

RESPONDERAM A ENQUETE?

Pessoal, preciso falar o mínimo hoje.

Peço a ajuda de vocês para falarem por mim quando puderem, por exemplo, respondendo dúvidas (quem souber). É até bom para eu ter uma ideia de se tem gente entendendo ;-)

ACH2024

Aula 25

Processamento cossequencial Ordenação externa (Intercalacão Balanceada)

Profa. Ariane Machado Lima

Aulas anteriores

Algoritmos e estruturas de dados para lida com memória secundária

- Organização interna de arquivos
- Acesso à memória secundária (por blocos - seeks)
- Tipos de alocação de arquivos na memória secundária:
 - Sequencial (ordenado e não ordenado)
 - Ligada
 - Indexada
 - Árvores-B
 - Hashing (veremos também hashing em memória principal)

Algoritmos de processamento cossequencial e ordenação em disco

Processamento cossequencial

- Processamento coordenado e sequencial de duas ou mais listas (no intuito de formar uma lista única)
- Exemplo: obtenção da intersecção ou união de listas ordenadas, podendo percorrer cada lista apenas uma vez

Input	Array 1	10	15	22	80	
	Array 2	5	10	11	22	70

Output:

Union :5 10 11 15 22 70 80 90
Intersection :10 22

Processamento cossequencial

- Etapas:
 - Inicialização (abrir arquivos, inicializar variáveis)
 - Sincronização (como avançar em cada lista)
 - Condições de fim de lista (o que fazer)
 - Reconhecimento de possíveis fontes de erros (ex: pode haver duplicações? Elementos fora de ordem?)

É um bom exercício para fazerem!

Input	Array 1	10	15	22	80	
	Array 2	5	10	11	22	70

Output:

Union :5 10 11 15 22 70 80 90
Intersection :10 22

Processamento cossequencial

- Base para a ordenação externa
- Além disso: maximizar a manipulação em memória para minimizar o número de acessos ao disco

Ordenação externa

- Objetivo: ordenar um arquivo muito grande, que não cabe inteiro na memória
- O que fazer: ordenar pedaços desse arquivo (em memória), e depois combinar os pedaços

Ordenação Externa

- A ordenação externa consiste em ordenar arquivos de tamanho maior que a memória interna disponível.
- Os métodos de ordenação externa são muito diferentes dos de ordenação interna.
- Na ordenação externa os algoritmos devem diminuir o número de acesso as unidades de memória externa. Ou seja, seeks.
- Nas memórias externas, os dados ficam em um arquivo seqüencial.
- Apenas um registro pode ser acessado em um dado momento. Essa é uma restrição forte se comparada com as possibilidades de acesso em um vetor. (acesso sequencial x acesso aleatório)
- Logo, os métodos de ordenação interna são inadequados para ordenação externa.
- Técnicas de ordenação diferentes devem ser utilizadas.

Durante a ordenação. Por isso fitas são muitas vezes usadas para essa tarefa!

Ordenação Externa

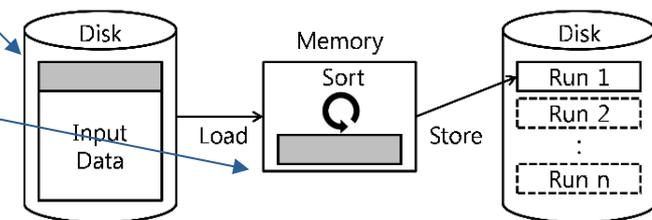
Fatores que determinam as diferenças das técnicas de ordenação externa:

1. Custo para acessar um item é algumas ordens de grandeza maior.
2. O custo principal na ordenação externa é relacionado a transferência de dados entre a memória interna e externa.
3. Existem restrições severas de acesso aos dados.
4. O desenvolvimento de métodos de ordenação externa é muito dependente do estado atual da tecnologia.
5. A variedade de tipos de unidades de memória externa torna os métodos dependentes de vários parâmetros.
6. Assim, apenas métodos gerais serão apresentados.

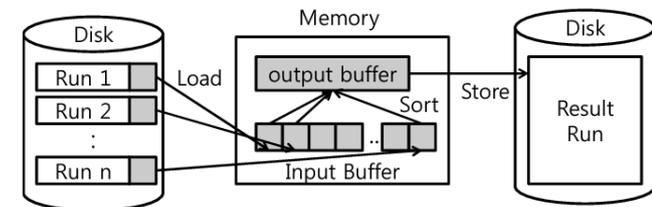
Ordenação Externa

- O método mais importante é o de ordenação por intercalação.
- Intercalar significa combinar dois ou mais blocos ordenados em um único bloco ordenado. (lembra do mergeSort?)
- A intercalação é utilizada como uma operação auxiliar na ordenação.
- Estratégia geral dos métodos de ordenação externa:
 1. Quebre o arquivo em blocos do tamanho da memória interna disponível.
 2. Ordene cada bloco na memória interna.
 3. Intercale os blocos ordenados, fazendo várias passadas sobre o arquivo.
 4. A cada passada são criados blocos ordenados cada vez maiores, até que todo o arquivo esteja ordenado.

