

ORDENAÇÃO EXTERNA DE ARQUIVOS: GERAÇÃO DE PARTIÇÕES CLASSIFICADAS

Profa. Taiane C. Ramos
Estruturas de Dados e Seus
Algoritmos
Turma de Verão 2023

ORDENAÇÃO DE ARQUIVOS GRANDES

Ao ler um arquivo, enchemos um buffer de memória.

Talvez o arquivo não caiba no buffer.

Se você quiser fazer uma troca entre registros que não cabem no mesmo buffer, você vai ter que fazer mais de um acesso ao disco e vai ficar bastante custoso.

Seria interessante quebrar o problema de forma que a maior parte das trocas coubesse no buffer.

Então vamos tentar ordenar o arquivo em pequenas partes.

ORDENAÇÃO EM DISCO

Vamos querer minimizar a quantidade de operações de entrada e saída.

E que as trocas sejam feitas preferencialmente entre elementos fisicamente próximos no arquivo.

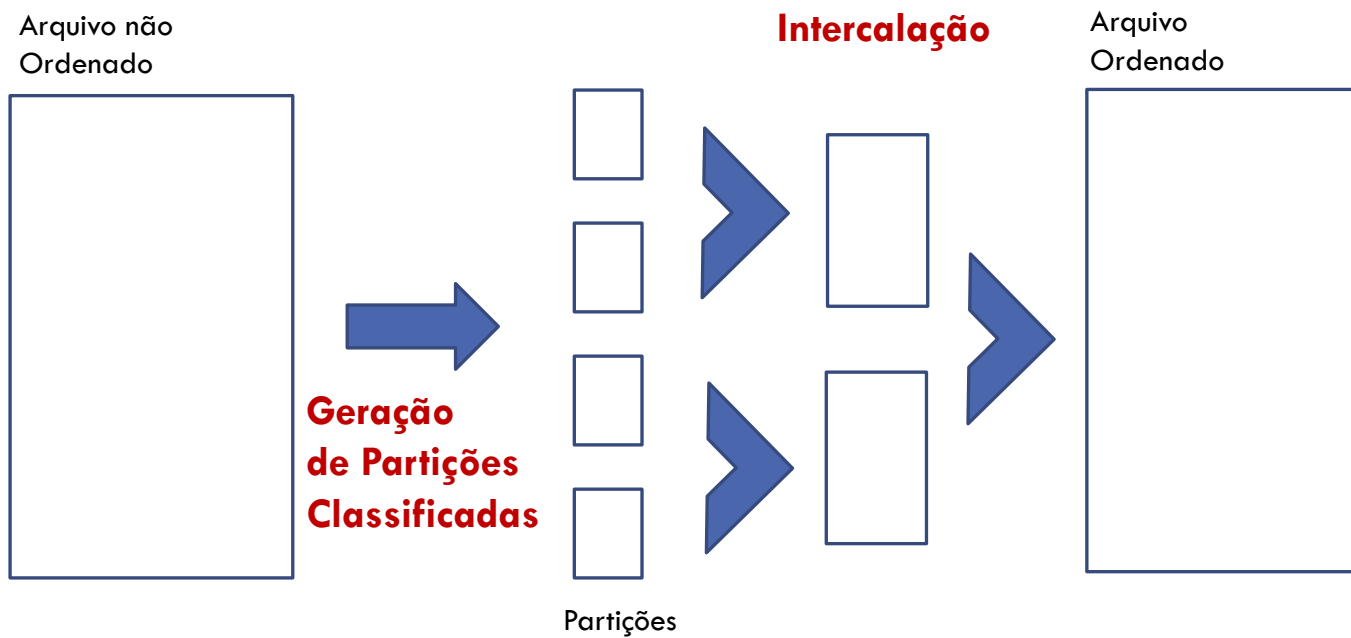
IDEIA BÁSICA DA ORDENAÇÃO EM DISCO

Vamos dividir o problema em problemas menores.

