

EDCO4B

ESTRUTURAS DE DADOS 2

Aula 11 - Busca e Ordenação
em Arquivos

Prof. Rafael G. Mantovani

Licença

Este trabalho está licenciado com uma Licença CC BY-NC-ND 4.0:



maiores informações:

https://creativecommons.org/licenses/by-nc-nd/4.0/deed.pt_BR

Roteiro



- 1** Introdução
- 2** Busca Binária
- 3** Key Sorting
- 4** Revisão
- 5** Referências

Roteiro

- 1** Introdução
- 2** Busca Binária
- 3** Key Sorting
- 4** Revisão
- 5** Referências

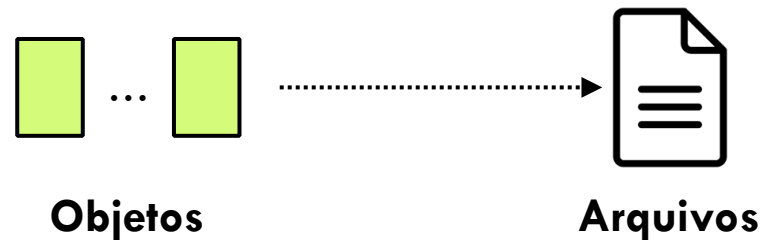
Introdução



- O que já fizemos:

Introdução

- O que já fizemos:
 - como organizar informações em arquivos



Introdução

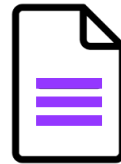
- O que já fizemos:
 - como organizar informações em arquivos
 - diferentes formas de representar as informações nos arquivos



**registro
tamanho
fixo**



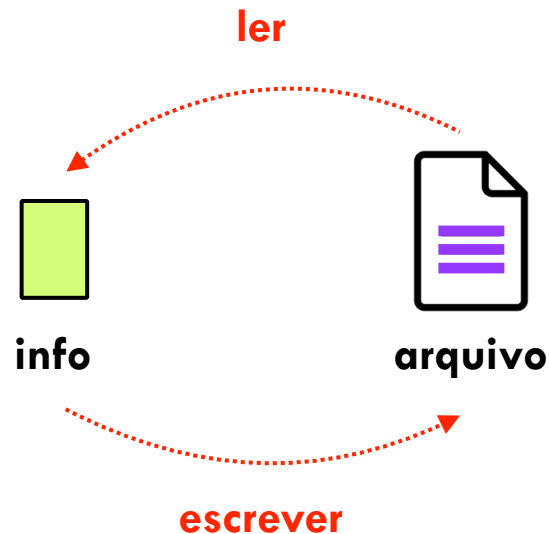
**registro
tamanho
variado**



etc ...

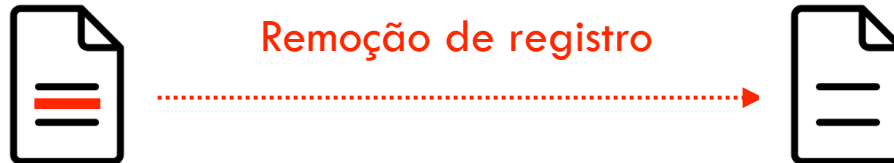
Introdução

- O que já fizemos:
 - como organizar informações em arquivos
 - diferentes formas de representar as informações nos arquivos
 - como ler e escrever informações dos arquivos



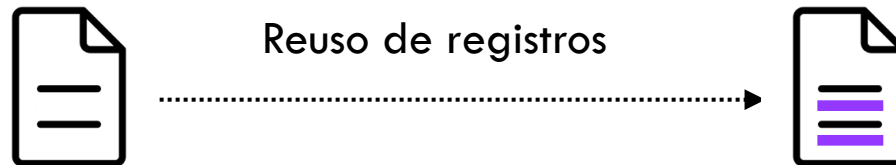
Introdução

- O que já fizemos:
 - como organizar informações em arquivos
 - diferentes formas de representar as informações nos arquivos
 - como ler e escrever informações dos arquivos
 - como remover registros de arquivos



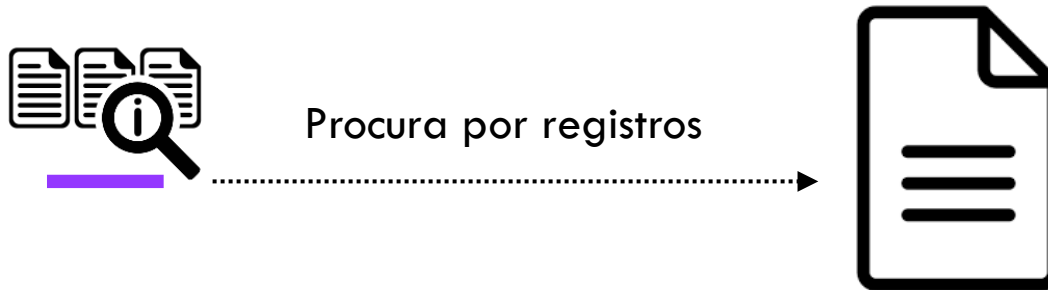
Introdução

- O que já fizemos:
 - como organizar informações em arquivos
 - diferentes formas de representar as informações nos arquivos
 - como ler e escrever informações dos arquivos
 - como remover registros de arquivos
 - como otimizar o uso dos arquivos / reuso



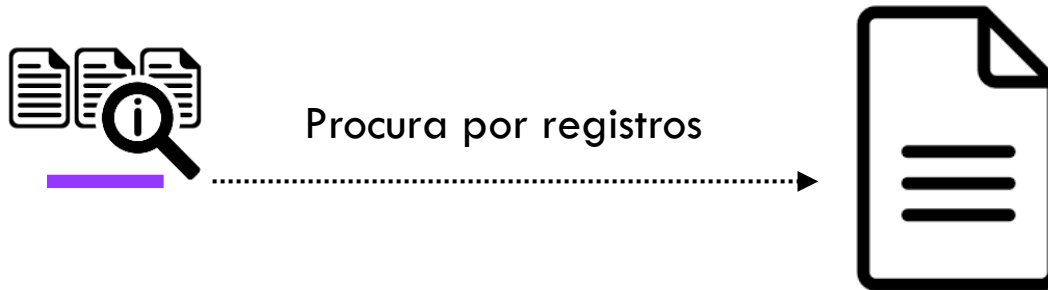
Introdução

- Como procuramos/recuperamos uma informação desejada?



Introdução

- Como procuramos/recuperamos uma informação desejada?



- eficiência depende do número de leituras (seek)
- o que vale para procura também vale para ordenação

Introdução



Introdução



Começo de uma quest ...

