

# MATA54 - Estruturas de Dados e Algoritmos II

## Ordenação Externa

Flávio Assis

Versão gerada a partir de slides do Prof. George Lima

IC - Instituto de Computação

Salvador, outubro de 2021

# Motivação: memória principal insuficiente

## Aplicações

Ordenar sequências é frequentemente necessário em diversas aplicações. Com grande quantidade de dados, nem sempre pode-se carregar todos os dados em memória principal ao mesmo tempo para ordená-los.

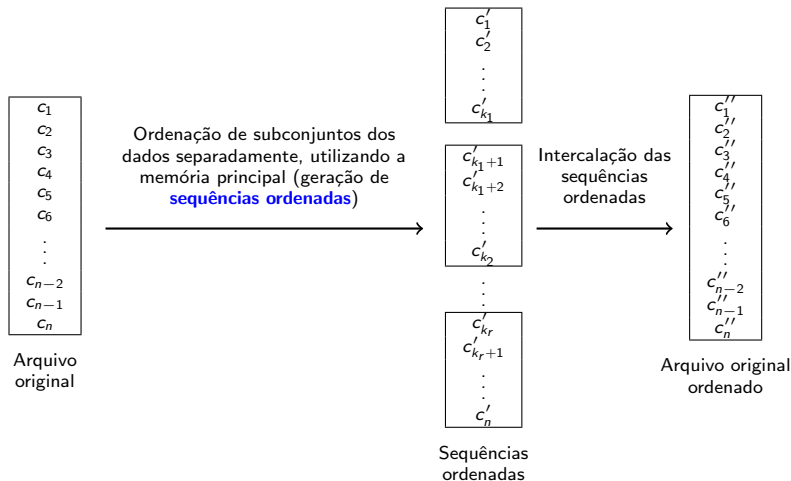
## Ordenação externa

Estratégia: carregar parte dos dados na memória principal, gerando **sequências ordenadas (runs)**. Em seguida, intercalam-se estas sequências até haver apenas uma única sequência ordenada.

## Métodos a serem estudados

- ▶ Intercalação balanceada de  $p$  caminhos
- ▶ Intercalação polifásica
- ▶ Intercalação em cascata

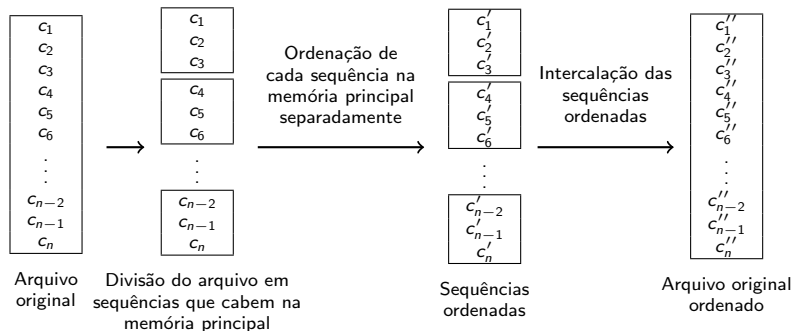
# Ordenação por Intercalação: Idéia Geral



A seqüência  $\langle c''_1, c''_2, \dots, c''_n \rangle$  é uma permutação de  $\langle c_1, c_2, \dots, c_n \rangle$ , tal que  $c''_1 \leq c''_2 \leq \dots \leq c''_n$

## Intercalação Balanceada de $p$ Caminhos

# Ordenação por Intercalação: Idéia Básica

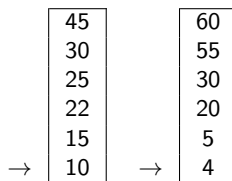


A seqüência  $\langle c_1'', c_2'', \dots, c_n'' \rangle$  é uma permutação de  $\langle c_1, c_2, \dots, c_n \rangle$ , tal que  $c_1'' \leq c_2'' \leq \dots \leq c_n''$

# Intercalação de $p$ -Caminhos

## Ordenação de $p$ sequências ordenadas.

Exemplo para  $p = 2$ :

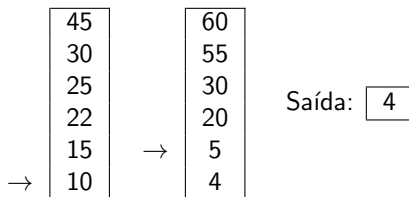


Saída:

# Intercalação de $p$ -Caminhos

## Ordenação de $p$ sequências ordenadas.

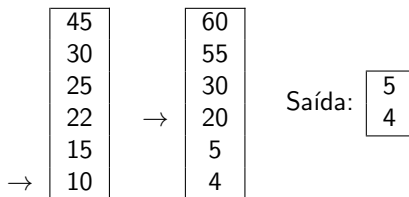
Exemplo para  $p = 2$ :



# Intercalação de $p$ -Caminhos

## Ordenação de $p$ seqüências ordenadas.

Exemplo para  $p = 2$ :