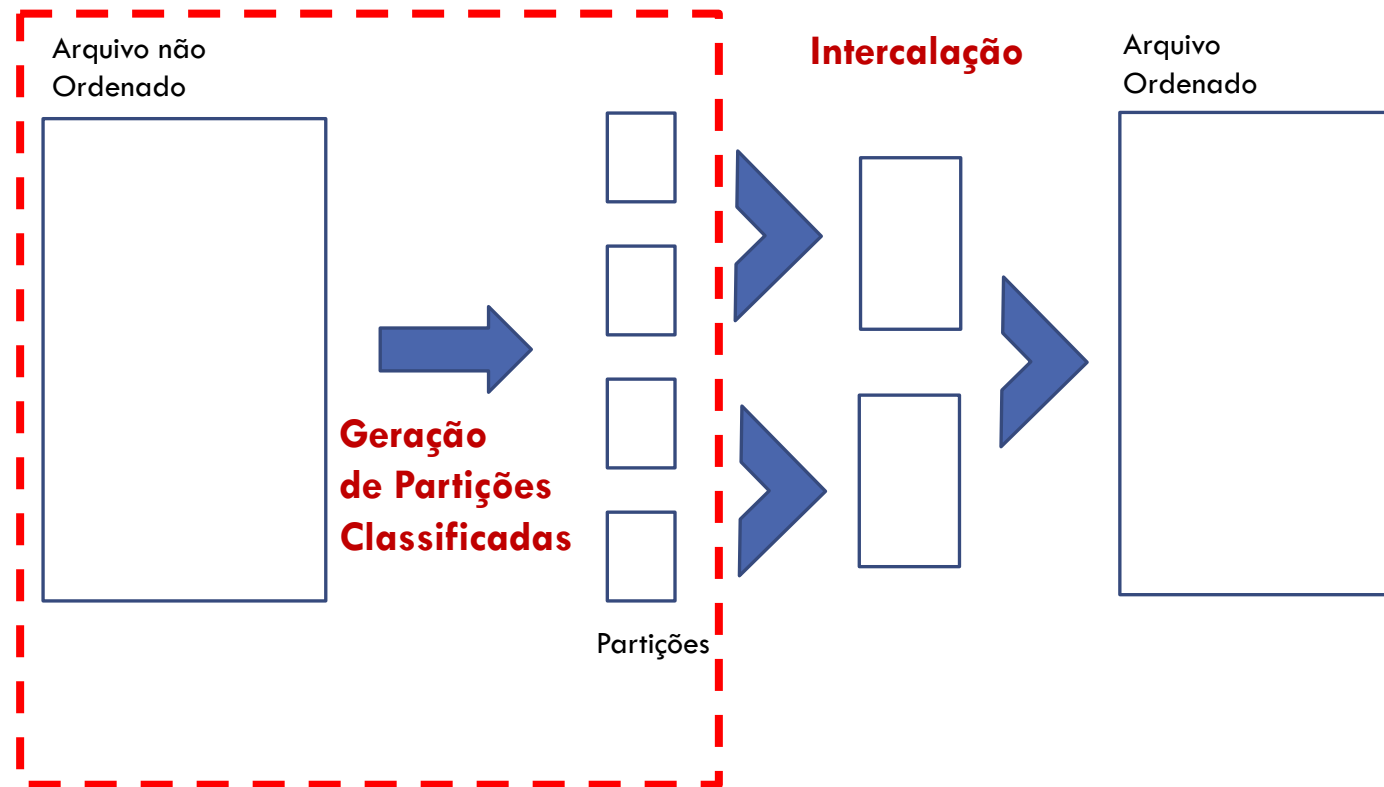
Funciona em duas etapas:

- Ordenação de partições
- Intercalação das partições

MODELO



NESSA AULA: ETAPA DE GERAÇÃO DE PARTIÇÕES ORDENADAS



ETAPA DE GERAÇÃO DE PARTIÇÕES ORDENADAS

Partição: sequência ordenada de n registros.