Cuidado: esse bloco não é o bloco do disco – um melhor termo seria **“segmento de dados”**



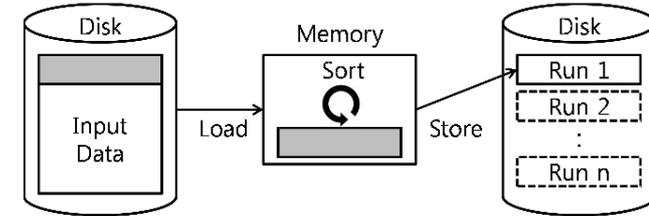
(a) Run formation phase



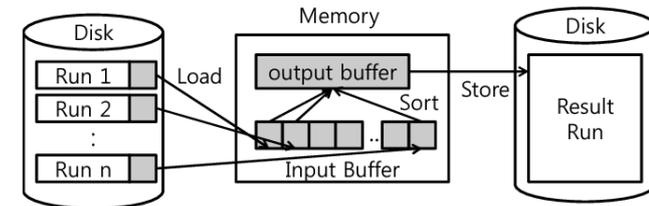
(b) Merge phase

Ordenação Externa

- Os algoritmos para ordenação externa devem reduzir o número de passadas sobre o arquivo.
- Uma boa medida de complexidade de um algoritmo de ordenação por intercalação é o número de vezes que um item é lido ou escrito na memória auxiliar.
- Os bons métodos de ordenação geralmente envolvem no total menos do que dez passadas sobre o arquivo.



(a) Run formation phase



(b) Merge phase

Ordenação externa: histórico das fitas...

- Por um bom tempo as fitas magnéticas eram o dispositivo comum para memória secundária, e portanto utilizadas para a ordenação externa
- Até hoje, mesmo que discos sejam usados para a ordenação, pensar em ordenação usando fitas é uma boa abstração para o entendimento dos algoritmos de ordenação externa:
 - Há um conjunto de fitas de entrada com dados a serem ordenados (intercalados)
 - Há um conjunto de fitas de saída que recebem o resultado da intercalação

Pense nessas fitas como sendo cada uma delas um disco com acesso sequencial...
(se não tiver o nr de discos necessários, trechos sequenciais desse disco)

Principais abordagens gerais de ordenação externa

- **Intercalação balanceada**
- Geração de corridas iniciais usando **Seleção por substituição**
- Intercalação **Polifásica**

Principais abordagens gerais de ordenação externa

- **Intercalação balanceada**
- Geração de corridas iniciais usando **Seleção por substituição**
- Intercalação **Polifásica**

Intercalação Balanceada de Vários Caminhos

- Considere um arquivo armazenado em uma fita de entrada:

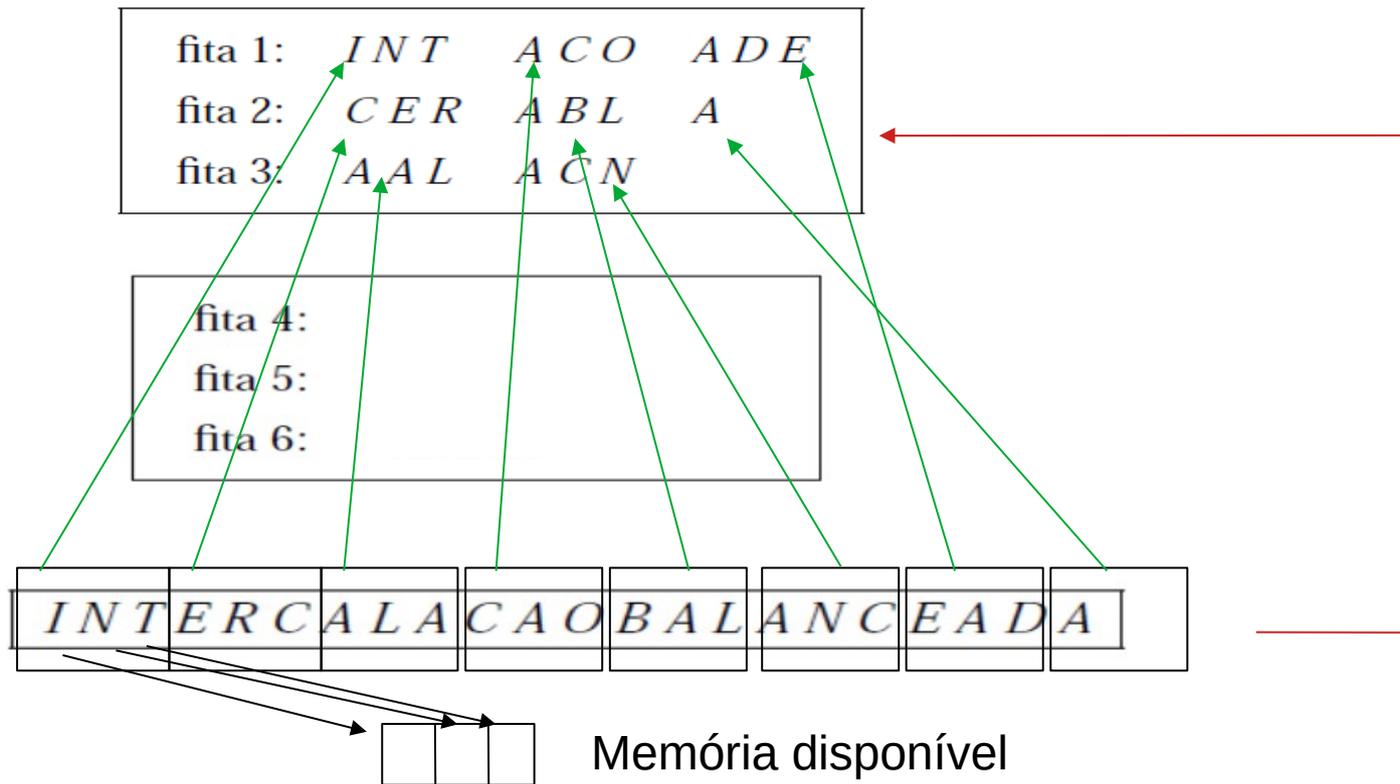
INTERCALACA OBALANCEADA

Considere que cada letra é um registro

- Objetivo:
 - Ordenar os 22 registros e colocá-los em uma fita de saída.
- Os registros são lidos um após o outro.
- Considere uma memória interna com capacidade para para três registros. ($M = 3$)
- Considere que esteja disponível seis unidades de fita magnética.
2f fitas: f fitas de entrada e f fitas de saída (neste exemplo, $f = 3$)

Intercalação Balanceada de Vários Caminhos

- Fase de criação dos segmentos ordenados (**corridas**).