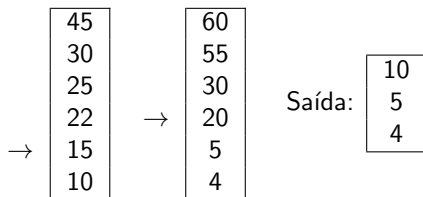




# Intercalação de $p$ -Caminhos

## Ordenação de $p$ seqüências ordenadas.

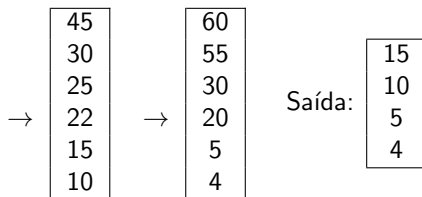
Exemplo para  $p = 2$ :



# Intercalação de $p$ -Caminhos

## Ordenação de $p$ seqüências ordenadas.

Exemplo para  $p = 2$ :

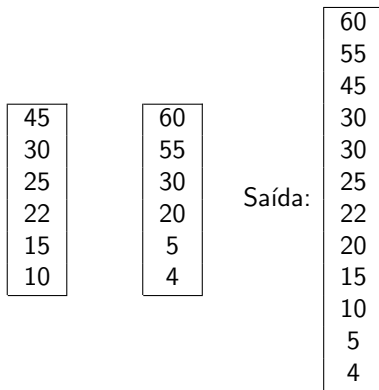


# Intercalação de $p$ -Caminhos

## Ordenação de $p$ seqüências ordenadas.

Exemplo para  $p = 2$ :

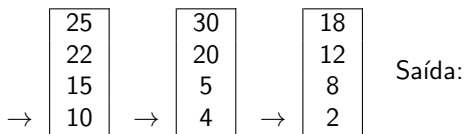
Continuando o processo ...



# Intercalação de $p$ -Caminhos

## Ordenação de $p$ sequências ordenadas.

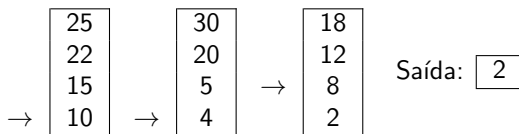
Exemplo para  $p = 3$ :



# Intercalação de $p$ -Caminhos

## Ordenação de $p$ sequências ordenadas.

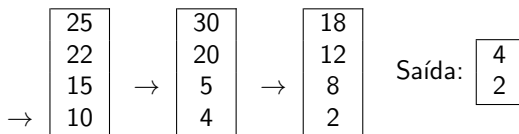
Exemplo para  $p = 3$ :



# Intercalação de $p$ -Caminhos

## Ordenação de $p$ seqüências ordenadas.

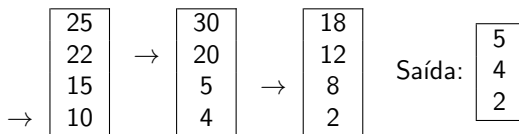
Exemplo para  $p = 3$ :



# Intercalação de $p$ -Caminhos

## Ordenação de $p$ seqüências ordenadas.

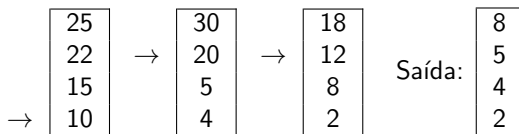
Exemplo para  $p = 3$ :



# Intercalação de $p$ -Caminhos

## Ordenação de $p$ seqüências ordenadas.

Exemplo para  $p = 3$ :

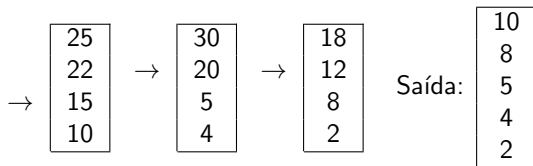




# Intercalação de $p$ -Caminhos

## Ordenação de $p$ sequências ordenadas.

Exemplo para  $p = 3$ :

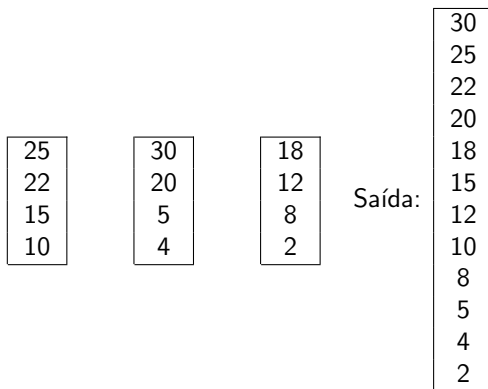


# Intercalação de $p$ -Caminhos

## Ordenação de $p$ sequências ordenadas.

Exemplo para  $p = 3$ :

Continuando o processo ...



# Intercalação Balanceada de $p$ -Caminhos

## Idéia básica:

$2p$  arquivos ( $p$  de entrada e  $p$  de saída); intercalar arquivos de entrada produzindo os de saída. Se memória interna armazena  $m$  registros:

# Intercalação Balanceada de $p$ -Caminhos

## Idéia básica:

$2p$  arquivos ( $p$  de entrada e  $p$  de saída); intercalar arquivos de entrada produzindo os de saída. Se memória interna armazena  $m$  registros:

1. Distribuir os  $n$  registros:  $\lceil \frac{n}{m} \rceil$  blocos ordenados com (até)  $m$  registros em  $p$  arquivos de forma **balanceada (round robin)**.

