

ORDENAÇÃO DE ARQUIVOS

profa. Taiane C. Ramos
Estruturas de Dados e Seus
Algoritmos

BUSCA EM ARQUIVO BINÁRIO

Suponha que um banco mantém seus funcionários em um arquivo (mais de 10000 funcionários)

O banco deseja dar um aumento para o funcionário de código 305

Como encontrar o funcionário no arquivo?

ALTERNATIVA 1 – BUSCA SEQUENCIAL

Ler arquivo do início até encontrar o funcionário

? Muito custoso

? Pior caso: funcionário não existe ou é o último, lê o arquivo inteiro

BUSCA SEQUENCIAL POR FUNCIONÁRIO 305




Código	Nome	Salário
102	Joao Silva	1000
123	Carlos Albuquerque	1500
143	Ana Bueno	1500
200	Caio Gusmao	4000
239	Bianca Amarilo	3000
254	Arnaldo Souza	4300
305	Marisa Clara	5000
403	Bruno Simao	4500
410	Guilherme Santos	2000
502	Tatiana Andrade	2500

BUSCA SEQUENCIAL POR FUNCIONÁRIO 305



Código	Nome	Salário
102	Joao Silva	1000
123	Carlos Albuquerque	1500
143	Ana Bueno	1500
200	Caio Gusmao	4000
239	Bianca Amarilo	3000
254	Arnaldo Souza	4300
305	Marisa Clara	5000
403	Bruno Simao	4500
410	Guilherme Santos	2000
502	Tatiana Andrade	2500

BUSCA SEQUENCIAL POR FUNCIONÁRIO 305



Código	Nome	Salário
102	Joao Silva	1000
123	Carlos Albuquerque	1500
143	Ana Bueno	1500
200	Caio Gusmao	4000
239	Bianca Amarilo	3000
254	Arnaldo Souza	4300
305	Marisa Clara	5000
403	Bruno Simao	4500
410	Guilherme Santos	2000
502	Tatiana Andrade	2500

BUSCA SEQUENCIAL POR FUNCIONÁRIO 305

Código	Nome	Salário
102	Joao Silva	1000
123	Carlos Albuquerque	1500
143	Ana Bueno	1500
200	Caio Gusmao	4000
239	Bianca Amarilo	3000
254	Arnaldo Souza	4300
305	Marisa Clara	5000
403	Bruno Simao	4500
410	Guilherme Santos	2000
502	Tatiana Andrade	2500



BUSCA SEQUENCIAL POR FUNCIONÁRIO 305

Código	Nome	Salário
102	Joao Silva	1000
123	Carlos Albuquerque	1500
143	Ana Bueno	1500
200	Caio Gusmao	4000
239	Bianca Amarilo	3000
254	Arnaldo Souza	4300
305	Marisa Clara	5000
403	Bruno Simao	4500
410	Guilherme Santos	2000
502	Tatiana Andrade	2500



BUSCA SEQUENCIAL POR FUNCIONÁRIO 305

Código	Nome	Salário
102	Joao Silva	1000
123	Carlos Albuquerque	1500
143	Ana Bueno	1500
200	Caio Gusmao	4000
239	Bianca Amarilo	3000
254	Arnaldo Souza	4300
305	Marisa Clara	5000
403	Bruno Simao	4500
410	Guilherme Santos	2000
502	Tatiana Andrade	2500



BUSCA SEQUENCIAL POR FUNCIONÁRIO 305

Código	Nome	Salário
102	Joao Silva	1000
123	Carlos Albuquerque	1500
143	Ana Bueno	1500
200	Caio Gusmao	4000
239	Bianca Amarilo	3000
254	Arnaldo Souza	4300
305	Marisa Clara	5000
403	Bruno Simao	4500
410	Guilherme Santos	2000
502	Tatiana Andrade	2500



ALTERNATIVA 2

Se arquivo está ordenado, faz **busca binária**

BUSCA BINÁRIA POR FUNCIONÁRIO

305



	Código	Nome	Salário
1	102	Joao Silva	1000
2	123	Carlos Albuquerque	1500
3	143	Ana Bueno	1500
4	200	Caio Gusmao	4000
5	239	Bianca Amarilo	3000
6	254	Arnaldo Souza	4300
7	305	Marisa Clara	5000
8	403	Bruno Simao	4500
9	410	Guilherme Santos	2000
10	502	Tatiana Andrade	2500

Lê registro do meio e compara
chave buscada com a chave do
registro lido

início = 1
fim = 10
 $\text{meio} = \text{trunc}((\text{início} + \text{fim})/2) = 5$
 $305 > 239$

BUSCA BINÁRIA POR FUNCIONÁRIO

305

	Código	Nome	Salário
1	102	Joao Silva	1000
2	123	Carlos Albuquerque	1500
3	143	Ana Bueno	1500
4	200	Caio Gusmao	4000
5	239	Bianca Amarilo	3000
6	254	Arnaldo Souza	4300
7	305	Marisa Clara	5000
8	403	Bruno Simao	4500
9	410	Guilherme Santos	2000
10	502	Tatiana Andrade	2500