Introdução

- Até agora:



Procura por registros



Introdução

- Até agora:



Procura por registros



RRN

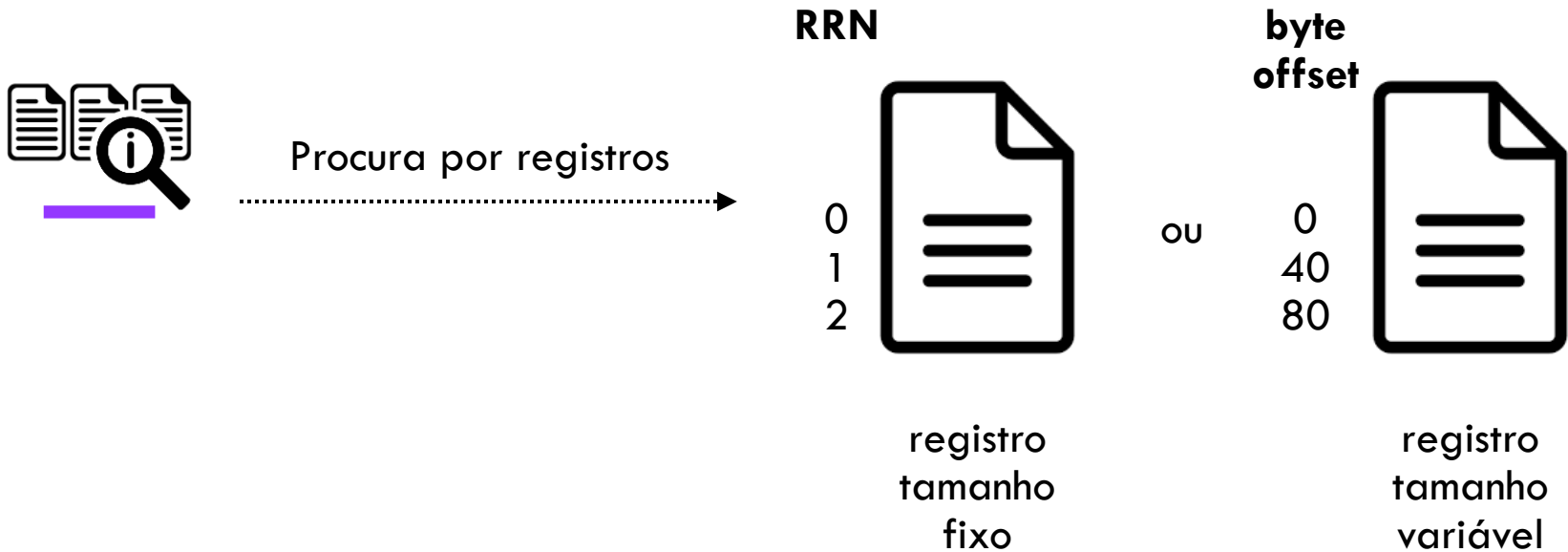
0
1
2



registro
tamanho
fixo

Introdução

- Até agora:



Introdução

- Até agora:



Procura por registros



E se não temos o **RNN** ou **byte offset**?

Introdução

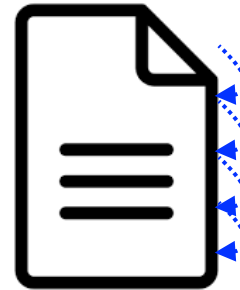
- Até agora:



Procura por registros



E se não temos o RNN ou byte offset?



Busca sequencial

Introdução

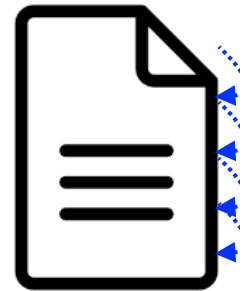
- Até agora:



Procura por registros



E se não temos o RNN ou byte offset?



Busca sequencial



Muito ruim!
Precisamos de uma solução melhor!

Roteiro



- 1 Introdução
- 2 Busca Binária
- 3 Key Sorting
- 4 Revisão
- 5 Referências

Busca binária



Procura por registros



registro
tamanho
fixo

Busca binária

Ames | Mary | 123 Maple | Stillwater | OK | 74075 |
Morrison | Sebastian | 9035 South Hill | Forest Village | OK | 74820 |
Mason | Alan | 90 Eastgate | Ada | OK | 74820 |
Brown | Martha | 625 Kimbark | Des Moines | IA | 50311 |
Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |
DaMassa | Andrezão | 67 Arquitetura dos computa | Apucarana PR | 86800 | ...
Da Salsicha | Tamara | 10 Compila Street | Londrina PR | 85800 |
Vegano | Fernando | 07 Redes N7 | Apucarana | PR | 86800 |
Maumau | Maurício | 524 Rua dos Controles | Apucarana | PR | 86800 |
Naldo | Eri | 12 FPGA | Apucarana | PR | 86800 |

Busca binária

- Para funcionar:

Ames	Mary	123 Maple	Stillwater	OK	74075
Morrison	Sebatian	9035 South Hill	Forest Village	OK	74820
Mason	Alan	90 Eastgate	Ada	OK	74820
Brown	Mar				
Mantovani					
DaMassa	A					86800 ...
Da Salsicha	Tamara	10 Compila Street	Londrina	PR	85800
Vegano	Fernando	07 Redes N7	Apucarana	PR	86800
Maumau	Maurício	524 Rua dos Controles	Apucarana	PR	86800
Naldo	Eri	12 FPGA	Apucarana	PR	86800

- * arquivo com registros de tamanho fixo
- * arquivo ordenado de forma crescente (chave)

Busca binária

Key
(chave canônica)

Ames	Mary	123 Maple	Stillwater	OK	74075
Morrison	Sebatian	9035 South Hill	Forest Village	OK	74820
Mason	Alan	90 Eastgate	Ada	OK	74820
Brown	Martha	625 Kimbark	Des Moines	IA	50311
Mantovani	Rafael	14 Rua Cobol	Rolandópolis	PR	86605
DaMassa	Andrezão	67 Arquitetura dos computa	Apucarana	PR	86800	...
Da Salsicha	Tamara	10 Compila Street	Londrina	PR	85800
Vegano	Fernando	07 Redes N7	Apucarana	PR	86800
Maumau	Maurício	524 Rua dos Controles	Apucarana	PR	86800
Naldo	Eri	12 FPGA	Apucarana	PR	86800

Busca binária

Key
(chave canônica)

MARY AMES

SEBASTIAN MORISSON

ALAN MASON

MARTHA BROWN

RAFAEL MANTOVANI

ANDREZÃO DAMASSA

TAMARA DA SALSICHA

FERNANDO VEGANO

MAURICIO MAUMAU

ERI NALDO

Ames Mary 123 Maple Stillwater OK 74075
Morrison Sebastian 9035 South Hill Forest Village OK 74820
Mason Alan 90 Eastgate Ada OK 74820
Brown Martha 625 Kimbark Des Moines IA 50311
Mantovani Rafael 14 Rua Cobol Rolandópolis PR 86605
DaMassa Andreção 67 Arquitetura dos computa Apucarana PR 86800 ...
Da Salsicha Tamara 10 Compila Street Londrina PR 85800
Vegano Fernando 07 Redes N7 Apucarana PR 86800
Maumau Maurício 524 Rua dos Controles Apucarana PR 86800
Naldo Eri 12 FPGA Apucarana PR 86800