# Intercalação Balanceada de $p$ -Caminhos

## Idéia básica:

$2p$  arquivos ( $p$  de entrada e  $p$  de saída); intercalar arquivos de entrada produzindo os de saída. Se memória interna armazena  $m$  registros:

1. Distribuir os  $n$  registros:  $\lceil \frac{n}{m} \rceil$  blocos ordenados com (até)  $m$  registros em  $p$  arquivos de forma **balanceada (round robin)**.
2. Enquanto os registros não estiverem em um único arquivo ordenado:
  - 2.1 intercalar os blocos ordenados que estão nos arquivos de entrada ( $p$  blocos por vez) e distribuir os blocos resultantes nos  $p$  arquivos de saída de forma balanceada
  - 2.2 tomar os arquivos de saída como de entrada

# Intercalação Balanceada de $p$ -Caminhos: Exemplo

Exemplo:  $p = 2, m = 3, n = 25$

18, 7, 3, 24, 15, 5, 20, 25, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10

$A_1$	$A_2$	$A_3$	$A_4$
10			
23	12		
17	6		
8	2		
13	22		
4	11		
1	9		
25	21		
20	19		
16	14		
18	24		
7	15		
3	5		

# Intercalação Balanceada de $p$ -Caminhos: Exemplo

Exemplo:  $p = 2, m = 3, n = 25$

18, 7, 3, 24, 15, 5, 20, 25, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10

$A_1$

$A_2$

$A_3$

$A_4$

10
22
13
11
9
4
1
24
18
15
7
5
3

23
17
12
8
6
2
25
21
20
19
16
14

# Intercalação Balanceada de $p$ -Caminhos: Exemplo

Exemplo:  $p = 2, m = 3, n = 25$

18, 7, 3, 24, 15, 5, 20, 25, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10

$A_1$	$A_2$	$A_3$	$A_4$
10			
25	23		
24	22		
21	17		
20	13		
19	12		
18	11		
16	9		
15	8		
14	6		
7	4		
5	2		
3	1		



# Intercalação Balanceada de $p$ -Caminhos: Exemplo

 $A_1$  $A_2$  $A_3$  $A_4$ 

25  
24  
23  
22  
21  
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
9  
8  
7  
6  
5  
4  
3  
2  
1

10

# Intercalação Balanceada de $p$ -Caminhos: Exemplo

 $A_1$  $A_2$  $A_3$  $A_4$ 

25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

# Intercalação Balanceada de $p$ -Caminhos: Exemplo

Exemplo:  $p = 3, m = 3, n = 25$

18, 7, 3, 24, 15, 5, 20, 25, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$
23	12				
17	6				
8	2	10			
21	13	22			
19	4	11			
14	1	9			
18	24	25			
7	15	20			
3	5	16			

# Intercalação Balanceada de $p$ -Caminhos: Exemplo

Exemplo:  $p = 3, m = 3, n = 25$

18, 7, 3, 24, 15, 5, 20, 25, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10

$A_1$

$A_2$

$A_3$

$A_4$

$A_5$

$A_6$

25
24
20
18
16
15
7
5
3

22
21
19
14
13
11
9
4
1

23
17
12
10
8
6
2

# Intercalação Balanceada de $p$ -Caminhos: Exemplo

 $A_1$  $A_2$  $A_3$  $A_4$  $A_5$  $A_6$ 

25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

# Intercalação Balanceada de $p$ -Caminhos

- ▶ Quantas passagens nos  $n$  registros (**rodadas**) são necessárias?

$$\left\lceil \log_p \left( \frac{n}{m} \right) \right\rceil$$

- ▶ Os recursos disponíveis no sistema irão definir o valor para  $p$ . Por exemplo, o número de arquivos que podem ser mantidos abertos simultaneamente.

## Uso de *heap* mínima como fila de prioridades

- ▶ Fila de prioridades de tamanho *m*

## Uso de *heap* mínima como fila de prioridades

- ▶ Fila de prioridades de tamanho  $m$
- ▶ Inserção e remoção da fila tem complexidade  $O(\log m)$



# Seleção por Substituição

## Uso de *heap* mínima como fila de prioridades

- ▶ Fila de prioridades de tamanho  $m$
- ▶ Inserção e remoção da fila tem complexidade  $O(\log m)$
- ▶ Relevante quando  $m$  é grande (durante intercalação)

# Seleção por Substituição

## Uso de *heap* mínima como fila de prioridades

- ▶ Fila de prioridades de tamanho  $m$
- ▶ Inserção e remoção da fila tem complexidade  $O(\log m)$
- ▶ Relevante quando  $m$  é grande (durante intercalação)
- ▶ Relevante para:
  - ▶ formar os blocos iniciais para serem intercalados: **aumento do tamanho das sequências ordenadas**

# Seleção por Substituição

## Uso de *heap* mínima como fila de prioridades

- ▶ Fila de prioridades de tamanho  $m$
- ▶ Inserção e remoção da fila tem complexidade  $O(\log m)$
- ▶ Relevante quando  $m$  é grande (durante intercalação)
- ▶ Relevante para:
  - ▶ formar os blocos iniciais para serem intercalados: **aumento do tamanho das sequências ordenadas**
  - ▶ **intercalação de sequências ordenadas**

