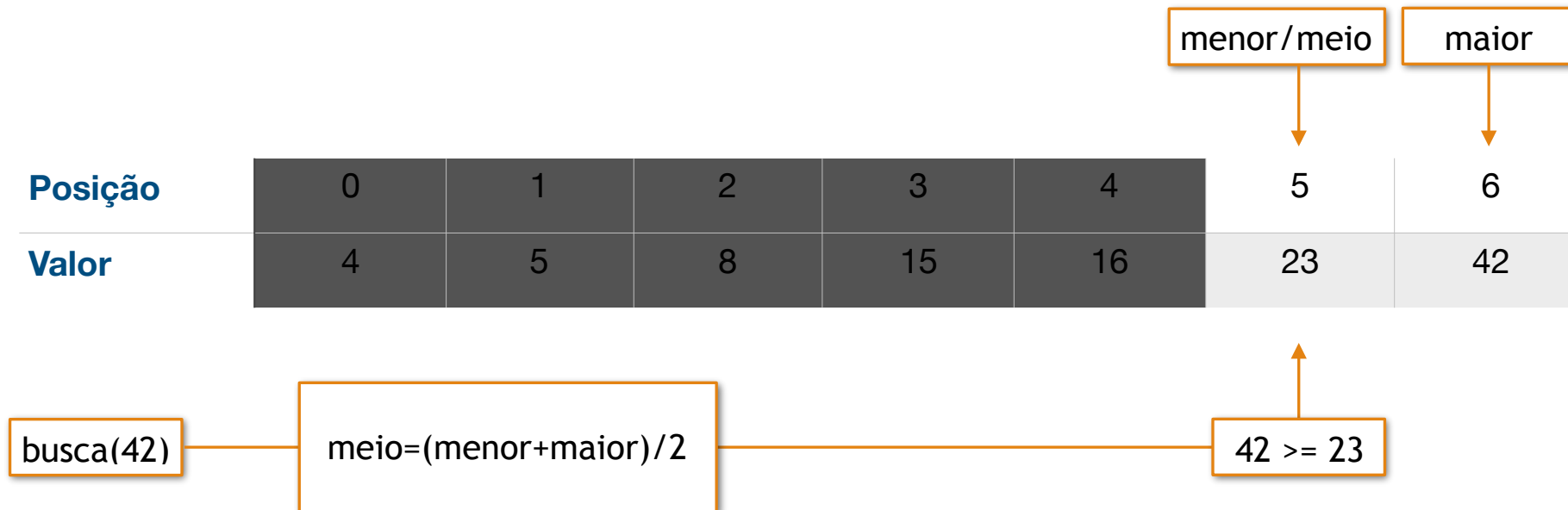
Busca binária

- Considere o conjunto de dados abaixo (ordenado):



Busca binária

- Considere o conjunto de dados abaixo (ordenado):



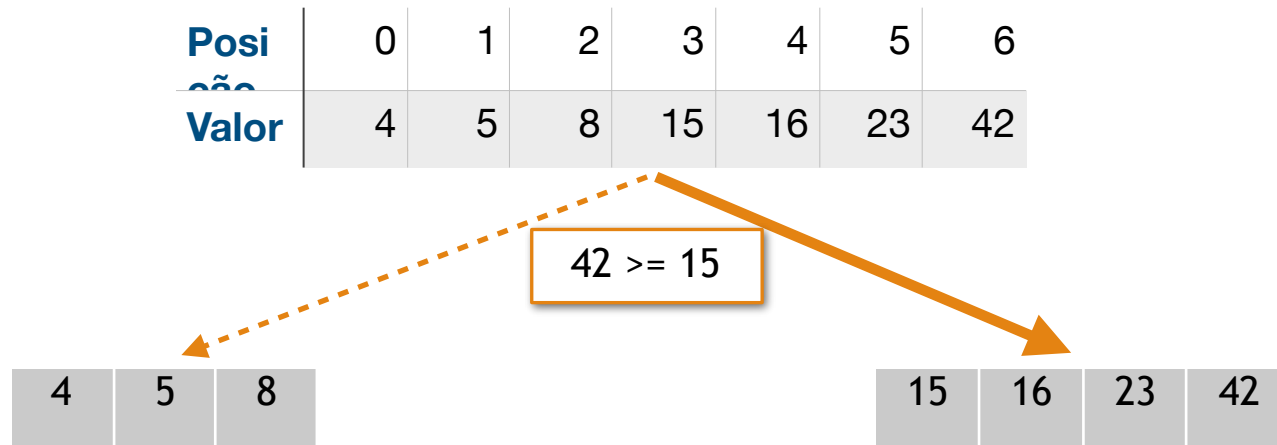
Busca binária

- Considere o conjunto de dados abaixo (ordenado):

