

# Ordenação e Busca em Arquivos

---

Cristina D. A. Ciferri

Thiago A. S. Pardo

Leandro C. Cintra

M.C.F. de Oliveira

Moacir Ponti Jr.



# Exemplos de Busca

---

- Registros de tamanho fixo

M A R I A		R U A	b	1		S A O	b	C A R L O S		b	b	b	b	b	b	b
J O A O		R U A	b	A		R I O	b	C L A R O		b	b	b	b	b	b	b
P E D R O		R U A	b	X V		S A O	b	C A R L O S		b	b	b	b	b	b	b
A N T O N I A		R U A	b	X V	b	D E	b	M A I O		I B A T E		b				
A N A		R U A	b	A U G U S T O	b	P A I V A		I B A T E		b	b					

1. Recupere os dados do registro relativo ao **João**
2. Recupere os dados do registro relativo ao **Pedro**



# Exemplos de Busca

---

- Registros de tamanho variável

M A R I A | R U A b 1 | S A O b C A R L O S | # J O A O | R  
U A b A | R I O b C L A R O | # P E D R O | R U A b X V | S  
A O b C A R L O S | # A N T O N I A | R U A b X V b D E b M  
A I O | I B A T E | # A N A | R U A b A U G U S T O b P A I  
V A | I B A T E | #

3. Recupere os dados do registro relativo ao João

4. Recupere os dados do registro relativo ao Pedro



# Ordenação

---

- Facilita a busca
- Pode ajudar a diminuir o número de acessos a disco



# Exemplo de Busca

---

- Registros de tamanho fixo

A N A		R U A	b	A U G U S T O	b	P A I V A		I B A T E		b b				
A N T O N I A		R U A	b	X V	b	D E	b	M A I O		I B A T E		b		
J O A O		R U A	b	A		R I O	b	C L A R O		b b	b b	b b	b b	b b
M A R I A		R U A	b	1		S A O	b	C A R L O S		b b	b b	b b	b b	b b
P E D R O		R U A	b	X V		S A O	b	C A R L O S		b b	b b	b b	b b	b b

5. Recupere os dados do registro relativo ao João
6. Recupere os dados do registro relativo ao Pedro



# Busca Sequencial e Binária

---

- Busca sequencial
  - recupera cada registro do arquivo, verificando se os valores dos atributos satisfazem à condição de seleção
- Busca binária
  - recupera registros quando a condição de seleção envolve uma comparação de igualdade no atributo que determina a ordenação do arquivo



# Custos: Comparações

---

$$C_{\text{busca\_sequencial}} = n$$

- n: número de registros que são comparados
- todos os registros são varridos (pior caso)
- complexidade:  $O(n)$

$$C_{\text{busca\_binária}} = \log_2(n) + 1$$

- n: número de registros que são comparados
- complexidade:  $O(\log n)$



# Custos: Acessos a Disco

$$C_{\text{busca\_sequencial}} = b$$

- $b$ : número de blocos que contêm os registros
- todos os blocos são varridos

$$C_{\text{busca\_binária}} = \log_2(b) + \lceil s/bfr \rceil - 1$$

- $\log_2(b)$ : custo para localizar o primeiro registro
- $\lceil s/bfr \rceil$ : blocos ocupados pelos registros que satisfazem à condição de seleção
- 1: custo para recuperar o primeiro registro





# Como Ordenar o Arquivo

## - Ordenação em RAM -

---

- *Arquivo completo **cabe** em RAM*
- Estratégia
  - leitura de todos os registros armazenados em disco para a RAM
  - ordenação dos registros em RAM
    - escolha do campo base para ordenação
    - uso de um método de ordenação
  - escrita de todos os registros armazenados em RAM para o disco

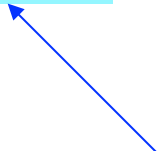


# Arquivo Ordenado

---

- Registros de tamanho fixo

A N A		R U A	b	A U G U S T O	b	P A I V A		I B A T E		b b		
A N T O N I A		R U A	b	X V	b	D E	b	M A I O		I B A T E		b
J O A O		R U A	b	A		R I O	b	C L A R O		b b b b b b b b		
M A R I A		R U A	b	1		S A O	b	C A R L O S		b b b b b b b		
P E D R O		R U A	b	X V		S A O	b	C A R L O S		b b b b b b b		



ordenação baseada em um determinado campo,  
usando suas chaves



# Chave (KEY)

---

- Está associada a um registro e permite a sua recuperação
- Chave **primária**
  - identifica univocamente um registro
  - não tem repetição
- Chave **secundária**
  - não identifica univocamente um registro
  - tem repetição