Procedimento:

# Seleção por Substituição

## Procedimento:

- ▶  $m$  registros são lidos e armazenados na fila de prioridade

# Seleção por Substituição

## Procedimento:

- ▶  $m$  registros são lidos e armazenados na fila de prioridade
- ▶ O menor é removido da fila e inserido ao final da sequência ordenada, dando lugar no heap ao próximo lido

# Seleção por Substituição

## Procedimento:

- ▶  $m$  registros são lidos e armazenados na fila de prioridade
- ▶ O menor é removido da fila e inserido ao final da sequência ordenada, dando lugar no heap ao próximo lido
- ▶ Se o próximo é menor que o último removido, este próximo valor será marcado. No heap, valores marcados são considerados maiores que valores não marcados. A ordem entre valores não marcados e valores marcados é a ordem natural

## Procedimento:

- ▶  $m$  registros são lidos e armazenados na fila de prioridade
- ▶ O menor é removido da fila e inserido ao final da sequência ordenada, dando lugar no heap ao próximo lido
- ▶ Se o próximo é menor que o último removido, este próximo valor será marcado. No heap, valores marcados são considerados maiores que valores não marcados. A ordem entre valores não marcados e valores marcados é a ordem natural
- ▶ Quando todos os valores no heap estiverem marcados, eles são desmarcados e se dá início a uma nova sequência ordenada



# Uso de Heaps - Exemplo

Exemplo:  $m = 3, n = 25$

**18, 7, 3,** 24, 15, 5, 20, 25, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10

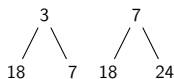


Sequência 1:

# Uso de Heaps - Exemplo

Exemplo:  $m = 3, n = 25$

18, 7, 3, **24**, 15, 5, 20, 25, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10



Sequência 1: **3**

# Uso de Heaps - Exemplo

Exemplo:  $m = 3, n = 25$

18, 7, 3, 24, **15**, 5, 20, 25, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10



Sequência 1: 3, **7**

# Uso de Heaps - Exemplo

Exemplo:  $m = 3, n = 25$

18, 7, 3, 24, 15, **5**, 20, 25, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10

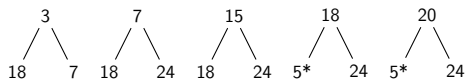


Sequência 1: 3, 7, **15**

# Uso de Heaps - Exemplo

Exemplo:  $m = 3, n = 25$

18, 7, 3, 24, 15, 5, **20**, 25, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10

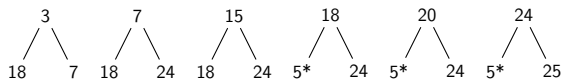


Sequência 1: 3, 7, 15, **18**

# Uso de Heaps - Exemplo

Exemplo:  $m = 3, n = 25$

18, 7, 3, 24, 15, 5, 20, **25**, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10

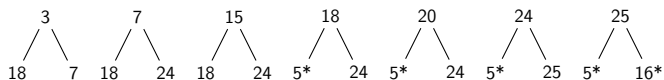


Sequência 1: 3, 7, 15, 18, **20**

# Uso de Heaps - Exemplo

Exemplo:  $m = 3, n = 25$

18, 7, 3, 24, 15, 5, 20, 25, **16**, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10

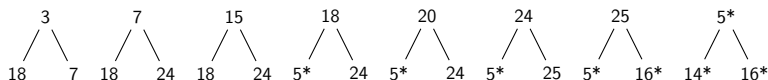


Sequência 1: 3, 7, 15, 18, 20, **24**

# Uso de Heaps - Exemplo

Exemplo:  $m = 3, n = 25$

18, 7, 3, 24, 15, 5, 20, 25, 16, **14**, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10



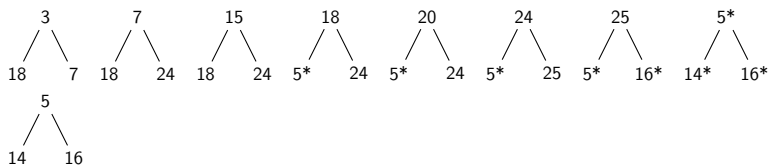
Sequência 1: 3, 7, 15, 18, 20, 24, **25**



# Uso de Heaps - Exemplo

Exemplo:  $m = 3, n = 25$

18, 7, 3, 24, 15, 5, 20, 25, 16, **14**, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10

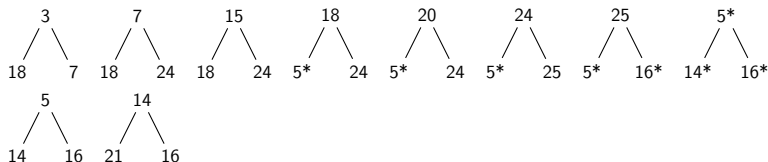


Sequência 1: 3, 7, 15, 18, 20, 24, **25**

# Uso de Heaps - Exemplo

Exemplo:  $m = 3, n = 25$

18, 7, 3, 24, 15, 5, 20, 25, 16, 14, **21**, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10