Repete procedimento na metade do arquivo correspondente (se chave menor, na metade de cima, se chave maior, na metade de baixo)

$$\text{início} = \text{meio} + 1$$

BUSCA BINÁRIA POR FUNCIONÁRIO

305

	Código	Nome	Salário
1	102	Joao Silva	1000
2	123	Carlos Albuquerque	1500
3	143	Ana Bueno	1500
4	200	Caio Gusmao	4000
5	239	Bianca Amarilo	3000
6	254	Arnaldo Souza	4300
7	305	Marisa Clara	5000
8	403	Bruno Simao	4500
9	410	Guilherme Santos	2000
10	502	Tatiana Andrade	2500

Lê registro do meio e compara
chave buscada com a chave do
registro lido

início = 6
fim = 10
 $\text{meio} = \text{trunc}((\text{início} + \text{fim})/2) = 8$
 $305 < 403$

BUSCA BINÁRIA POR FUNCIONÁRIO

305

	Código	Nome	Salário
1	102	Joao Silva	1000
2	123	Carlos Albuquerque	1500
3	143	Ana Bueno	1500
4	200	Caio Gusmao	4000
5	239	Bianca Amarilo	3000
6	254	Arnaldo Souza	4300
7	305	Marisa Clara	5000
8	403	Bruno Simao	4500
9	410	Guilherme Santos	2000
10	502	Tatiana Andrade	2500

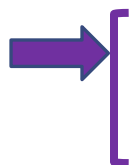
Repete procedimento na metade do arquivo correspondente (se chave menor, na metade de cima, se chave maior, na metade de baixo)

$\text{fim} = \text{meio} - 1$

BUSCA BINÁRIA POR FUNCIONÁRIO

305

	Código	Nome	Salário
1	102	Joao Silva	1000
2	123	Carlos Albuquerque	1500
3	143	Ana Bueno	1500
4	200	Caio Gusmao	4000
5	239	Bianca Amarilo	3000
6	254	Arnaldo Souza	4300
7	305	Marisa Clara	5000
8	403	Bruno Simao	4500
9	410	Guilherme Santos	2000
10	502	Tatiana Andrade	2500



Lê registro do meio e compara
chave buscada com a chave do
registro lido

início = 6
fim = 7
 $\text{meio} = \text{trunc}((\text{início} + \text{fim})/2) = 6$
 $305 > 254$

BUSCA BINÁRIA POR FUNCIONÁRIO

305

	Código	Nome	Salário
1	102	Joao Silva	1000
2	123	Carlos Albuquerque	1500
3	143	Ana Bueno	1500
4	200	Caio Gusmao	4000
5	239	Bianca Amarilo	3000
6	254	Arnaldo Souza	4300
7	305	Marisa Clara	5000
8	403	Bruno Simao	4500
9	410	Guilherme Santos	2000
10	502	Tatiana Andrade	2500

Repete procedimento na metade do arquivo correspondente (se chave menor, na metade de cima, se chave maior, na metade de baixo)

$$\text{inicio} = \text{meio} + 1$$

BUSCA BINÁRIA POR FUNCIONÁRIO

305

	Código	Nome	Salário
1	102	Joao Silva	1000
2	123	Carlos Albuquerque	1500
3	143	Ana Bueno	1500
4	200	Caio Gusmao	4000
5	239	Bianca Amarilo	3000
6	254	Arnaldo Souza	4300
7	305	Marisa Clara	5000
8	403	Bruno Simao	4500
9	410	Guilherme Santes	2000
10	502	Tatiana Andrade	2500



Lê registro do meio e compara
chave buscada com a chave do
registro lido

início = 7
fim = 7
 $\text{meio} = \text{trunc}((\text{início} + \text{fim})/2) = 7$
 $305 = 305$

BUSCA BINÁRIA – DETALHES

? É necessário usar o **seek** para acessar os registros

? Usar cálculo de endereço visto anteriormente

Tem que saber quantos registros tem no arquivo

? Usar função **tamanho_arquivo** vista anteriormente

COMPARAÇÃO

Na busca sequencial, lemos 7 registros

Na busca binária, lemos 4 registros

Assumindo que o arquivo tem n registros:

? Complexidade da busca sequencial: $O(n)$

? Complexidade da busca binária: $O(\log n)$

Código da busca binária

Dado um arquivo de funcionários, ordenado, implementar uma função que faz busca binária no arquivo

```
TFunc *busca_binaria(int cod, FILE *arq, int tam)
```

Abrir o código busca_binaria.c que está disponível no classroom.

ORDENAÇÃO

ORDENAÇÃO

Busca binária exige que arquivo esteja ordenado

Como ordenar um arquivo?