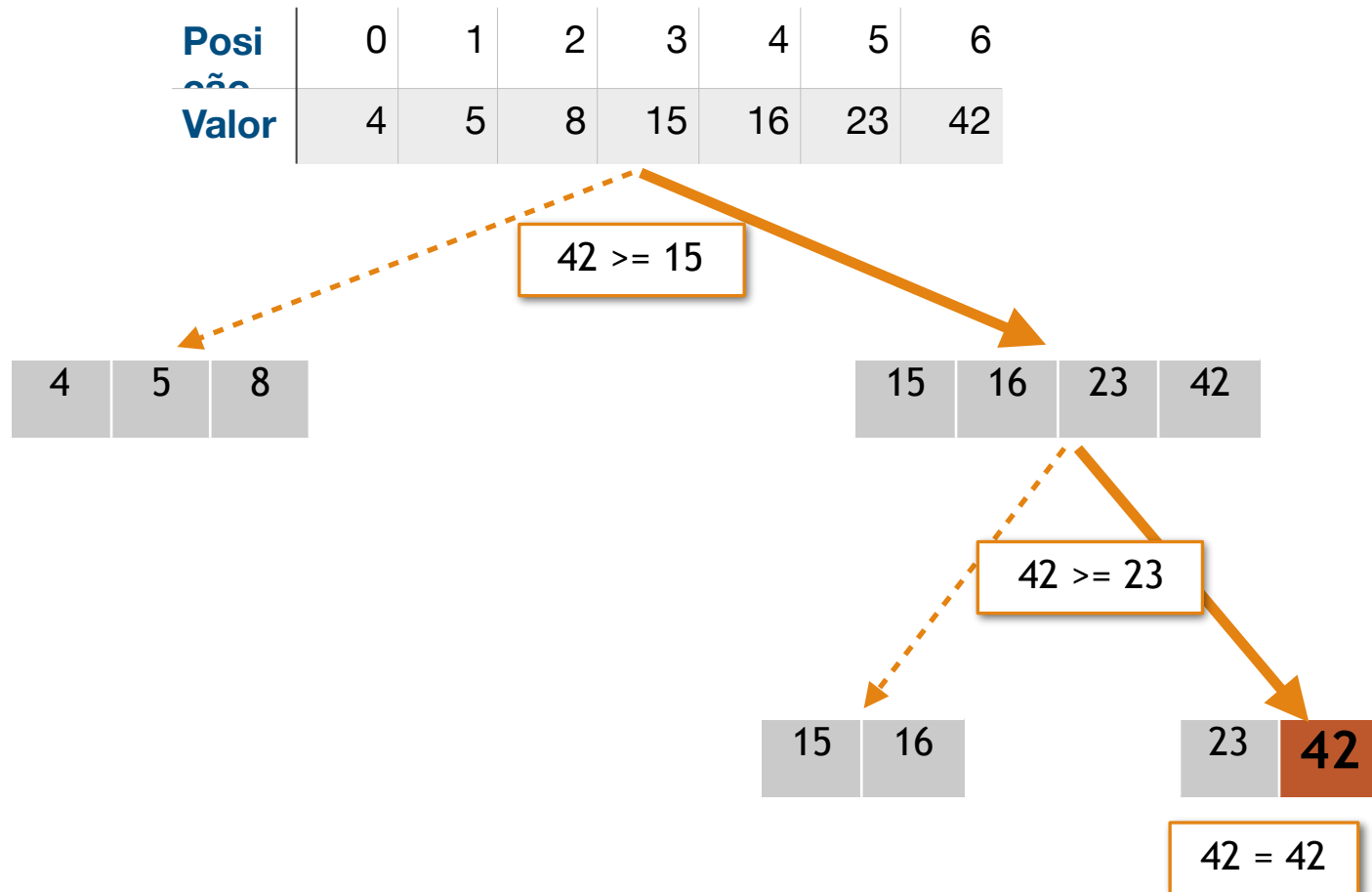
Posição	0	1	2	3	4	5	6
Valor	4	5	8	15	16	23	42



Resumindo



Resumindo



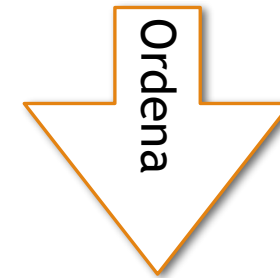
Agenda

- Busca Sequencial e Binária
- **Ordenação: Seleção e QuickSort**
- Listas e SkipList
- Árvores Balanceadas
- Hash
- Compressão

Ordenação

Facilita a busca, pois em coleções de dados, como em bancos de dados, as tuplas (ou linhas) não possuem uma ordem específica

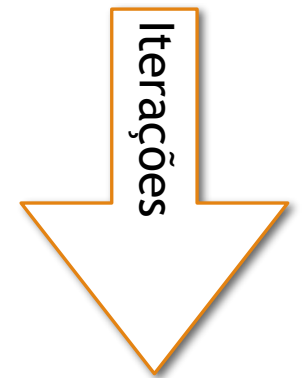
Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4



Posição	0	1	2	3	4	5	6
Valor	4	5	8	15	16	23	42

Ordenação (seleção)