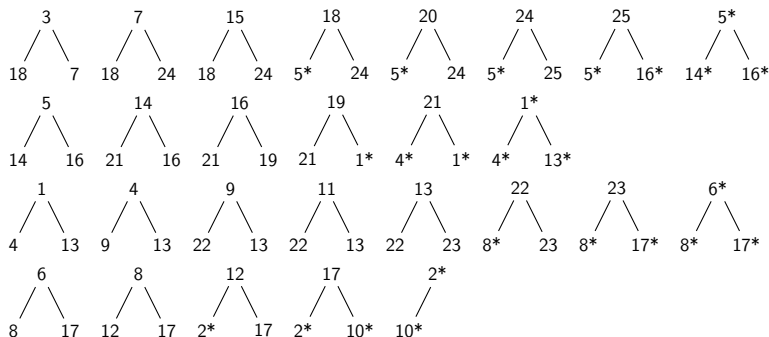
Sequência 1: 3, 7, 15, 18, 20, 24, **25**

Sequência 2: **5**

# Uso de Heaps - Exemplo

Exemplo:  $m = 3, n = 25$

18, 7, 3, 24, 15, 5, 20, 25, 16, 14, 21, 19, 1, 4, 13, 9, 22, 11, 23, 8, 17, 6, 12, 2, 10



Sequência 1: 3, 7, 15, 18, 20, 24, 25

Sequência 2: 5, 14, 16, 19, 21

Sequência 3: 1, 4, 9, 11, 13, 22, 23

Sequência 4: 6, 8, 12, 17

Sequência 5: 2, 10

# Intercalação Balanceada de $p$ Caminhos: Notação

## Notação:

$r_q^p$ :  $q$  seqüências ordenadas de tamanho  $p$

Por exemplo:  $r_{20}^{15}$ : **20** seqüências ordenadas de tamanho **15**

Representação da intercalação balanceada de  $p$  caminhos para o exemplo anterior:

$A_1$	$A_2$	$A_3$		$A_4$	$A_5$	$A_6$
$r_3^3$	$r_3^3$	$r_2^3, r_1^1$		-	-	-
-	-	-		$r_1^9$	$r_1^9$	$r_1^7$
$r_1^{25}$	-	-		-	-	-

# Número de Registros Processados

Qual foi o número de registros processados durante a ordenação?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$
$r_3^3$	$r_3^3$	$r_2^3, r_1^1$	-	-	-
-	-	-	$r_1^9$	$r_1^9$	$r_1^7$
$r_1^{25}$	-	-	-	-	-

Número total:  $25 + 25 = \mathbf{50}$

Total de **2.0** vezes o tamanho do arquivo original.

O arquivo todo é processado a cada rodada do algoritmo.

# Intercalação Polifásica

## Motivação

- ▶ Em princípio, quanto mais sequências ordenadas forem intercaladas a cada vez, mais rapidamente se deve chegar à ordenação de todos os dados.
- ▶ Havendo  $q$  arquivos (no total), por que não intercalar sequências de  $q - 1$  arquivos a cada vez?

## Motivação

- ▶ Em princípio, quanto mais sequências ordenadas forem intercaladas a cada vez, mais rapidamente se deve chegar à ordenação de todos os dados.
- ▶ Havendo  $q$  arquivos (no total), por que não intercalar sequências de  $q - 1$  arquivos a cada vez?
- ▶ Idéia básica do método:
  - ▶ Arquivos de entrada são intercalados até que um deles seja completamente processado - este se torna arquivo de saída



## Motivação

- ▶ Em princípio, quanto mais sequências ordenadas forem intercaladas a cada vez, mais rapidamente se deve chegar à ordenação de todos os dados.
- ▶ Havendo  $q$  arquivos (no total), por que não intercalar sequências de  $q - 1$  arquivos a cada vez?
- ▶ Idéia básica do método:
  - ▶ Arquivos de entrada são intercalados até que um deles seja completamente processado - este se torna arquivo de saída
  - ▶ Repete-se o passo anterior até que a sequência esteja toda em um único arquivo

# Distribuição de Sequências Ordenadas

- ▶ Mas como distribuir as sequências ordenadas nos arquivos?
- ▶ Distribuições aleatórias podem gerar a necessidade de se copiarem sequências em arquivos desnecessariamente

# Distribuição de Sequências Ordenadas

- ▶ Mas como distribuir as sequências ordenadas nos arquivos?
- ▶ Distribuições aleatórias podem gerar a necessidade de se copiarem sequências em arquivos desnecessariamente

## Estratégia

Definir os tamanhos das sequências passo-a-passo, de forma a se eliminar cópias. Quando não houver um número exato de sequências, completa-se o número com **sequências falsas**.

# Distribuição de Sequências Ordenadas

- ▶ Mas como distribuir as sequências ordenadas nos arquivos?
- ▶ Distribuições aleatórias podem gerar a necessidade de se copiarem sequências em arquivos desnecessariamente

## Estratégia

Definir os tamanhos das sequências passo-a-passo, de forma a se eliminar cópias. Quando não houver um número exato de sequências, completa-se o número com **sequências falsas**.