Busca binária

Arquivo Ordenado em Ordem Crescente

ANDREZÃO DAMASSA
ALAN MASON
ERI NALDO
MARY AMES
FERNANDO VEGANO
MARTHA BROWN
MAURICIO MAUMAU
RAFAEL MANTOVANI
SEBASTIAN MORISSON
TAMARA DA SALSICHA

DaMassa | Andrezão | 67 Arquitetura dos computa | Apucarana PR | 86800 | ...
Mason | Alan | 90 Eastgate | Ada | OK | 74820 |
Naldo | Eri | 12 FPGA | Apucarana | PR | 86800 |
Ames | Mary | 123 Maple | Stillwater | OK | 74075 |
Vegano | Fernando | 07 Redes N7 | Apucarana | PR | 86800 |
Brown | Martha | 625 Kimbark | Des Moines | IA | 50311 |
Maumau | Maurício | 524 Rua dos Controles | Apucarana | PR | 86800 |
Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |
Morrison | Sebastian | 9035 South Hill | Forest Village | OK | 74820 |
Da Salsicha | Tamara | 10 Compila Street | Londrina PR | 85800 |

Busca binária

DaMassa		Andrezão		67	Arquitetura dos computa		Apucarana PR		86800		...
Mason		Alan		90	Eastgate		Ada		OK		74820
Naldo		Eri		12	FPGA		Apucarana		PR		86800
Ames		Mary		123	Maple		Stillwater		OK		74075
Vegano		Fernando		07	Redes N7		Apucarana		PR		86800
Brown		Martha		625	Kimbarck		Des Moines		IA		50311
Maumau		Maurício		524	Rua dos Controles		Apucarana		PR		86800
Mantovani		Rafael		14	Rua Cobol		Rolandópolis		PR		86605
Morrison		Sebatian		9035	South Hill		Forest Village		OK		74820
Da Salsicha		Tamara		10	Compila Street		Londrina PR		85800	

Busca binária

Procurar:

Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |

DaMassa | Andrezão | 67 Arquitetura dos computa | Apucarana PR | 86800 | ...
Mason | Alan | 90 Eastgate | Ada | OK | 74820 |
Naldo | Eri | 12 FPGA | Apucarana | PR | 86800 |
Ames | Mary | 123 Maple | Stillwater | OK | 74075 |
Vegano | Fernando | 07 Redes N7 | Apucarana | PR | 86800 |
Brown | Martha | 625 Kimbark | Des Moines | IA | 50311 |
Maumau | Maurício | 524 Rua dos Controles | Apucarana | PR | 86800 |
Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |
Morrison | Sebastian | 9035 South Hill | Forest Village | OK | 74820 |
Da Salsicha | Tamara | 10 Compila Street | Londrina PR | 85800 |

Busca binária

Procurar:

Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |

RRNs

0	DaMassa Andrezão 67 Arquitetura dos computa Apucarana PR 86800 ...
1	Mason Alan 90 Eastgate Ada OK 74820
2	Naldo Eri 12 FPGA Apucarana PR 86800
3	Ames Mary 123 Maple Stillwater OK 74075
4	Vegano Fernando 07 Redes N7 Apucarana PR 86800
5	Brown Martha 625 Kimbark Des Moines IA 50311
6	Maumau Maurício 524 Rua dos Controles Apucarana PR 86800
7	Mantovani Rafael 14 Rua Cobol Rolandópolis PR 86605
8	Morrison Sebastian 9035 South Hill Forest Village OK 74820
9	Da Salsicha Tamara 10 Compila Street Londrina PR 85800

Busca binária

Procurar:

Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |

RRNs

Início →

0

DaMassa | Andrezão | 67 Arquitetura dos computa | Apucarana PR | 86800 | ...

1

Mason | Alan | 90 Eastgate | Ada | OK | 74820 |

2

Naldo | Eri | 12 FPGA | Apucarana | PR | 86800 |

3

Ames | Mary | 123 Maple | Stillwater | OK | 74075 |

4

Vegano | Fernando | 07 Redes N7 | Apucarana | PR | 86800 |

5

Brown | Martha | 625 Kimbark | Des Moines | IA | 50311 |

6

Maumau | Maurício | 524 Rua dos Controles | Apucarana | PR | 86800 |

7

Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |

8

Morrison | Sebastian | 9035 South Hill | Forest Village | OK | 74820 |

Fim →

9

Da Salsicha | Tamara | 10 Compila Street | Londrina PR | 85800 |

$$\text{Meio} = (\text{Início} + \text{Fim}) / 2 = (0 + 10) / 2 = 5$$

Busca binária

Procurar:

Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |

RRNs

Início →

0

DaMassa | Andrezão | 67 Arquitetura dos computa | Apucarana PR | 86800 | ...

1

Mason | Alan | 90 Eastgate | Ada | OK | 74820 |

2

Naldo | Eri | 12 FPGA | Apucarana | PR | 86800 |

3

Ames | Mary | 123 Maple | Stillwater | OK | 74075 |

Meio



4

Vegano | Fernando | 07 Redes N7 | Apucarana | PR | 86800 |

5

Brown | Martha | 625 Kimbark | Des Moines | IA | 50311 |

6

Maumau | Maurício | 524 Rua dos Controles | Apucarana | PR | 86800 |

7

Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |

8

Morrison | Sebastian | 9035 South Hill | Forest Village | OK | 74820 |

Fim →

9

Da Salsicha | Tamara | 10 Compila Street | Londrina PR | 85800 |

$$\text{Meio} = (\text{Início} + \text{Fim}) / 2 = (0 + 9) / 2 = 4.5 = 4$$

Busca binária

Procurar:

Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |

RRNs

Início →

Fim →

0	DaMassa Andrezão 67 Arquitetura dos computa Apucarana PR 86800 ...
1	Mason Alan 90 Eastgate Ada OK 74820
2	Naldo Eri 12 FPGA Apucarana PR 86800
3	Ames Mary 123 Maple Stillwater OK 74075
4	Vegano Fernando 07 Redes N7 Apucarana PR 86800
5	Brown Martha 625 Kimbark Des Moines IA 50311
6	Maumau Maurício 524 Rua dos Controles Apucarana PR 86800
7	Mantovani Rafael 14 Rua Cobol Rolandópolis PR 86605
8	Morrison Sebastian 9035 South Hill Forest Village OK 74820
9	Da Salsicha Tamara 10 Compila Street Londrina PR 85800