Ordena e joga de forma **balanceada** pelas f fitas de entrada (metade das fitas disponíveis inicialmente) – cada vez coloco em uma fita

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:

fita 1: INT ACO ADE

fita 2: CER ABL A

fita 3: AAL ACN

fita 4:

fita 5:

fita 6:

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:

fita 1: INT ACO ADE

fita 2: CER ABL A

fita 3: AAL ACN

fita 4: A

fita 5:

fita 6:

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:

fita 1: INT ACO ADE

fita 2: CER ABL A

fita 3: AAL ACN

fita 4: AA

fita 5:

fita 6:

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:

fita 1: INT ACO ADE

fita 2: CER ABL A

fita 3: AAL ACN

fita 4: AAC

fita 5:

fita 6:

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:

fita 1: INT ACO ADE

fita 2: CER R ABL A

fita 3: AAL L ACN

fita 4: **AACE**

fita 5:

fita 6:

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:

fita 1: *INT ACO ADE*

fita 2: *CER ABL A*

fita 3: *AAL ACN*

fita 4: *AACEI*

fita 5:

fita 6:

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:

fita 1: *INT* *ACO* *ADE*

fita 2: *CER* *ABL* *A*

fita 3: *AAL* *ACN*

fita 4: *AACEIL*

fita 5:

fita 6:

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:

fita 1: *INT* ACO *ADE*

fita 2: *CER* ABL *A*

fita 3: *AAL* ACN

fita 4: *AACEILN*

fita 5:

fita 6:

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:

fita 1: *INT* *ACO* *ADE*

fita 2: *CER* *ABL* *A*

fita 3: *AAL* *ACN*

fita 4: *AACEILNR*

fita 5:

fita 6:

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:

fita 1: *INT _ ACO ADE*

fita 2: *CER _ ABL A*

fita 3: *AAL _ ACN*

fita 4: *AACEILNRT*

fita 5:

fita 6:

Qual o tamanho máximo de cada corrida intercalada?

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:

fita 1: *INT_ ACO ADE*

fita 2: *CER_ ABL A*

fita 3: *AAL_ ACN*

fita 4: *AACEILNRT*

fita 5:

fita 6:

Qual o tamanho máximo de cada corrida intercalada

$M * f$

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:
 1. O primeiro registro de cada fita é lido.
 2. Retire o registro contendo a menor chave.
 3. Armazene-o em uma fita de saída.
 4. Leia um novo registro da fita de onde o registro retirado é proveniente.
 5. Ao ler o terceiro registro de uma corrida sua fita fica inativa.
 6. A fita é reativada quando o terceiro registro das outras fitas forem lidos.
 7. Neste instante 1 corrida de nove registros ordenados foi formado na fita de saída. (por que nove?)
 8. Repita o processo para as corridas restantes.

fita 1:	<i>I N T</i>	<u><i>A C O</i></u>	<i>A D E</i>
fita 2:	<i>C E R</i>	<u><i>A B L</i></u>	<i>A</i>
fita 3:	<i>A A L</i>	<u><i>A C N</i></u>	

fita 4:	<i>A A C E I L N R T</i>
fita 5:	
fita 6:	

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:
 1. O primeiro registro de cada fita é lido.
 2. Retire o registro contendo a menor chave.
 3. Armazene-o em uma fita de saída.
 4. Leia um novo registro da fita de onde o registro retirado é proveniente.
 5. Ao ler o terceiro registro de uma corrida sua fita fica inativa.
 6. A fita é reativada quando o terceiro registro das outras fitas forem lidos.
 7. Neste instante 1 corrida de nove registros ordenados foi formado na fita de saída. $9 = 3 \text{ (fitas)} * 3 \text{ (tamanho de cada segmento, nesta primeira passada = tam da memória disponível)}$
 8. Repita o processo para as corridas restantes.

fita 1:	<i>INT</i>	<u><i>ACO</i></u>	<i>ADE</i>
fita 2:	<i>CER</i>	<u><i>ABL</i></u>	<i>A</i>
fita 3:	<i>AAL</i>	<u><i>ACN</i></u>	

fita 4:	<i>AACEILNRT</i>
fita 5:	
fita 6:	

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:
 1. O primeiro registro de cada fita é lido.
 2. Retire o registro contendo a menor chave.
 3. Armazene-o em uma fita de saída.
 4. Leia um novo registro da fita de onde o registro retirado é proveniente.
 5. Ao ler o terceiro registro de uma corrida sua fita fica inativa.
 6. A fita é reativada quando o terceiro registro das outras fitas forem lidos.
 7. Neste instante 1 corrida de nove registros ordenados foi formado na fita de saída.
 8. Repita o processo para as corridas restantes.

fita 1:	<i>I N T</i>	<i>A <u>C</u> O</i>	<i>A D E</i>
fita 2:	<i>C E R</i>	<i><u>A</u> B L</i>	<i>A</i>
fita 3:	<i>A A L</i>	<i><u>A</u> C N</i>	

fita 4:	<i>AACEILNRT</i>
fita 5:	<i>A</i>
fita 6:	