MÉTODOS DE ORDENAÇÃO DE ARQUIVOS

Ordenação em memória

1. Ler arquivo e armazenar os dados num array em memória
2. Ordenar o array
3. Gravar novo arquivo com os dados ordenados

INSERTION SORT

1. Assume que o primeiro valor já está ordenado
2. Pega o próximo valor, compara com os anteriores até descobrir em que posição ele deveria estar
3. Abre espaço no vetor para encaixar o valor na posição correta
4. Encaixa o valor na posição correta
5. Se vetor ainda não terminou, volta para o passo 2

INSERTION SORT



The diagram illustrates the Insertion Sort algorithm. It features a horizontal array of five boxes containing the numbers 3, 4, 5, and 7, followed by the number 2. The number 2 is positioned to the right of the array, indicating it is the current element being inserted. Below the array, the text 'Insertion Sort' is written in a reddish-brown color. The entire diagram is flanked by two vertical black bars: a solid one on the left and a patterned one on the right.

3	4	5	7	2
---	---	---	---	---

Insertion Sort

FUNCIONAMENTO DO INSERTION SORT

1. Assume que o primeiro valor já está ordenado
2. **Pega o próximo valor, compara com os anteriores até descobrir em que posição ele deveria estar**
3. Abre espaço no vetor para encaixar o valor na posição correta
4. Encaixa o valor na posição correta
5. Se vetor ainda não terminou, volta para o passo 2



FUNCIONAMENTO DO INSERTION SORT

1. Assume que o primeiro valor já está ordenado
2. Pega o próximo valor, compara com os anteriores até descobrir em que posição ele deveria estar
3. **Abre espaço no vetor para encaixar o valor na posição correta**
4. Encaixa o valor na posição correta
5. Se vetor ainda não terminou, volta para o passo 2



FUNCIONAMENTO DO INSERTION SORT

1. Assume que o primeiro valor já está ordenado
2. Pega o próximo valor, compara com os anteriores até descobrir em que posição ele deveria estar
3. Abre espaço no vetor para encaixar o valor na posição correta
4. **Encaixa o valor na posição correta**
5. Se vetor ainda não terminou, volta para o passo 2



FUNCIONAMENTO DO INSERTION SORT

1. Assume que o primeiro valor já está ordenado
2. **Pega o próximo valor, compara com os anteriores até descobrir em que posição ele deveria estar**
3. Abre espaço no vetor para encaixar o valor na posição correta
4. Encaixa o valor na posição correta
5. Se vetor ainda não terminou, volta para o passo 2



FUNCIONAMENTO DO INSERTION SORT

1. Assume que o primeiro valor já está ordenado
2. Pega o próximo valor, compara com os anteriores até descobrir em que posição ele deveria estar
3. **Abre espaço no vetor para encaixar o valor na posição correta**
4. Encaixa o valor na posição correta
5. Se vetor ainda não terminou, volta para o passo 2



FUNCIONAMENTO DO INSERTION SORT

1. Assume que o primeiro valor já está ordenado
2. Pega o próximo valor, compara com os anteriores até descobrir em que posição ele deveria estar
3. Abre espaço no vetor para encaixar o valor na posição correta
4. **Encaixa o valor na posição correta**
5. Se vetor ainda não terminou, volta para o passo 2



FUNCIONAMENTO DO INSERTION SORT

1. Assume que o primeiro valor já está ordenado
2. **Pega o próximo valor, compara com os anteriores até descobrir em que posição ele deveria estar**
3. Abre espaço no vetor para encaixar o valor na posição correta
4. Encaixa o valor na posição correta
5. Se vetor ainda não terminou, volta para o passo 2



FUNCIONAMENTO DO INSERTION SORT

1. Assume que o primeiro valor já está ordenado
2. Pega o próximo valor, compara com os anteriores até descobrir em que posição ele deveria estar
3. **Abre espaço no vetor para encaixar o valor na posição correta**
4. Encaixa o valor na posição correta
5. Se vetor ainda não terminou, volta para o passo 2



FUNCIONAMENTO DO INSERTION SORT

1. Assume que o primeiro valor já está ordenado
2. Pega o próximo valor, compara com os anteriores até descobrir em que posição ele deveria estar
3. Abre espaço no vetor para encaixar o valor na posição correta
4. **Encaixa o valor na posição correta**
5. Se vetor ainda não terminou, volta para o passo 2



INSERTION SORT PARA ORDENAR FUNCIONÁRIOS USANDO VETOR EM MEMÓRIA

Abrir código `insertion_sort_memoria.c` disponível no classroom.

COMPLEXIDADE DO INSERTION SORT

Complexidade pior caso $O(n^2)$

Complexidade caso médio $O(n^2)$

Complexidade melhor caso $O(n)$

Complexidade de espaço $O(n)$

OUTROS ALGORITMOS DE ORDENAÇÃO

Selection Sort

Bubble Sort

Quick Sort

...

ORDENAÇÃO DE ARQUIVOS

E se o arquivo não cabe na memória?

? Fazer a ordenação direto no arquivo, **sem usar um vetor auxiliar**

EXERCÍCIO: INSERTION SORT DIRETO NO ARQUIVO

Abra o código `insertion_sort_disco_completar.c` e tente fazer o algoritmo de insertion sort direto no disco com a ajuda do seu grupo.

A solução está disponível no arquivo `insertion_sort_disco_solucão`.

Use as funções que estão prontas no arquivo!

Quem acabar, pode ir pro intervalo!