Busca binária

Procurar:

Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |

RRNs

	0	DaMassa Andrezão 67 Arquitetura dos computa Apucarana PR 86800 ...
	1	Mason Alan 90 Eastgate Ada OK 74820
	2	Naldo Eri 12 FPGA Apucarana PR 86800
	3	Ames Mary 123 Maple Stillwater OK 74075
	4	Vegano Fernando 07 Redes N7 Apucarana PR 86800
Início →	5	Brown Martha 625 Kimbark Des Moines IA 50311
	6	Maumau Maurício 524 Rua dos Controles Apucarana PR 86800
	7	Mantovani Rafael 14 Rua Cobol Rolandópolis PR 86605
	8	Morrison Sebastian 9035 South Hill Forest Village OK 74820
Fim →	9	Da Salsicha Tamara 10 Compila Street Londrina PR 85800

$$\text{Meio} = (\text{Início} + \text{Fim}) / 2 = (5 + 9) / 2 = 7$$

Busca binária

Procurar:

Mantovani | Rafael | 14 Rua Cobol | Rolandópolis | PR | 86605 |

RRNs

	0	DaMassa Andrezão 67 Arquitetura dos computa Apucarana PR 86800 ...
	1	Mason Alan 90 Eastgate Ada OK 74820
	2	Naldo Eri 12 FPGA Apucarana PR 86800
	3	Ames Mary 123 Maple Stillwater OK 74075
	4	Vegano Fernando 07 Redes N7 Apucarana PR 86800
Início →	5	Brown Martha 625 Kimbark Des Moines IA 50311
	6	Maumau Maurício 524 Rua dos Controles Apucarana PR 86800
Meio →	7	Mantovani Rafael 14 Rua Cobol Rolandópolis PR 86605
	8	Morrison Sebastian 9035 South Hill Forest Village OK 74820
Fim →	9	Da Salsicha Tamara 10 Compila Street Londrina PR 85800

Achamos :))

Busca binária

/ Função para busca binária em arquivos*

Parametros:

- **arq**: arquivo com os registros já ordenados
- **chave**: a chave do registro que procuramos
- **reg**: objeto para retornar o registro encontrado

Retorno (booleano):

- **true**: se encontrar o registro e salvar o conteúdo corretamente em reg
- **false**: se não encontrar o registro, e der algum erro durante o processo de leitura

bool buscaBinaria (FILE *arq, **char*** chave, **Registro*** reg);

Busca binária

```
1. bool buscaBinaria (FILE *arq, char* chave, Registro* reg) {  
2.     inicio = 0;  
3.     fim = numeroRegistros(arq) - 1;  
4.     enquanto (inicio <= fim) {  
5.         meio = (inicio + fim)/2  
6.         reg = lerRegistroComRRN(arq, meio)  
7.         se reg.chave == chave return true;  
8.         se reg.chave < chave: fim = meio - 1;  
9.         senão: inicio = meio + 1;  
10.    }  
11.    return false;  
12. }
```

Busca binária

- É atrativa:
 - mas só funciona se os **registros** estiveram **ordenados**
 - ordenar os registros com base em uma **chave**
 - **ordenação é custosa**, mas também um processo muito importante no processamento de arquivos

Ordenação Interna

- **Solução:** ordenar os registros



Ordenação Interna

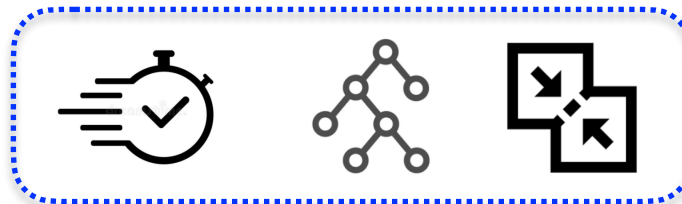
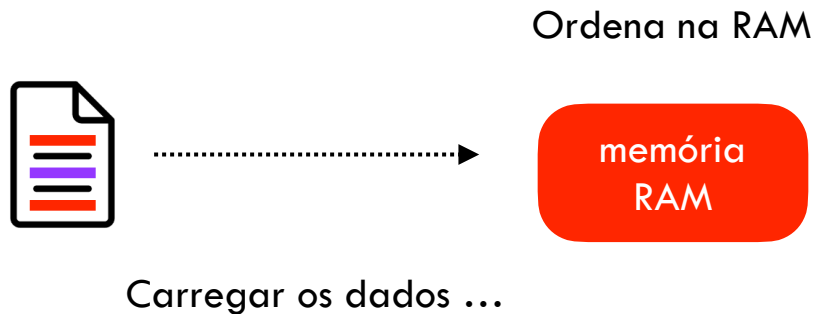
- **Solução:** ordenar os registros



Carregar os dados ...