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:
 1. O primeiro registro de cada fita é lido.
 2. Retire o registro contendo a menor chave.
 3. Armazene-o em uma fita de saída.
 4. Leia um novo registro da fita de onde o registro retirado é proveniente.
 5. Ao ler o terceiro registro de uma corrida sua fita fica inativa.
 6. A fita é reativada quando o terceiro registro das outras fitas forem lidos.
 7. Neste instante 1 corrida de nove registros ordenados foi formado na fita de saída.
 8. Repita o processo para as corridas restantes.

fita 1:	<i>I N T</i>	<i>A C O</i>	<u><i>A D E</i></u>
fita 2:	<i>C E R</i>	<i>A B L</i>	<u><i>A</i></u>
fita 3:	<i>A A L</i>	<i>A C N</i>	<u> </u>

fita 4:	<i>AACEILNRT</i>
fita 5:	<i>AAABCCLNO</i>
fita 6:	

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:
 1. O primeiro registro de cada fita é lido.
 2. Retire o registro contendo a menor chave.
 3. Armazene-o em uma fita de saída.
 4. Leia um novo registro da fita de onde o registro retirado é proveniente.
 5. Ao ler o terceiro registro de uma corrida sua fita fica inativa.
 6. A fita é reativada quando o terceiro registro das outras fitas forem lidos.
 7. Neste instante 1 corrida de nove registros ordenados foi formado na fita de saída.
 8. Repita o processo para as corridas restantes.

fita 1:	<i>INT</i>	<i>ACO</i>	<i>ADE</i>	<u> </u>
fita 2:	<i>CER</i>	<i>ABL</i>	<i>A</i>	<u> </u>
fita 3:	<i>AAL</i>	<i>ACN</i>	<u> </u>	

fita 4:	<i>AACEILNRT</i>
fita 5:	<i>AAABCCLNO</i>
fita 6:	<i>AADE</i>

Se tivéssemos mais segmentos, em qual fita colocaríamos?

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:
 1. O primeiro registro de cada fita é lido.
 2. Retire o registro contendo a menor chave.
 3. Armazene-o em uma fita de saída.
 4. Leia um novo registro da fita de onde o registro retirado é proveniente.
 5. Ao ler o terceiro registro de uma corrida sua fita fica inativa.
 6. A fita é reativada quando o terceiro registro das outras fitas forem lidos.
 7. Neste instante 1 corrida de nove registros ordenados foi formado na fita de saída.
 8. Repita o processo para as corridas restantes.

fita 1:	<i>INT</i>	<i>ACO</i>	<i>ADE</i>	<u> </u>
fita 2:	<i>CER</i>	<i>ABL</i>	<i>A</i>	<u> </u>
fita 3:	<i>AAL</i>	<i>ACN</i>	<u> </u>	

fita 4:	<i>AACEILNRT</i>
fita 5:	<i>AAABCCLNO</i>
fita 6:	<i>AADE</i>

Se tivéssemos mais segmentos, em qual fita colocaríamos?

Na 4, depois 5, depois 6, depois 4, ...

(ou seja, de forma balanceada, como fizemos com as corridas iniciais)

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:
 1. O primeiro registro de cada fita é lido.
 2. Retire o registro contendo a menor chave.
 3. Armazene-o em uma fita de saída.
 4. Leia um novo registro da fita de onde o registro retirado é proveniente.
 5. Ao ler o terceiro registro de uma corrida sua fita fica inativa.
 6. A fita é reativada quando o terceiro registro das outras fitas forem lidos.
 7. Neste instante 1 corrida de nove registros ordenados foi formado na fita de saída.
 8. Repita o processo para as corridas restantes.

fita 1:	<i>I N T</i>	<i>A C O</i>	<i>A D E</i>	<u> </u>
fita 2:	<i>C E R</i>	<i>A B L</i>	<i>A</i>	<u> </u>
fita 3:	<i>A A L</i>	<i>A C N</i>	<u> </u>	

fita 4:	<i>AACEILNRT</i>
fita 5:	<i>AAABCCLNO</i>
fita 6:	<i>AADE</i>

E aí, o que fazemos?

Intercalação Balanceada de Vários Caminhos

- Fase de intercalação - Primeira passada:
 1. O primeiro registro de cada fita é lido.
 2. Retire o registro contendo a menor chave.
 3. Armazene-o em uma fita de saída.
 4. Leia um novo registro da fita de onde o registro retirado é proveniente.
 5. Ao ler o terceiro registro de uma corrida sua fita fica inativa.
 6. A fita é reativada quando o terceiro registro das outras fitas forem lidos.
 7. Neste instante 1 corrida de nove registros ordenados foi formado na fita de saída.
 8. Repita o processo para as corridas restantes.

fita 1:

fita 2:

fita 3:

fita 4: AACEILNRT

fita 5: AABCCCLNO

fita 6: AADE

Próxima passada faz o mesmo, mas agora usando fitas as 4 a 6 como entrada e as fitas 1 a 3 como saída, e assim sucessivamente até gerar uma única corrida

Intercalação Balanceada de Vários Caminhos

- Quantas passadas são necessárias para ordenar um arquivo de tamanho arbitrário? (de intercalação)

Intercalação Balanceada de Vários Caminhos

- Quantas passadas são necessárias para ordenar um arquivo de tamanho arbitrário?
 - Seja n , o número de registros do arquivo.
 - Suponha que cabem m registros na memória interna.
 - A primeira etapa produz n/m corridas ordenadas.
 - Seja $P(n)$ o número de passadas para a fase de intercalação.
 - Seja f o número de fitas utilizadas em cada passada.
 - Assim:

$$P(n) = \left\lceil \log_f \left\lceil \frac{n}{m} \right\rceil \right\rceil$$

No exemplo acima, $n=22$, $m=3$ e $f=3$ temos:

$$P(n) = \left\lceil \log_3 \left\lceil \frac{22}{3} \right\rceil \right\rceil = 2.$$