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-------	-------	-------	-------	-------	------------------

# Distribuição de Sequências Ordenadas

- ▶ Mas como distribuir as sequências ordenadas nos arquivos?
- ▶ Distribuições aleatórias podem gerar a necessidade de se copiarem sequências em arquivos desnecessariamente

## Estratégia

Definir os tamanhos das sequências passo-a-passo, de forma a se eliminar cópias. Quando não houver um número exato de sequências, completa-se o número com **sequências falsas**.

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1

# Distribuição de Sequências Ordenadas

- ▶ Mas como distribuir as sequências ordenadas nos arquivos?
- ▶ Distribuições aleatórias podem gerar a necessidade de se copiarem sequências em arquivos desnecessariamente

## Estratégia

Definir os tamanhos das sequências passo-a-passo, de forma a se eliminar cópias. Quando não houver um número exato de sequências, completa-se o número com **sequências falsas**.

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4

# Distribuição de Sequências Ordenadas

- ▶ Mas como distribuir as sequências ordenadas nos arquivos?
- ▶ Distribuições aleatórias podem gerar a necessidade de se copiarem sequências em arquivos desnecessariamente

## Estratégia

Definir os tamanhos das sequências passo-a-passo, de forma a se eliminar cópias. Quando não houver um número exato de sequências, completa-se o número com **sequências falsas**.

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4
2	2	2	-	1	7

# Distribuição de Sequências Ordenadas

- ▶ Mas como distribuir as sequências ordenadas nos arquivos?
- ▶ Distribuições aleatórias podem gerar a necessidade de se copiarem sequências em arquivos desnecessariamente

## Estratégia

Definir os tamanhos das sequências passo-a-passo, de forma a se eliminar cópias. Quando não houver um número exato de sequências, completa-se o número com **sequências falsas**.

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4
2	2	2	-	1	7
-	4	4	2	3	13



# Distribuição de Sequências Ordenadas

- ▶ Mas como distribuir as sequências ordenadas nos arquivos?
- ▶ Distribuições aleatórias podem gerar a necessidade de se copiarem sequências em arquivos desnecessariamente

## Estratégia

Definir os tamanhos das sequências passo-a-passo, de forma a se eliminar cópias. Quando não houver um número exato de sequências, completa-se o número com **sequências falsas**.

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4
2	2	2	-	1	7
-	4	4	2	3	13
4	-	8	6	7	25

# Distribuição de Sequências Ordenadas

- ▶ Mas como distribuir as sequências ordenadas nos arquivos?
- ▶ Distribuições aleatórias podem gerar a necessidade de se copiarem sequências em arquivos desnecessariamente

## Estratégia

Definir os tamanhos das sequências passo-a-passo, de forma a se eliminar cópias. Quando não houver um número exato de sequências, completa-se o número com **sequências falsas**.

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4
2	2	2	-	1	7
-	4	4	2	3	13
4	-	8	6	7	25
12	8	-	14	15	49
...					

# Intercalação Polifásica - Exemplo

Como seriam os passos para ordenação de **49** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

# Intercalação Polifásica - Exemplo

Como seriam os passos para ordenação de **49** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-------	-------	-------	-------	-------	------------------

$r_{12}^1$	$r_8^1$	-	$r_{14}^1$	$r_{15}^1$	49
------------	---------	---	------------	------------	----

# Intercalação Polifásica - Exemplo

Como seriam os passos para ordenação de **49** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
$r_4^1$	-	$r_8^4$	$r_6^1$	$r_7^1$	25
$r_{12}^1$	$r_8^1$	-	$r_{14}^1$	$r_{15}^1$	49

# Intercalação Polifásica - Exemplo

Como seriam os passos para ordenação de **49** sequências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	$r_4^7$	4	$r_2^1$	$r_3^1$	13
$r_4^1$	-	$r_8^4$	$r_6^1$	$r_7^1$	25
$r_{12}^1$	$r_8^1$	-	$r_{14}^1$	$r_{15}^1$	49

# Intercalação Polifásica - Exemplo

Como seriam os passos para ordenação de **49** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
$r_2^{13}$	$r_2^7$	$r_2^4$	-	$r_1^1$	7
-	$r_4^7$	4	$r_2^1$	$r_3^1$	13
$r_4^1$	-	$r_8^4$	$r_6^1$	$r_7^1$	25
$r_{12}^1$	$r_8^1$	-	$r_{14}^1$	$r_{15}^1$	49

# Intercalação Polifásica - Exemplo

Como seriam os passos para ordenação de **49** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
$r_1^{13}$	$r_1^7$	$r_1^4$	$r_1^{25}$	-	4
$r_2^{13}$	$r_2^7$	$r_2^4$	-	$r_1^1$	7
-	$r_4^7$	4	$r_2^1$	$r_3^1$	13
$r_4^1$	-	$r_8^4$	$r_6^1$	$r_7^1$	25
$r_{12}^1$	$r_8^1$	-	$r_{14}^1$	$r_{15}^1$	49



# Intercalação Polifásica - Exemplo

Como seriam os passos para ordenação de **49** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	$r_1^{49}$	1
$r_1^{13}$	$r_1^7$	$r_1^4$	$r_1^{25}$	-	4
$r_2^{13}$	$r_2^7$	$r_2^4$	-	$r_1^1$	7
-	$r_4^7$	4	$r_2^1$	$r_3^1$	13
$r_4^1$	-	$r_8^4$	$r_6^1$	$r_7^1$	25
$r_{12}^1$	$r_8^1$	-	$r_{14}^1$	$r_{15}^1$	49