Ordenação Interna

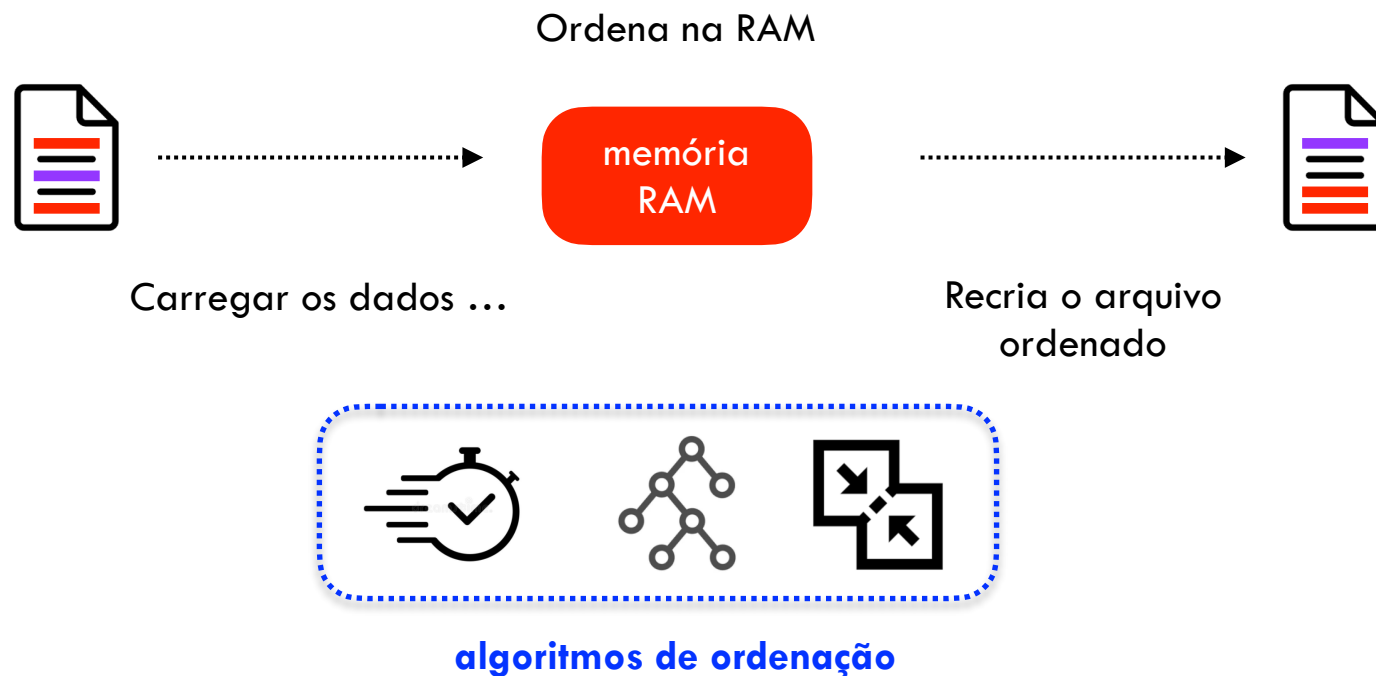
- **Solução:** ordenar os registros



algoritmos de ordenação

Ordenação Interna

- **Solução:** ordenar os registros



Ordenação Interna



- Limitações:

Ordenação Interna

□ Limitações:

- busca binária ainda realiza **muitas** operações de **leitura**
- **manter** o arquivo sempre **ordenado** é **custoso**
- **ordenação interna** só funciona com **arquivos pequenos**

Exercícios

1) Implemente uma função que realiza a busca binária em um arquivo de registros com tamanho fixo. Use os registros de heróis da aula passada como formato de registro (gravação/record).

/ Parâmetros:*

arq: arquivo de entrada com os registros gravados

key: chave no formato canônico que será procurada

hero: registro retornado, quando encontrado

Returns: retorna true se encontrou

*retorna false se não deletou, pq o registro não foi encontrado */*

bool binarySearchFile (FILE *arq, **char*** key, **Heroi*** hero);

Exercícios

```
// struct para representar um Heroi
typedef struct {
    char primeiroNome[16];    // primeiro nome
    char sobrenome[16];       // último nome do heroi
    char nomeHeroi[16];       // nome de heroi, alias
    char poder[16];           // poder
    char fraqueza[21];        // fraqueza
    char cidade[21];          // cidade onde vive
    char profissao[21];       // profissão
} Heroi;
```

Roteiro



- 1** Introdução
- 2** Busca Binária
- 3** Key Sorting
- 4** Revisão
- 5** Referências

KeySorting

- KeySorting: ordenação por chaves
 - tentar solucionar o problema da ordenação interna
 - tag sort
 - ideia: quando queremos ordenar registros, ordenamos mesmo apenas as chaves

KeySorting

- KeySorting: ordenação por chaves
 - assumir que temos **registros** de **tamanho fixo** (pode funcionar para tamanhos variados também)
 - registro de cabeçalho no arquivo (**header**) com o **número de registros** do arquivo
 - algoritmo usa um vetor de **pares ordenados** {chave, RRN}
 - esse vetor se chama KEYNODES []

KeySorting

Arquivo com Registros

Mantovani	Rafael	14	Rua Cobol
DaMassa	Andrezão	67	Arquitetur
Da Salsicha	Tamara	10	Compila
Vegano	Fernando	07	Redes N7
Maumau	Maurício	524	Rua dos
Naldo	Eri	12	FPGA	Apucarana.....

KeySorting

vetor KEYNODES []

Arquivo com Registros

Key

RAFAEL MANTOVANI
ANDREZÃO DAMASSA
TAMARA DA SALSICHA
FERNANDO VEGANO
MAURICIO MAUMAU
ERI NALDO

Mantovani | Rafael | 14 Rua Cobol |
DaMassa | Andrezão | 67 Arquitetur.....
Da Salsicha | Tamara | 10 Compila
Vegano | Fernando | 07 Redes N7 |
Maumau | Maurício | 524 Rua dos
Naldo | Eri | 12 FPGA | Apucarana.....

Criar chaves únicas

KeySorting

vetor KEYNODES []

Key

RAFAEL MANTOVANI
ANDREZÃO DAMASSA
TAMARA DA SALSICHA
FERNANDO VEGANO
MAURICIO MAUMAU
ERI NALDO

Arquivo com Registros

0	Mantovani Rafael 14 Rua Cobol
1	DaMassa Andrezão 67 Arquitetur.....
2	Da Salsicha Tamara 10 Compila
3	Vegano Fernando 07 Redes N7
4	Maumau Maurício 524 Rua dos
5	Naldo Eri 12 FPGA Apucarana.....

KeySorting

vetor KEYNODES []

Key	RRN
RAFAEL MANTOVANI	0
ANDREZÃO DAMASSA	1
TAMARA DA SALSICHA	2
FERNANDO VEGANO	3
MAURICIO MAUMAU	4
ERI NALDO	5

Associar com RRN

Arquivo com Registros

0	Mantovani Rafael 14 Rua Cobol
1	DaMassa Andrezão 67 Arquitetur.....
2	Da Salsicha Tamara 10 Compila
3	Vegano Fernando 07 Redes N7
4	Maumau Maurício 524 Rua dos
5	Naldo Eri 12 FPGA Apucarana.....

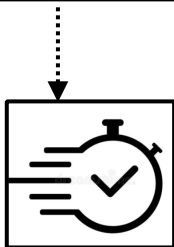
KeySorting

vetor KEYNODES []

Arquivo com Registros

Key	RRN
RAFAEL MANTOVANI	0
ANDREZÃO DAMASSA	1
TAMARA DA SALSICHA	2
FERNANDO VEGANO	3
MAURICIO MAUMAU	4
ERI NALDO	5

0	Mantovani Rafael 14 Rua Cobol
1	DaMassa Andrezão 67 Arquitetur.....
2	Da Salsicha Tamara 10 Compila
3	Vegano Fernando 07 Redes N7
4	Maumau Maurício 524 Rua dos
5	Naldo Eri 12 FPGA Apucarana.....



SORT

Ordenar o arquivo com os
pares de informação
(KEYNODES [])

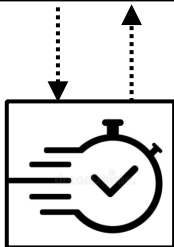
KeySorting

vetor KEYNODES []

Arquivo com Registros

Key	RRN
ANDREZÃO DAMASSA	1
ERI NALDO	5
FERNANDO VEGANO	3
MAURICIO MAUMAU	4
RAFAEL MANTOVANI	0
TAMARA DA SALSICHA	2

0	Mantovani Rafael 14 Rua Cobol
1	DaMassa Andrezão 67 Arquitetur.....
2	Da Salsicha Tamara 10 Compila
3	Vegano Fernando 07 Redes N7
4	Maumau Maurício 524 Rua dos
5	Naldo Eri 12 FPGA Apucarana.....