Registros são lidos do arquivo de entrada (não ordenado)

Estes registros são ordenados e gravados em arquivos de saída de partições ordenadas

GERAÇÃO DE PARTIÇÕES ORDENADAS



MÉTODOS DE GERAÇÃO DE PARTIÇÕES ORDENADAS

Métodos

- Ordenação em principal (Classificação interna)
- Seleção com substituição
- Seleção natural

Memória principal armazena **M** registros

ORDENAÇÃO EM MEMÓRIA
PRINCIPAL |

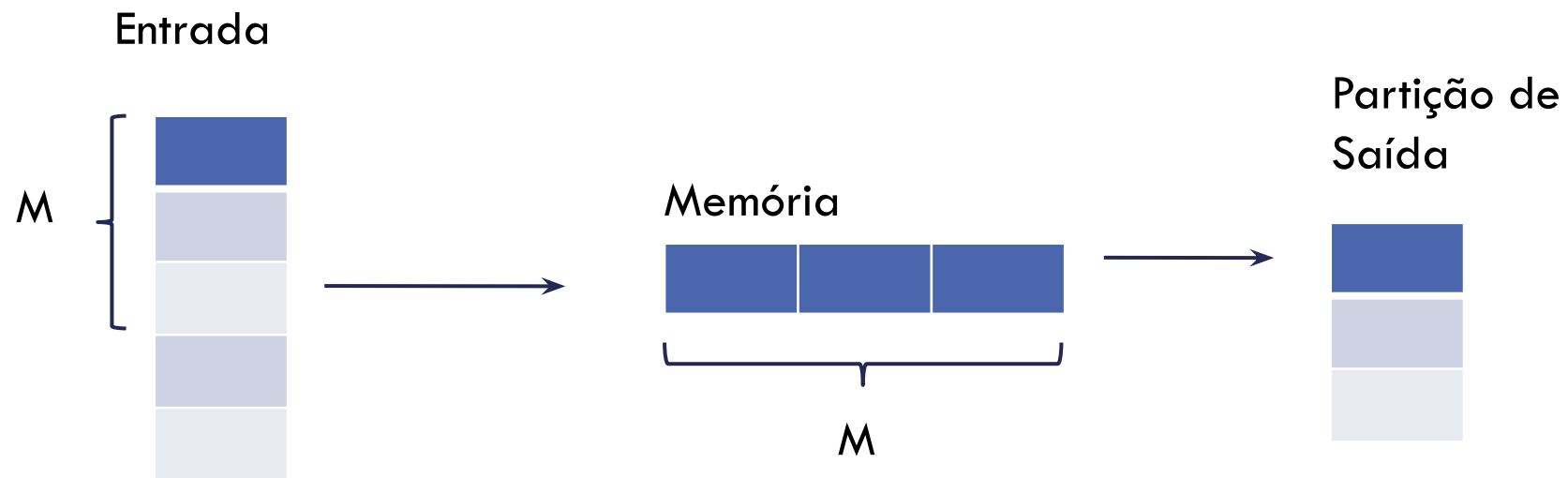
ORDENAÇÃO EM MEMÓRIA PRINCIPAL

Critério eficiência: número de comparações entre chaves

- Lê M registros pra memória
- Ordena esses registros
- Grava em uma partição

Todas as partições contém M registros (exceto a última que tem o que sobrou)

VISÃO GERAL



EXEMPLO

Assumir que na memória cabem 6 registros ($M = 6$)