# Número de Registros Processados

Qual foi o número de registros processados durante a ordenação?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	$r_1^{49}$	1
$r_1^{13}$	$r_1^7$	$r_1^4$	$r_1^{25}$	-	4
$r_2^{13}$	$r_2^7$	$r_2^4$	-	$r_1^1$	7
-	$r_4^7$	4	$r_2^1$	$r_3^1$	13
$r_4^1$	-	$r_8^4$	$r_6^1$	$r_7^1$	25
$r_{12}^1$	$r_8^1$	-	$r_{14}^1$	$r_{15}^1$	49

Número total:  $32 + 28 + 26 + 25 + 49 = \mathbf{160}$

Aproximadamente  $3.27$  vezes o tamanho do arquivo original

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos? Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-------	-------	-------	-------	-------	-------	------------------

---

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-	-	-	-	-	1	1

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-	-	-	-	-	1	1
1	1	1	1	1	-	5

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-	-	-	-	-	1	1
1	1	1	1	1	-	5
-	2	2	2	2	1	9

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-	-	-	-	-	1	1
1	1	1	1	1	-	5
-	2	2	2	2	1	9
2	-	4	4	4	5	19

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-	-	-	-	-	1	1
1	1	1	1	1	-	5
-	2	2	2	2	1	9
2	-	4	4	4	5	19
7	5	9	9	9	-	39



# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-	-	-	-	-	1	1
1	1	1	1	1	-	5
-	2	2	2	2	1	9
2	-	4	4	4	5	19
7	5	9	9	9	-	39
16	14	18	18	-	9	75
...						

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$		# Seqs Ordenadas
-------	-------	-------	-------	-------	-------	--	------------------

---

$r_{16}^1$	$r_{14}^1$	$r_{18}^1$	$r_{18}^1$	-	$r_9^1$		75
							...

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-------	-------	-------	-------	-------	-------	------------------

$r_7^1$	$r_5^1$	$r_9^1$	$r_9^1$	$r_9^5$	-	39
---------	---------	---------	---------	---------	---	----

$r_{16}^1$	$r_{14}^1$	$r_{18}^1$	$r_{18}^1$	-	$r_9^1$	75
------------	------------	------------	------------	---	---------	----

...

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
$r_2^1$	-	$r_4^1$	$r_4^1$	$r_4^5$	$r_5^9$	19
$r_7^1$	$r_5^1$	$r_9^1$	$r_9^1$	$r_9^5$	-	39
$r_{16}^1$	$r_{14}^1$	$r_{18}^1$	$r_{18}^1$	-	$r_9^1$	75
...						

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-	$r_2^{17}$	$r_2^1$	$r_2^1$	$r_2^5$	$r_1^9$	9
$r_2^1$	-	$r_4^1$	$r_4^1$	$r_4^5$	$r_5^9$	19
$r_7^1$	$r_5^1$	$r_9^1$	$r_9^1$	$r_9^5$	-	39
$r_{16}^1$	$r_{14}^1$	$r_{18}^1$	$r_{18}^1$	-	$r_9^1$	75
...						

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
$r_1^{33}$	$r_1^{17}$	$r_1^1$	$r_1^1$	$r_1^5$	-	5
-	$r_2^{17}$	$r_2^1$	$r_2^1$	$r_2^5$	$r_1^9$	9
$r_2^1$	-	$r_4^1$	$r_4^1$	$r_4^5$	$r_5^9$	19
$r_7^1$	$r_5^1$	$r_9^1$	$r_9^1$	$r_9^5$	-	39
$r_{16}^1$	$r_{14}^1$	$r_{18}^1$	$r_{18}^1$	-	$r_9^1$	75
...						

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-	-	-	-	-	$r_1^{75}$	1
$r_1^{33}$	$r_1^{17}$	$r_1^1$	$r_1^1$	$r_1^5$	-	5
-	$r_2^{17}$	$r_2^1$	$r_2^1$	$r_2^5$	$r_1^9$	9
$r_2^1$	-	$r_4^1$	$r_4^1$	$r_4^5$	$r_5^9$	19
$r_7^1$	$r_5^1$	$r_9^1$	$r_9^1$	$r_9^5$	-	39
$r_{16}^1$	$r_{14}^1$	$r_{18}^1$	$r_{18}^1$	-	$r_9^1$	75
...						

# Intercalação Polifásica - Exercício

Como seriam os passos para ordenação de **75** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos? Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-	-	-	-	-	$r_1^{75}$	1
$r_1^{33}$	$r_1^{17}$	$r_1^1$	$r_1^1$	$r_1^5$	-	5
-	$r_2^{17}$	$r_2^1$	$r_2^1$	$r_2^5$	$r_1^9$	9
$r_2^1$	-	$r_4^1$	$r_4^1$	$r_4^5$	$r_5^9$	19
$r_7^1$	$r_5^1$	$r_9^1$	$r_9^1$	$r_9^5$	-	39
$r_{16}^1$	$r_{14}^1$	$r_{18}^1$	$r_{18}^1$	-	$r_9^1$	75
...						