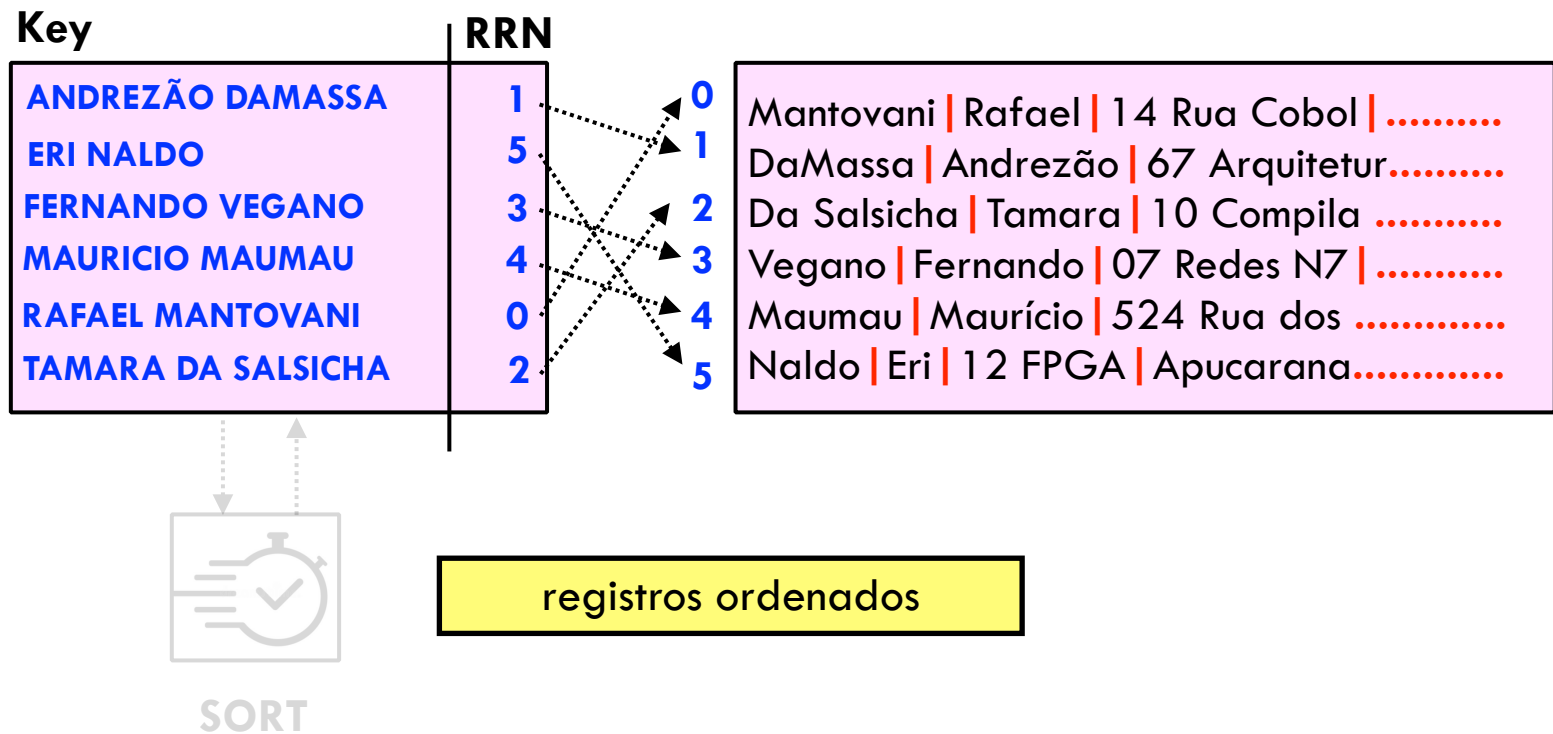
SORT

Pares ordenados
(KEYNODES [])

KeySorting

vetor KEYNODES []