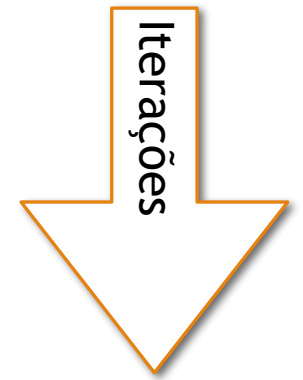
Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4



1

Ordenação (seleção)

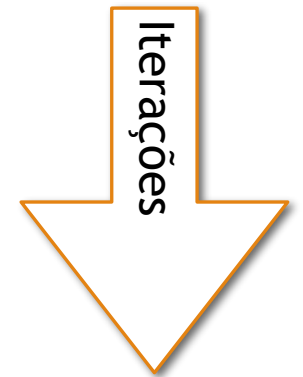
Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
	4	23	5	42	15	8	16



2

Ordenação (seleção)

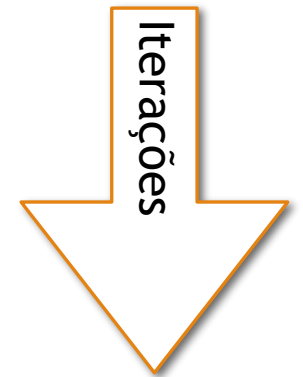
Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
	4	23	5	42	15	8	16
	4	5	23	42	15	8	16



3

Ordenação (seleção)

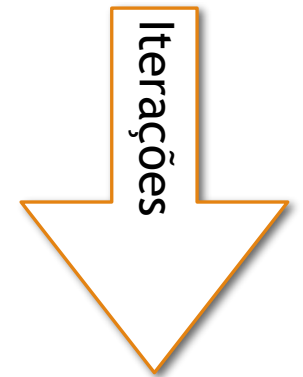
Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
	4	23	5	42	15	8	16
	4	5	23	42	15	8	16
	4	5	8	42	15	23	16



4

Ordenação (seleção)

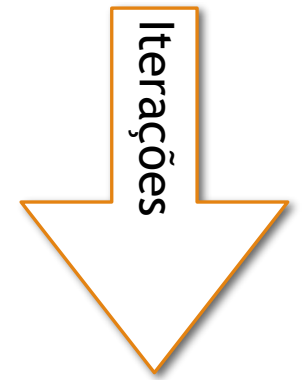
Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
4		23	5	42	15	8	16
4		5	23	42	15	8	16
4		5	8	42	15	23	16
4		5	8	15	42	23	16



5

Ordenação (seleção)

Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
4	23	5	42	15	8	16	
4	5	23	42	15	8	16	
4	5	8	42	15	23	16	
4	5	8	15	42	23	16	
4	5	8	15	16	23	42	
4	5	8	15	16	23	42	



6

Ordenação QuickSort

Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4

elemento pivot

Iterações

Ordenação QuickSort

Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
	5	4	8	15	16	23	42

elemento pivot

Iterações

1

Ordenação QuickSort

Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
	5	4	8	15	16	23	42



$n/2$ $n/2$

Iterações

1

Ordenação QuickSort

Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
	5	4	8	15	16	23	42

elemento pivot

Iterações

1

Ordenação QuickSort

Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
5	4	4	8	15	16	23	42
4		5	8	15	16	23	42

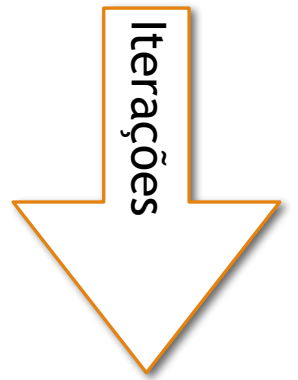
elemento pivot

Iterações

2

Ordenação QuickSort

Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
	5	4	8	15	16	23	42
	4	5	8	15	16	23	42
	n/4		n/4		n/4		n/4



2

Ordenação QuickSort

Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
5	4	4	8	15	16	23	42
4	4	5	8	15	16	23	42

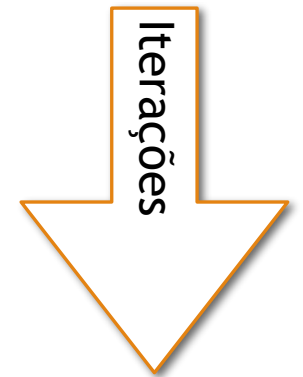
elemento pivot

Iterações

2

Ordenação QuickSort

Posição	0	1	2	3	4	5	6
Valor	16	23	5	42	15	8	4
	5	4	8	15	16	23	42
	4	5	8	15	16	23	42
	4	5	8	15	16	23	42



3

Ordenação

Melhor e pior caso em diferentes algoritmos

	Pior caso	Melhor caso
Seleção	n^2	n^2
QuickSort	$n^2/2$	$n(\log n)$
MergeSort	$n(\log n)$	$n/2(\log n)$

Será possível ter uma estrutura para armazenar dados sem necessidade de um algoritmo de ordenação?

INDÚSTRIA 4.0

Processamento de Big Data



Aula #2 - Algoritmos: Busca e Ordenação

EDUARDO CUNHA DE ALMEIDA