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

EXEMPLO

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

EXEMPLO

Memória Principal

29	14	76	75	59	6
----	----	----	----	----	---

Leitura



29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

EXEMPLO

Memória Principal

29	14	76	75	59	6
----	----	----	----	----	---



Ordenação

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

EXEMPLO

Memória Principal

6	14	29	59	75	76
---	----	----	----	----	----

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

EXEMPLO

Memória Principal

6	14	29	59	75	76
---	----	----	----	----	----

Partição 1 (em disco) ordenada

6	14	29	69	75	76
---	----	----	----	----	----

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

EXEMPLO

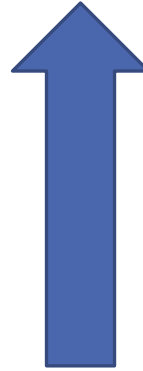
Memória Principal

7	74	48	46	10	18
---	----	----	----	----	----

Partição 1 (em disco) ordenada

6	14	29	69	75	76
---	----	----	----	----	----

Leitura



29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

EXEMPLO

Memória Principal

7	74	48	46	10	18
---	----	----	----	----	----



Ordenação

Partição 1 (em disco) ordenada

6	14	29	69	75	76
---	----	----	----	----	----

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

EXEMPLO

Memória Principal

7	10	18	46	48	74
---	----	----	----	----	----

Partição 1 (em disco) ordenada

6	14	29	69	75	76
---	----	----	----	----	----

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

EXEMPLO

Memória Principal

7	10	18	46	48	74
---	----	----	----	----	----

Partição 1 (em disco) ordenada

6	14	29	69	75	76
---	----	----	----	----	----

Partição 2 (em disco) ordenada

7	10	18	46	48	74
---	----	----	----	----	----

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31							

	Área de trabalho							Partições obtidas					
Memória	29	14	76	75	59	6		6	14	29	59	75	76
Memória	7	74	48	46	10	18		7	10	18	46	48	74
Memória	56	20	26	4	21	65		4	20	21	26	56	65
Memória	22	49	11	16	8	15		8	11	15	16	22	49
Memória	5	19	50	55	25	66		5	19	25	50	55	66
Memória	57	77	12	30	17	9		9	12	17	30	57	77
Memória	54	78	43	38	51	32		32	38	43	51	54	78
Memória	58	13	73	79	27	1		1	13	27	58	73	79
Memória	3	60	36	47	31			3	31	36	47	60	

$P / M = 6$:
 8 partições de tamanho M
 +
 1 partição de tamanho $< M$

SELEÇÃO COM SUBSTITUIÇÃO

SELEÇÃO COM SUBSTITUIÇÃO

Já vai escrevendo o menor na partição (disco) e liberando espaço em memória.

Permite criar partições maiores que M .

EXEMPLO

Assumir que na memória cabem 6 registros ($M = 6$)

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31	80						

	Área de trabalho						Partições obtidas														
Registros	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3ª substituição						24															
2ª substituição	14	18				74															
1ª substituição	46	48	16	26	56	7															
Memória	29	14	76	75	59	6	6	7	14	29	46	48	59	74	75	76					
A 1ª partição ficou com 10 registros																					
2ª substituição	19	16	11			15															
1ª substituição	65	22	21	8	5	49															
Memória	10	18	4	26	56	20	4	10	18	20	21	22	26	49	56	65					
A 2ª partição ficou com 10 registros																					
3ª substituição	43																				
2ª substituição	78	9	14	17	30	74															
1ª substituição	77	57	25	55	50	66															
Memória	19	16	11	8	5	15	5	8	11	15	16	19	25	50	55	57	66	77	78		
A 3ª partição ficou com 13 registros																					
3ª substituição		60																			
2ª substituição	36	73	27	13	3																
1ª substituição	79	38	51	32	58	1															
Memória	43	9	12	17	30	54	9	12	17	30	32	38	43	51	54	58	73	79			
A 4ª partição ficou com 12 registros																					
1ª substituição				80	31	47															
Memória	36	60	27	13	3	1	1	3	13	27	31	36	47	60	80						
A 5ª partição ficou com 9 registros																					
Legenda																					
Registros congelados											Divisão de regiões na tabela										

P / M = 6:
5 partições

SELEÇÃO COM SUBSTITUIÇÃO: ALGORITMO

1. Ler M registros do arquivo para a memória
2. Selecionar, no array em memória, o registro r com menor chave
3. Gravar o registro r na partição de saída
4. Substituir, no array em memória, o registro r pelo próximo registro do arquivo de entrada
5. Caso a chave deste último seja menor do que a chave recém gravada, considerá-lo congelado e ignorá-lo no restante do processamento
6. Caso existam em memória registros não congelados, voltar ao passo 2
7. Caso contrário:
 - fechar a partição de saída
 - descongelar os registros congelados
 - abrir nova partição de saída
 - voltar ao passo 2

TAMANHO DAS PARTIÇÕES GERADAS

Em média, o **tamanho das partições** obtidas pelo processo de seleção com substituição é de **$2 * M$**

SELEÇÃO NATURAL

SELEÇÃO NATURAL

Desvantagem da seleção com substituição: ocupamos a memória principal com os registros congelados.