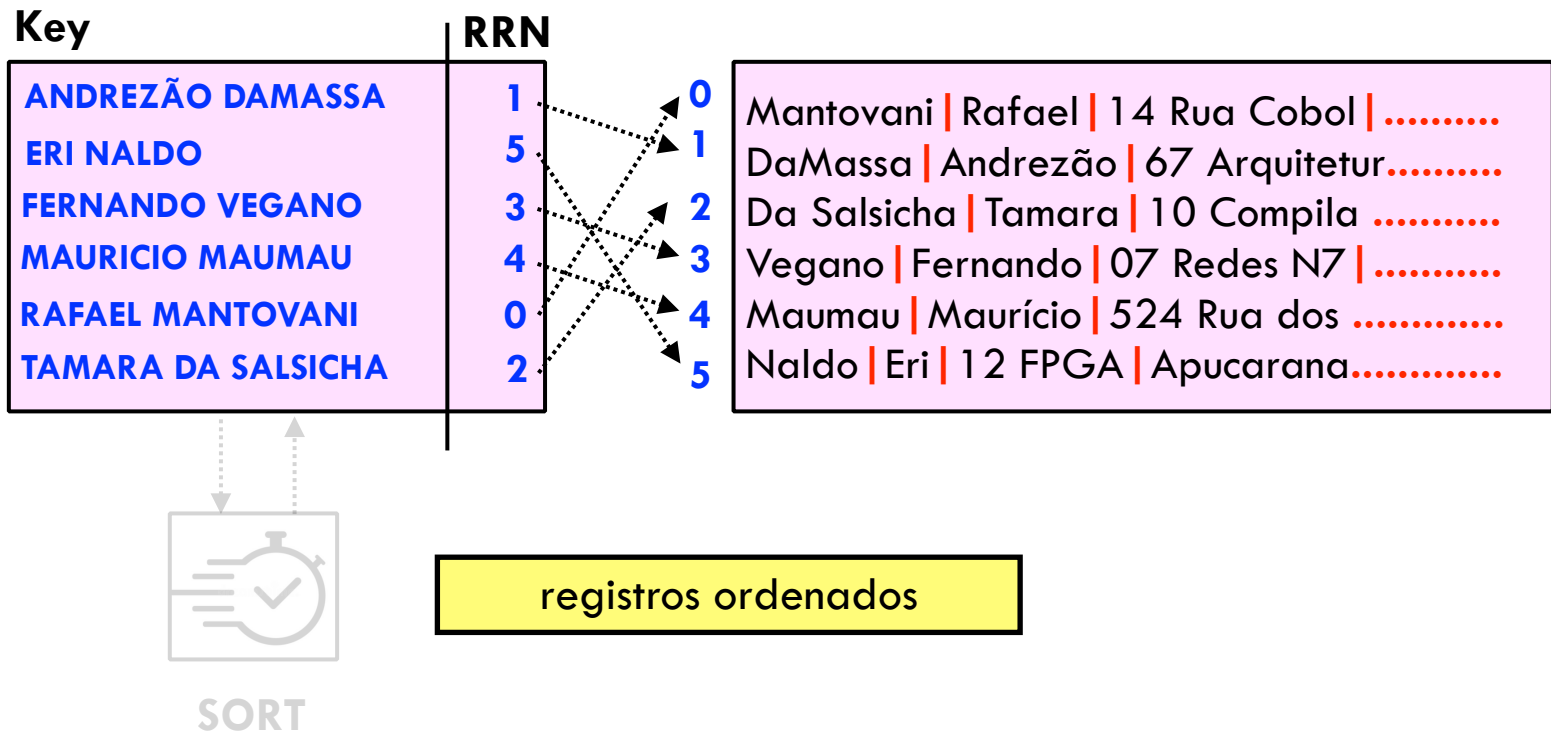
Arquivo com Registros



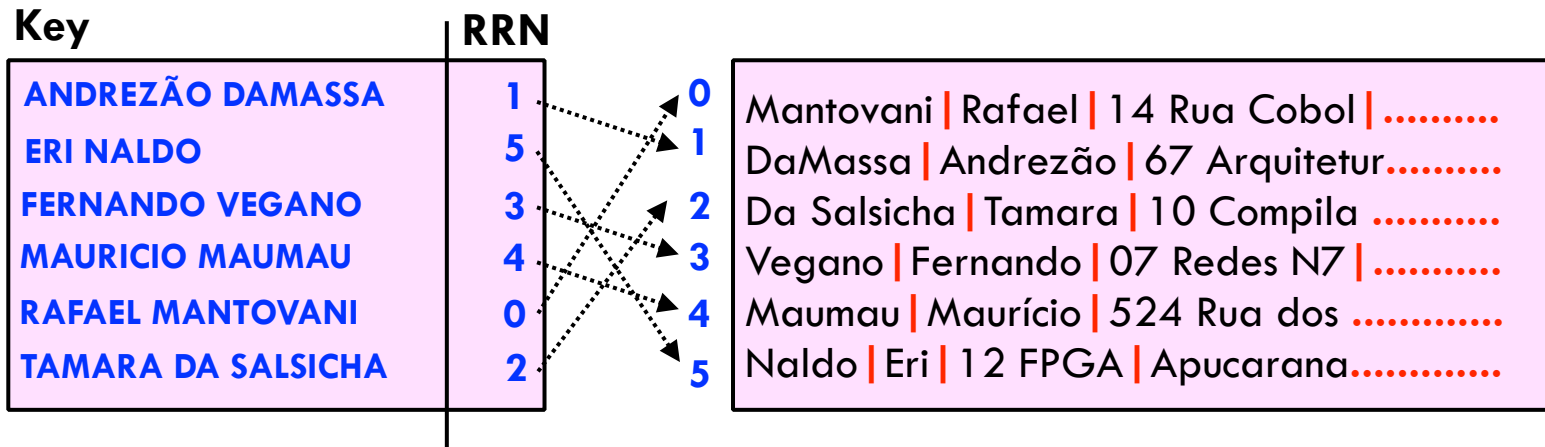
KeySorting

vetor KEYNODES [] (RAM)

Arquivo com Registros (Disco)



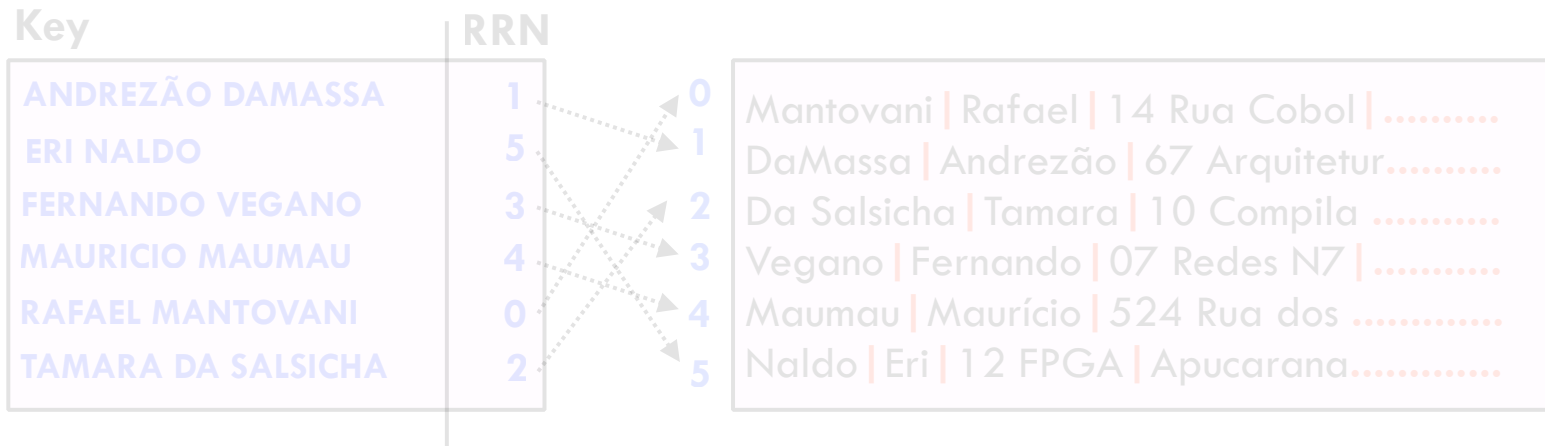
KeySorting



Gerar um novo arquivo
(ordenado)

DaMassa | Andrezão | 67 Arquitetur.....
Naldo | Eri | 12 FPGA | Apucarana.....
Vegano | Fernando | 07 Redes N7 |
Maumau | Maurício | 524 Rua dos
Mantovani | Rafael | 14 Rua Cobol |
Da Salsicha | Tamara | 10 Compila

KeySorting



Deleta os arquivos
anteriores

DaMassa | Andrezão | 67 Arquitetur.....
Naldo | Eri | 12 FPGA | Apucarana.....
Vegano | Fernando | 07 Redes N7 |

Maumau | Maurício | 524 Rua dos
Mantovani | Rafael | 14 Rua Cobol |
Da Salsicha | Tamara | 10 Compila

KeySorting



Procura por registros



DaMassa		Andrezão		67	Arquitetur
Naldo		Eri		12	FPGA	Apucarana.....
Vegano		Fernando		07	Redes N7
Maumau		Maurício		524	Rua dos
Mantovani		Rafael		14	Rua Cobol
Da Salsicha		Tamara		10	Compila