Número total:  $45 + 45 + 34 + 33 + 75 = \mathbf{232}$

Aproximadamente **3.09** vezes o tamanho do arquivo original



# Intercalação em Cascata

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$		# Seqs Ordenadas
-------	-------	-------	-------	-------	--	------------------

---

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4
-	1	2	3	4	10

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4
-	1	2	3	4	10
10	9	7	4	-	30

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Suponhamos que tenhamos cinco arquivos:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4
-	1	2	3	4	10
10	9	7	4	-	30
-	10	19	26	30	85
...					



# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Ajustando as sequências nos arquivos (eliminar cópias):

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$		# Seqs Ordenadas
-------	-------	-------	-------	-------	--	------------------

---

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Ajustando as sequências nos arquivos (eliminar cópias):

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Ajustando as sequências nos arquivos (eliminar cópias):

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Ajustando as sequências nos arquivos (eliminar cópias):

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4
1	-	2	4	3	10

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Ajustando as sequências nos arquivos (eliminar cópias):

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4
1	-	2	4	3	10
10	9	7	-	4	30

# Intercalação em Cascata: Motivação

- ▶ **Estratégia alternativa de intercalação:** passar por todos os registros em cada passo do algoritmo
- ▶ Como distribuir as sequências?

Ajustando as sequências nos arquivos (eliminar cópias):

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	1	1
1	1	1	1	-	4
1	-	2	4	3	10
10	9	7	-	4	30
-	10	19	26	30	85
...					

# Intercalação em Cascata - Exemplo

Como seriam os passos para ordenação de **85** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-------	-------	-------	-------	-------	------------------

-	$r_{10}^1$	$r_{19}^1$	$r_{26}^1$	$r_{30}^1$	85
		...			

# Intercalação em Cascata - Exemplo

Como seriam os passos para ordenação de **85** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
$r_{10}^4$	$r_9^3$	$r_7^2$	-	$r_4^1$	30
-	$r_{10}^1$	$r_{19}^1$	$r_{26}^1$	$r_{30}^1$	85
		...			



# Intercalação em Cascata - Exemplo

Como seriam os passos para ordenação de **85** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
$r_1^4$	-	$r_2^7$	$r_4^{10}$	$r_3^9$	10
$r_{10}^4$	$r_9^3$	$r_7^2$	-	$r_4^1$	30
-	$r_{10}^1$	$r_{19}^1$	$r_{26}^1$	$r_{30}^1$	85
		...			

# Intercalação em Cascata - Exemplo

Como seriam os passos para ordenação de **85** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
$r_1^{26}$	$r_1^{30}$	$r_1^{19}$	$r_1^{10}$	-	4
$r_1^4$	-	$r_2^7$	$r_4^{10}$	$r_3^9$	10
$r_{10}^4$	$r_9^3$	$r_7^2$	-	$r_4^1$	30
-	$r_{10}^1$	$r_{19}^1$	$r_{26}^1$	$r_{30}^1$	85
		...			

# Intercalação em Cascata - Exemplo

Como seriam os passos para ordenação de **85** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	$r_1^{85}$	1
$r_1^{26}$	$r_1^{30}$	$r_1^{19}$	$r_1^{10}$	-	4
$r_1^4$	-	$r_2^7$	$r_4^{10}$	$r_3^9$	10
$r_{10}^4$	$r_9^3$	$r_7^2$	-	$r_4^1$	30
-	$r_{10}^1$	$r_{19}^1$	$r_{26}^1$	$r_{30}^1$	85
		...			

# Taxa de Processamento de Registros

Como seriam os passos para ordenação de **85** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **5** arquivos?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	# Seqs Ordenadas
-	-	-	-	$r_1^{85}$	1
$r_1^{26}$	$r_1^{30}$	$r_1^{19}$	$r_1^{10}$	-	4
$r_1^4$	-	$r_2^7$	$r_4^{10}$	$r_3^9$	10
$r_{10}^4$	$r_9^3$	$r_7^2$	-	$r_4^1$	30
-	$r_{10}^1$	$r_{19}^1$	$r_{26}^1$	$r_{30}^1$	85
...					

Número total:  $81 + 81 + 75 + 85 = 322$

Aproximadamente **3.8** vezes o tamanho do arquivo original

# Intercalação em Cascata - Exercício

Como seriam os passos para ordenação de **190** seqüências ordenadas, originalmente todas de tamanho **1** (por exemplo), usando **6** arquivos?

Qual a taxa de processamento de registros?

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	# Seqs Ordenadas
-	-	-	-	-	$r_1^{190}$	1
$r_1^{50}$	$r_1^{55}$	$r_1^{41}$	$r_1^{29}$	$r_1^{15}$	-	5
$r_1^5$	-	$r_2^9$	$r_3^{12}$	$r_5^{15}$	$r_4^{14}$	15
$r_{15}^5$	$r_{14}^4$	$r_{12}^3$	$r_9^2$	-	$r_5^1$	55
-	$r_{15}^1$	$r_{29}^1$	$r_{41}^1$	$r_{50}^1$	$r_{55}^1$	190
...						

Número total:  $185 + 185 + 175 + 190 = 735$

Aproximadamente **3.9** vezes o tamanho do arquivo original