Podemos jogar os registros congelados para um arquivo “reservatório” em disco.

Reservatório de tamanho máximo M , pois voltaremos esses registros pra memória.

SELEÇÃO NATURAL

A formação de uma partição se encerra quando o reservatório estiver cheio ou quando terminarem os registros de entrada

Se o reservatório tem tamanho M , **comprimento médio das partições** é de $M * e$, onde $e = 2,718... .$

EXEMPLO

Chaves do arquivo a ordenar

□ (Sequência de leitura: 29, 14, 76,...)

Assumir que na memória cabem 6 registros simultaneamente ($M = 6$), e que o tamanho do reservatório também é 6 ($n = 6$)

29	14	76	75	59	6	7	74	48	46	10	18
56	20	26	4	21	65	22	49	11	16	8	15
5	19	50	55	25	66	57	77	12	30	17	9
54	78	43	38	51	32	58	13	73	79	27	1
3	60	36	47	31	80						

	Área de trabalho						Partições obtidas														
Registros	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2ª substituição	56					74															
1ª substituição	46	48				7															
Memória	29	14	76	75	59	6	6	7	14	29	46	48	56	59	74	75	76				
Reservatório	10	18	20	26	4	21															
							A 1ª partição ficou com 11 registros														
1ª substituição	22	49			65																
Memória	10	18	20	26	4	21	4	10	18	20	21	22	26	49	65						
Reservatório	11	16	8	15	5	19															
							A 2ª partição ficou com 9 registros														
3ª substituição	54																				
2ª substituição	30				78																
1ª substituição	25	57	55	66	50	77															
Memória	11	16	8	15	5	19	5	8	11	15	16	19	25	30	50	54	55	57	66	77	78
Reservatório	12	17	9	43	38	51															
							A 3ª partição ficou com 15 registros														
2ª substituição			79																		
1ª substituição	58	73	32	47	60																
Memória	12	17	9	43	38	51	9	12	17	32	38	43	47	51	58	60	73	79			
Reservatório	13	27	1	3	36	31															
							A 4ª partição ficou com 12 registros														
1ª substituição			80																		
Memória	13	27	1	3	36	31	1	3	13	27	31	36	80								
Reservatório							A 5ª partição ficou com 7 registros														

P / M = 6:
5 partições

SELEÇÃO NATURAL: ALGORITMO

1. Ler M registros do arquivo para a memória
2. Selecionar, no array em memória, o registro **r** com menor chave
3. Gravar o registro **r** na partição de saída
4. Substituir, no array em memória, o registro **r** pelo próximo registro do arquivo de entrada
5. Caso a chave deste último seja menor do que a chave recém gravada, gravá-lo no reservatório e substituir, no array em memória, o registro **r** pelo próximo registro do arquivo de entrada
6. Caso ainda exista espaço livre no reservatório, voltar ao passo 2
7. Caso contrário:
 - fechar a partição de saída
 - copiar os registros do reservatório para o array em memória
 - esvaziar o reservatório
 - abrir nova partição de saída
 - voltar ao passo 2

COMPARAÇÃO DOS PROCESSOS

A ordenação em memória (classificação interna) gera as menores partições, o que implica em mais arquivos a intercalar

Os processos de seleção geram partições maiores, reduzindo o tempo total de processamento

A seleção natural sofre o ônus adicional de utilizar mais operações de entrada e saída (devido ao reservatório estar em memória secundária), mas as partições maiores podem compensar esse custo.

EXERCÍCIO 1

Gerar partições classificadas segundo o método de **Seleção com Substituição** para a seguinte situação

Assumir que na memória cabem 7 registros simultaneamente

Arquivo a ordenar

30	14	15	75	32	6	5	81	48	41	87	18
56	20	26	4	21	65	22	49	11	16	8	12

REFERÊNCIA

Ferraz, I. N. Programação com Arquivos. Editora Manole Ltda. Barueri, 2003.