Arquivo ordenado

KeySorting

/ Função para ordenação via KeySorting*

Parametros:

- **arq**: arquivo com os registros a serem lidos e ordenados
- **arqSaida**: nome do novo arquivo que será gerado com os registros ordenados

void keySort (FILE *arq, **char*** arqSaida);

KeySorting

```
1. void keySort (FILE *arq, char* arqSaida) {  
2.     N = numeroRegistros(arq)  
3.     // cada KEYNODE é uma struct com = {chave, RRN}  
4.     criar o vetor KEYNODES [N]  
5.     para cada registro no arq:  
6.         reg = lerRegistroComRRN(arq, i)  
7.         KEYNODES[i] = {reg.chave, i} // i = RRN  
8.     ordenar KEYNODES[ ] com base nas chaves  
9.     criar arquivo de saída arqSaida  
10.    acessar o vetor ordenado KEYNODES[]  
11.    para cada posição de KEYNODES[]  
12.        reg = lerRegistroComRRN(arq, KEYNODES[i]) // RRN já ordenado  
13.        gravar o registro no novo arquivo  
14. }
```

KeySorting



- Limitações:

KeySorting

□ Limitações:

- ordenar é **custoso**
- durante a ordenação, lemos os registros **duas vezes** do arquivo de entrada (nunca desejado!)
- criar o arquivo ordenado requer **muitas operações** de busca, para ter os valores ordenados no arquivo de saída (não desejado)
- basicamente, fazemos tudo em **dobro!**

KeySorting



- Como melhorar?

Roteiro



- 1** Introdução
- 2** Busca Binária
- 3** Key Sorting
- 4** Revisão
- 5** Referências

Revisão



- Busca binária
- Ordenação Interna
- KeySorting

Revisão

- Busca binária
 - requer o arquivo ordenado
 - ainda custosa, mas melhor que a busca sequencial

- Ordenação Interna
 - move os registros para a memória e os ordena
 - só funciona para arquivos pequenos

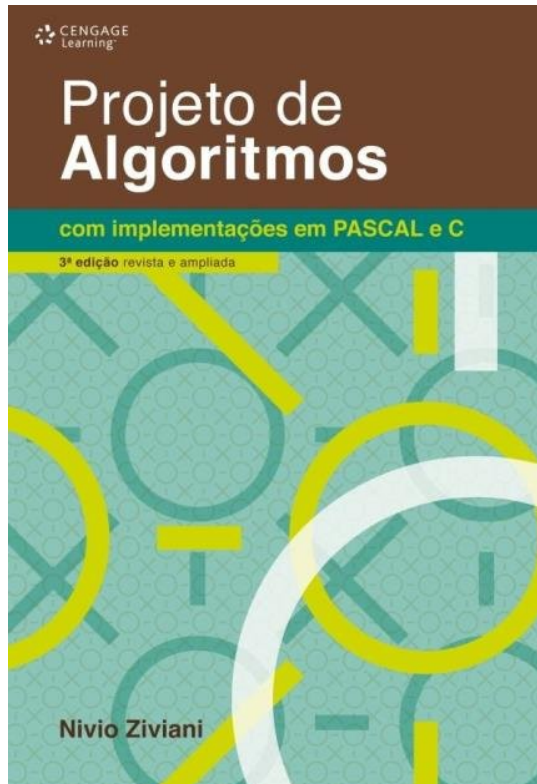
- KeySorting
 - resolve o problema da ordenação
 - mas gasta muitas operações para criar o arquivo já ordenado

Roteiro

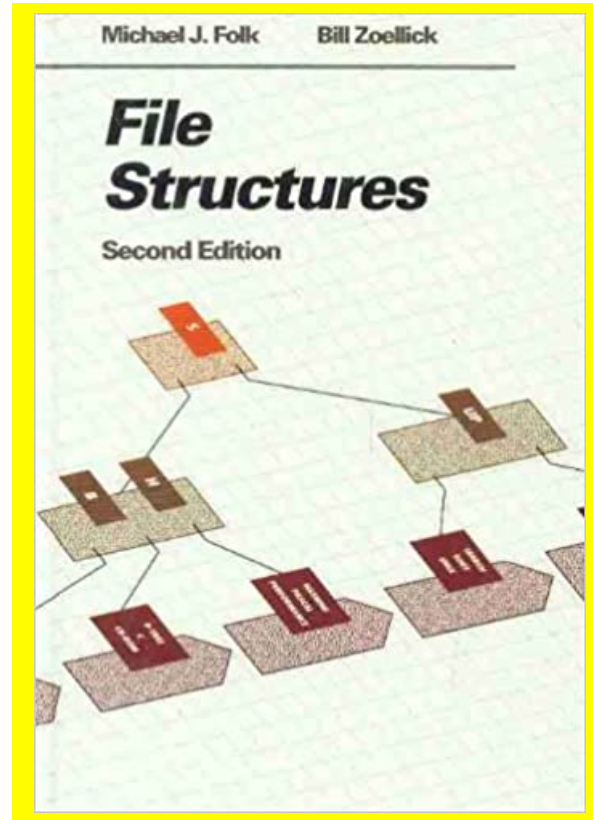


- 1** Introdução
- 2** Busca Binária
- 3** Key Sorting
- 4** Revisão
- 5** Referências

Referências sugeridas

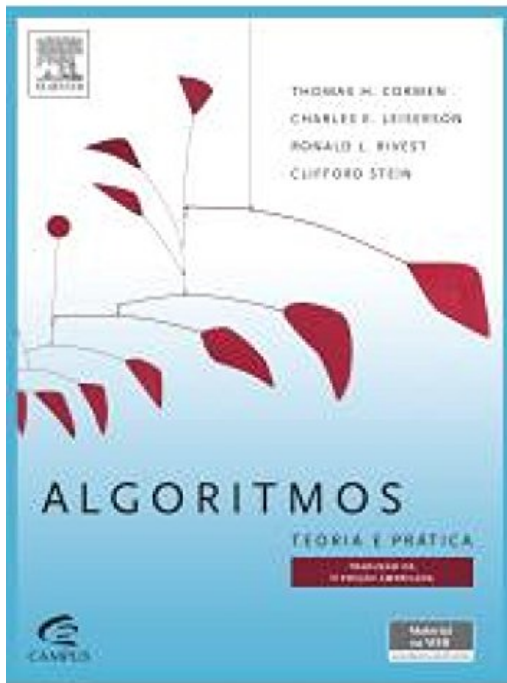


[Ziviani, 2010]



[Folk & Zoellick, 1992]

Referências sugeridas



[Cormen et al, 2018]



[Drozdek, 2017]

Perguntas?

Prof. Rafael G. **Mantovani**

rafaelmantovani@utfpr.edu.br