# Forma Canônica da Chave

---

- Uma única representação para uma determinada chave
- Exemplo
  - "Ana", "ANA", ou "ana" devem indicar o mesmo registro
  - Forma canônica: todos os caracteres em letras maiúsculas → ANA



# Como Ordenar o Arquivo

## - Ordenação em RAM -

---

- *Arquivo completo **não cabe** em RAM*
- Estratégia: ordenação por chave
  - conhecida como *keysorting*
  - armazena e ordena em RAM somente
    - *chaves* para ordenação
    - *RRNs* ou *byte offsets* dos registros

# Ordenação por Chave (*Keysorting*)

1. Leitura completa do arquivo de dados, trazendo para a RAM a chave e o RRN (ou *byte offset*) dos registros

<i>chave</i>	<i>RRN</i>	
M A R I A	0	M A R I A   R U A b 1   S ...
J O A O	1	J O A O   R U A b A   R I ...
P E D R O	2	P E D R O   R U A b X V   ...
A N T O N I A	3	A N T O N I A   R U A b X ...
A N A	4	A N A   R U A b A U G U S ...

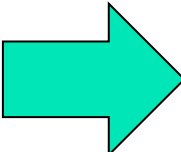
vetor em RAM

arquivo desordenado em disco

# Ordenação por Chave (*Keysorting*)

## 2. Ordenação do vetor em RAM

- uso de um método de ordenação

<i>chave</i>	<i>RRN</i>		<i>chave</i>	<i>RRN</i>
M A R I A	0		A N A	4
J O A O	1		A N T O N I A	3
P E D R O	2		J O A O	1
A N T O N I A	3		M A R I A	0
A N A	4		P E D R O	2

vetor desordenado em RAM

vetor ordenado em RAM

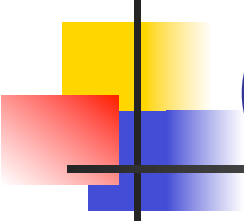


# Ordenação por Chave (*Keysorting*)

---

3. Para cada registro do vetor em RAM
  - obtém o **RRN**
  - identifica o *byte offset* do registro em disco ( $byte\ offset = RRN * tamRegistro$ )
  - **lê** o registro do arquivo em disco
    - arquivo de entrada desordenado
  - **escreve** o registro de forma ordenada em outro arquivo
    - arquivo de saída ordenado





# Ordenação por Chave (*Keysorting*)

---

- Arquivo ordenado em disco

A N A | R U A b A U G U S T O b P A I V A | I B A T E | b b  
A N T O N I A | R U A b X V b D E b M A I O | I B A T E | b  
J O A O | R U A b A | R I O b C L A R O | b b b b b b b b  
M A R I A | R U A b 1 | S A O b C A R L O S | b b b b b b  
P E D R O | R U A b X V | S A O b C A R L O S | b b b b b b



# Ordenação

---

A N A | R U A b A U G U S T O b P A I V A | I B A T E | b b  
A N T O N I A | R U A b X V b D E b M A I O | I B A T E | b  
J O A O | R U A b A | R I O b C L A R O | b b b b b b b b  
M A R I A | R U A b 1 | S A O b C A R L O S | b b b b b b  
P E D R O | R U A b X V | S A O b C A R L O S | b b b b b b

## ■ Perguntas

- e se a busca for feita por outro campo que não seja o campo ordenado?
- o que acontece quando BEATRIZ é inserida?



# Pensando em Índices

---



- *Por que realizar a tarefa custosa de escrever em disco a versão ordenada do arquivo?*
- Solução melhor ....
  - grava-se a ordenação da chave em um novo arquivo (arquivo de **índice**)
  - realiza-se busca binária no arquivo de índice, e recupera-se o RRN ou *byte offset*
  - realiza-se acesso direto no arquivo original (arquivo de dados)