Intercalação Balanceada de Vários Caminhos

- Quantas passadas são necessárias para ordenar um arquivo de tamanho arbitrário?
 - Seja n , o número de registros do arquivo.
 - Suponha que cabem m registros na memória interna.
 - A primeira etapa produz n/m corridas ordenadas.
 - Seja $P(n)$ o número de passadas para a fase de intercalação.
 - Seja f o número de fitas utilizadas em cada passada.
 - Assim:

$$P(n) = \left\lceil \log_f \left\lceil \frac{n}{m} \right\rceil \right\rceil$$

No exemplo acima, $n=22$, $m=3$ e $f=3$ temos:

$$P(n) = \left\lceil \log_3 \left\lceil \frac{22}{3} \right\rceil \right\rceil = 2.$$

Obs: quanto maior o m melhor, certo? Então como precisam ser os algoritmos de ordenação em memória principal para gerar as corridas iniciais?

Intercalação Balanceada de Vários Caminhos

- Quantas passadas são necessárias para ordenar um arquivo de tamanho arbitrário?
 - Seja n , o número de registros do arquivo.
 - Suponha que cabem m registros na memória interna.
 - A primeira etapa produz n/m corridas ordenadas.
 - Seja $P(n)$ o número de passadas para a fase de intercalação.
 - Seja f o número de fitas utilizadas em cada passada.
 - Assim:

$$P(n) = \left\lceil \log_f \left\lceil \frac{n}{m} \right\rceil \right\rceil$$

No exemplo acima, $n=22$, $m=3$ e $f=3$ temos:

$$P(n) = \left\lceil \log_3 \left\lceil \frac{22}{3} \right\rceil \right\rceil = 2.$$

Obs: quanto maior o m melhor, certo? Então como precisam ser os algoritmos de ordenação em memória principal para gerar as corridas iniciais?

In loco!!!

Intercalação Balanceada de Vários Caminhos

- No exemplo foram utilizadas $2f$ fitas para uma intercalação-de- f -caminhos.
- É possível usar apenas $f + 1$ fitas:
 - Encaminhe todas as corridas intercaladas para uma única fita de saída.
 - Redistribua as corridas entre as fitas de onde elas foram lidas.
 - O custo envolvido é uma passada a mais em cada intercalação.
- No caso do exemplo de 22 registros, apenas quatro fitas seriam suficientes:
 - A intercalação das corridas a partir das fitas 1, 2 e 3 seria toda dirigida para a fita 4.
 - Ao final, a segunda e a terceira corridas ordenadas de nove registros seriam transferidas de volta para as fitas 1 e 2, e a fita 3 usada como fita de saída

fitas 1: *INT ACO ADE*
fitas 2: *CER ABL A*
fitas 3: *AAL ACN*

fita 4: *AACEILNRT AAABCCLNO AADE*



fitas 1: *AAABCCLNO*
fitas 2: *AADE*
fitas 3:

fita 4: *AACEILNRT*

Qual a vantagem? Conseguir usar poucas fitas que tenha.

Ordenação externa em disco

- Na ordenação externa em fita, tenho $f+1$ fitas distintas → cada uma sendo lida sequencialmente
- Na ordenação externa em disco:
 - poderia semelhantemente utilizar $f+1$ discos, cada um deles sendo lido sequencialmente
 - ou se não tiver vários discos (pelo menos não tantos quantos eu desejaria para um dado f), “simular essas f fitas” em f cilindros distintos (se os dados lá couberem)
 - O problema é que para ler de cilindros distintos tenho que fazer um novo seek, então melhor já trazer e processar pelo menos todo o bloco (e não apenas um registro)

Ordenação externa em disco

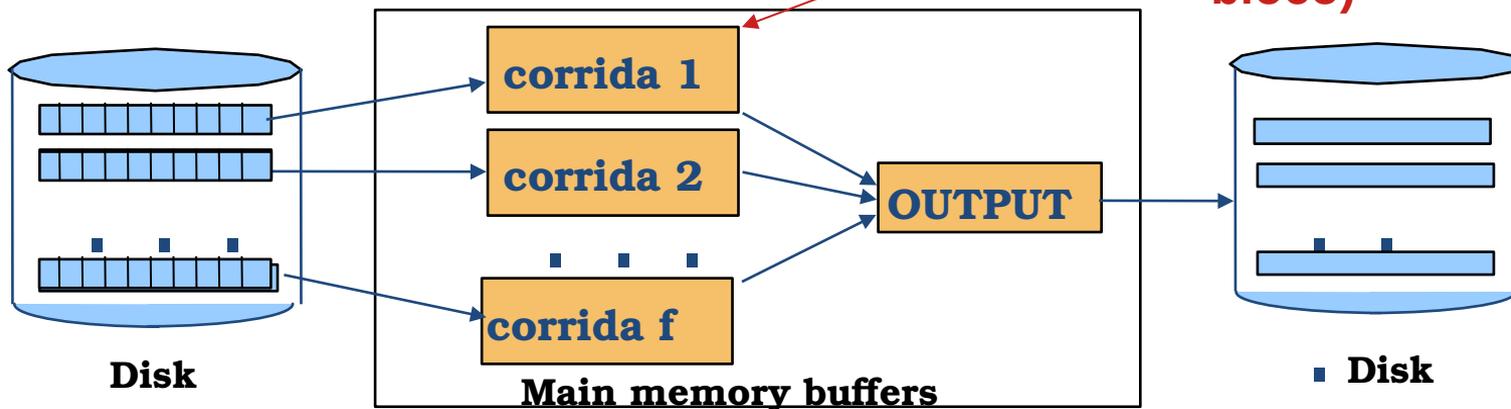
Tamanhos considerados:

- N: tamanho do arquivo original EM NÚMERO DE BLOCOS DE DISCO
- M: Tamanho da memória interna disponível para a ordenação EM NÚMERO DE BLOCOS (páginas)

Intercalação em f vias

- Intercala f corridas de cada vez

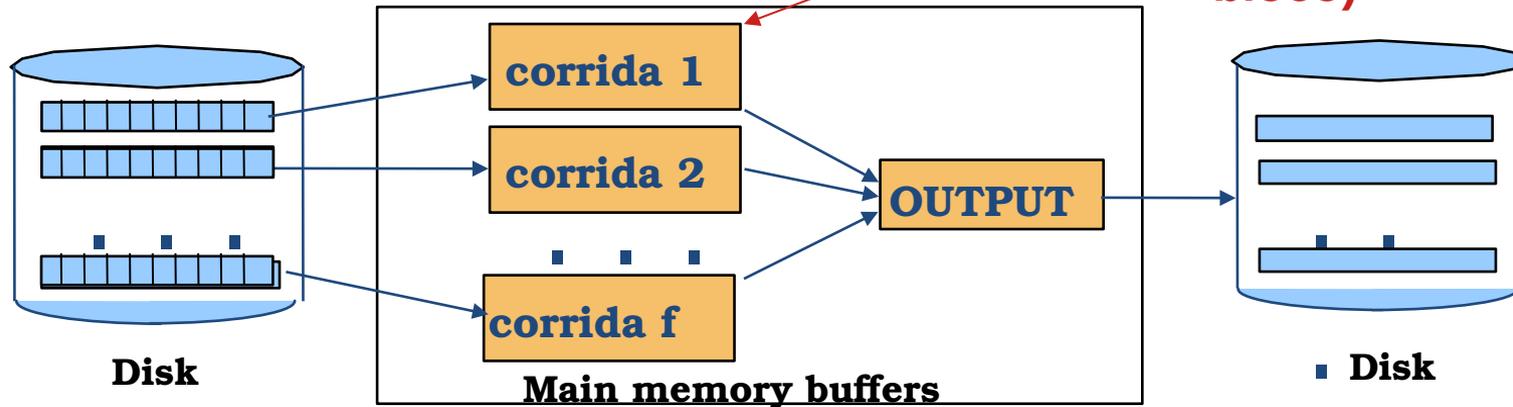
Na verdade, apenas um trecho da corrida i (no mínimo um bloco)



No máximo, $f = ?$

Intercalação em f vias

- Intercala f corridas de cada vez



Na verdade, apenas um trecho da corrida i (no mínimo um bloco)

- No máximo, $f = M - 1$ ($M =$ número de blocos da memória disponível, 1 bloco será utilizado para saída – resultado da intercalação)

Exemplo

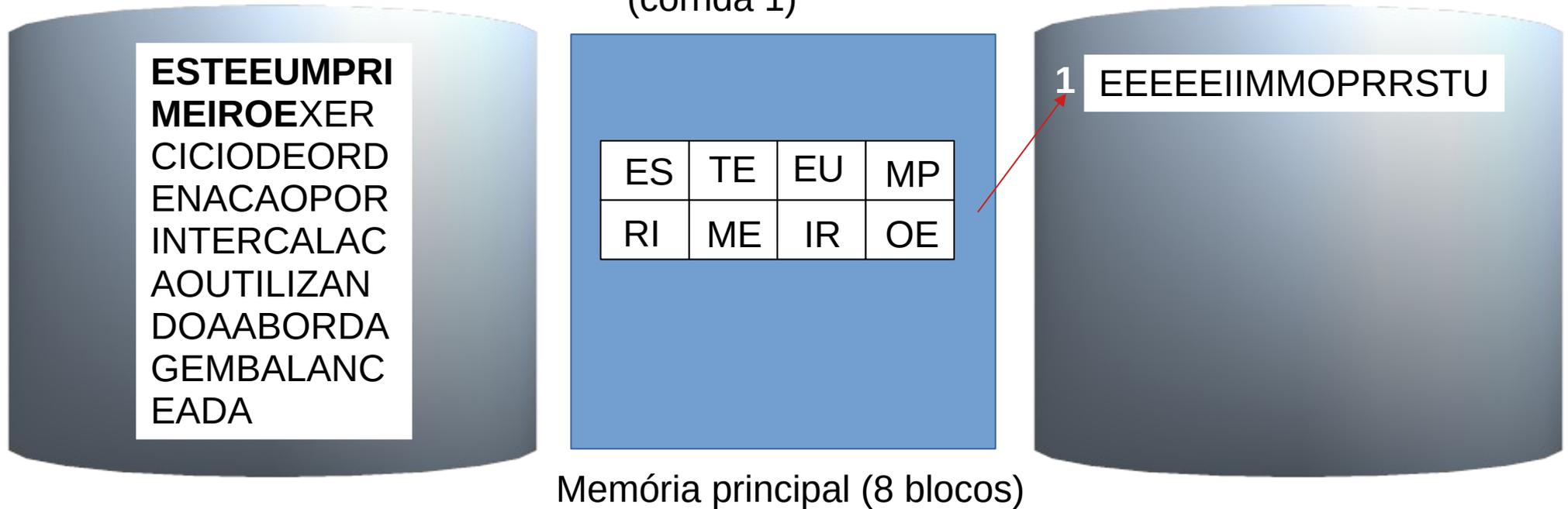
Considere uma memória de 8 blocos, e o uso de $f = 3$ vias, sendo:

- 6 blocos usados para as corridas, ou seja, 2 blocos disponíveis para cada uma das f corridas (ou vias)
- 2 blocos para o buffer de saída
- Cada bloco cabem 2 chaves

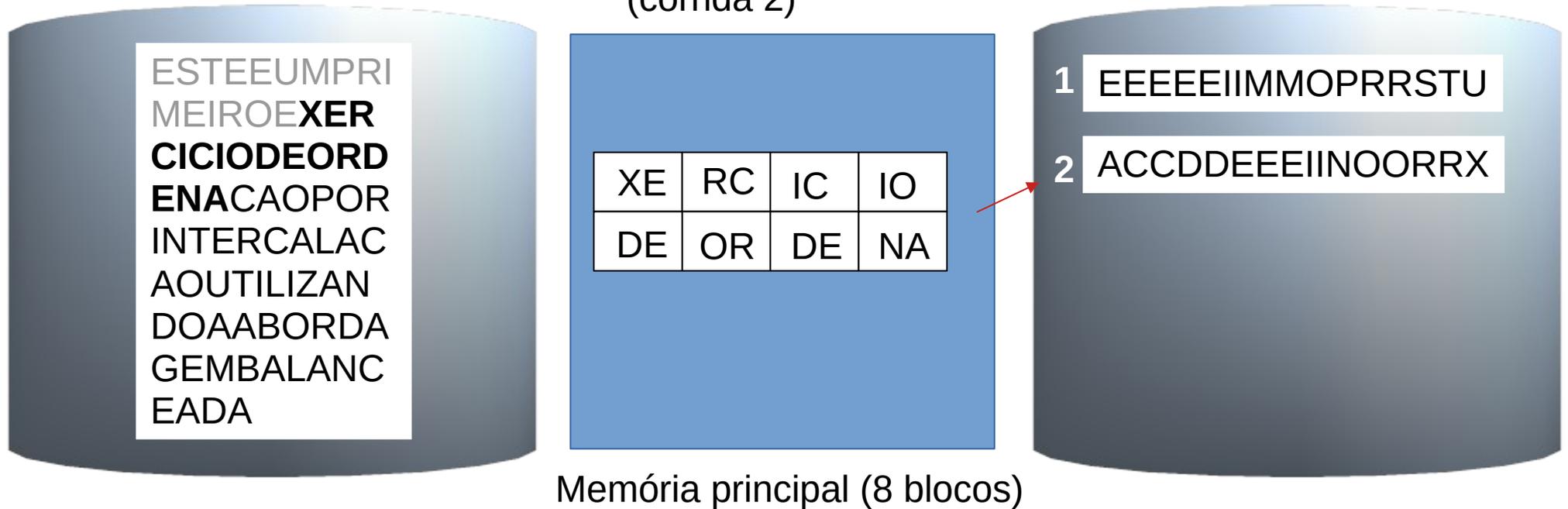
Mostre todos os passos da ordenação das chaves (letras) de um arquivo que estão inicialmente nesta ordem:

ESTEEUMPRIMEIROEXERCICIODEORDENACAOPORINTERCA
LACAOUTILIZANDO A BORDA GEM BALANCEADA

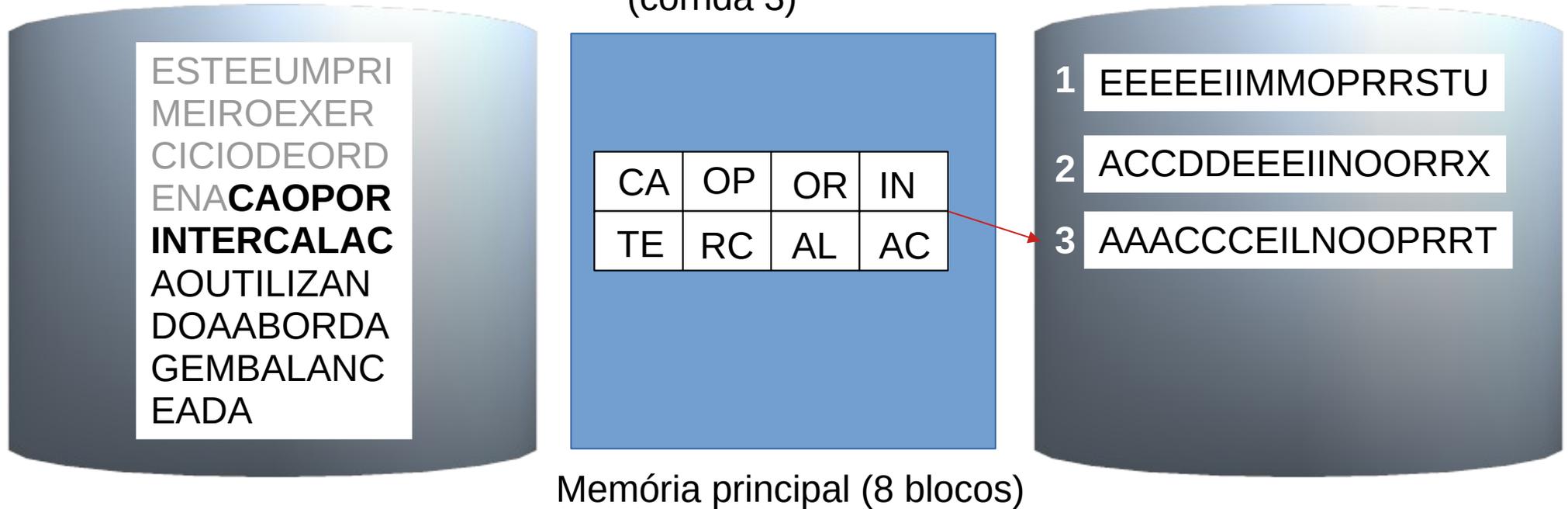
Geração das corridas iniciais



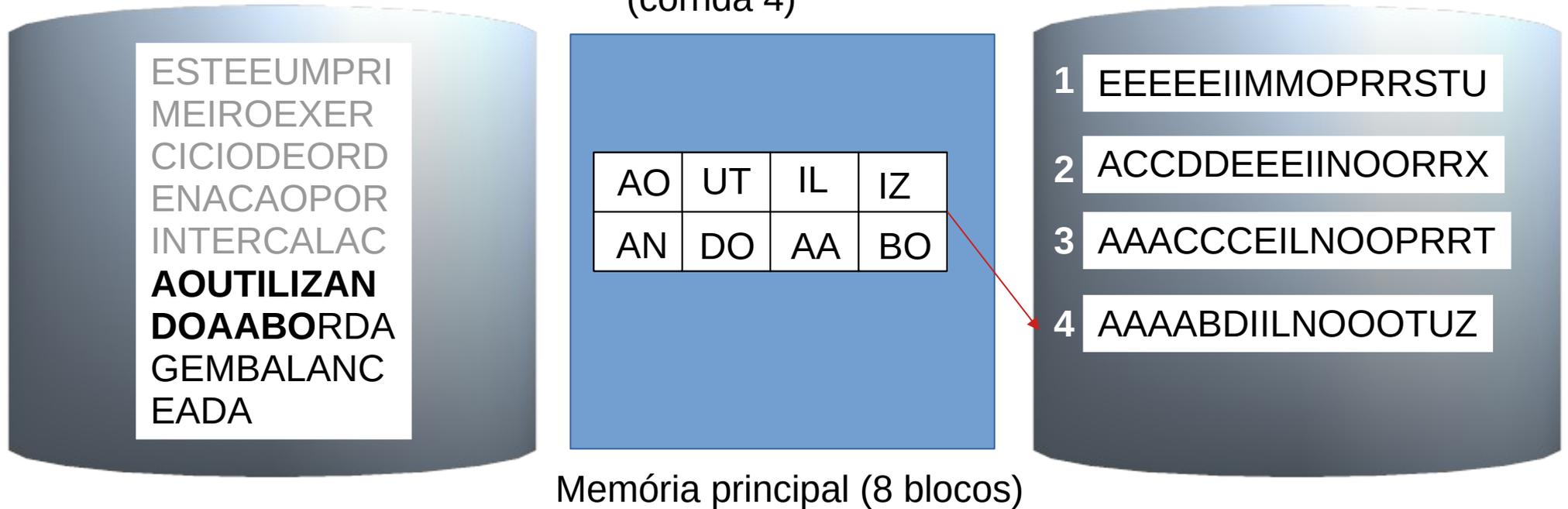
Geração das corridas iniciais



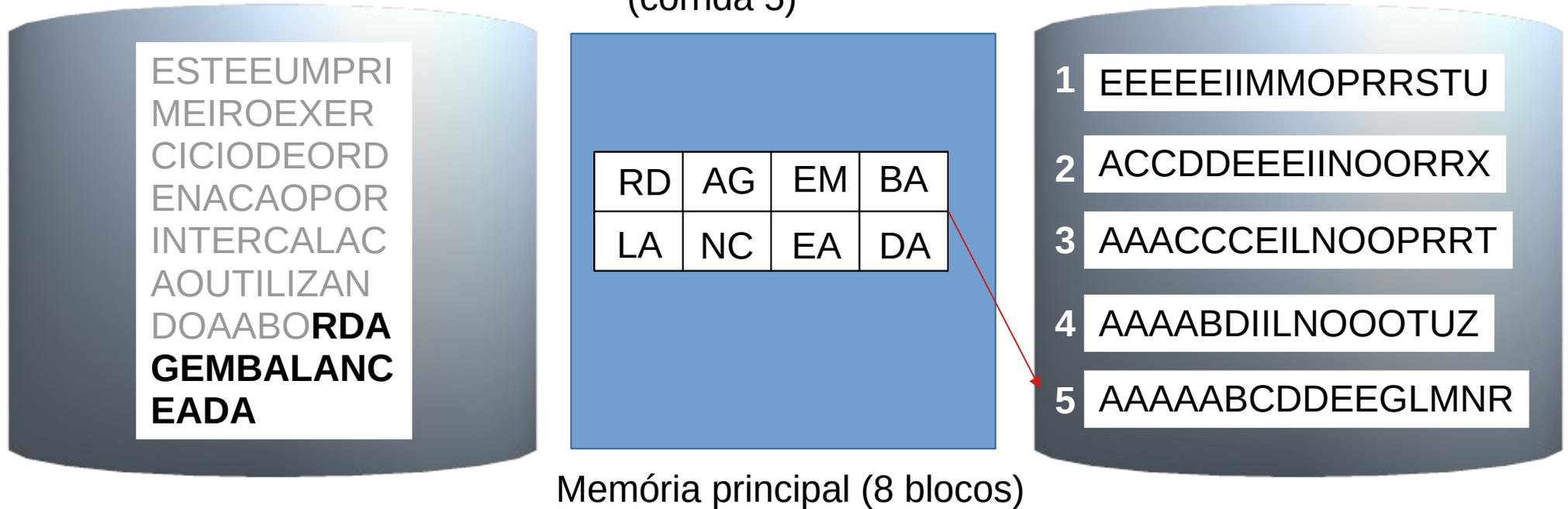
Geração das corridas iniciais



Geração das corridas iniciais

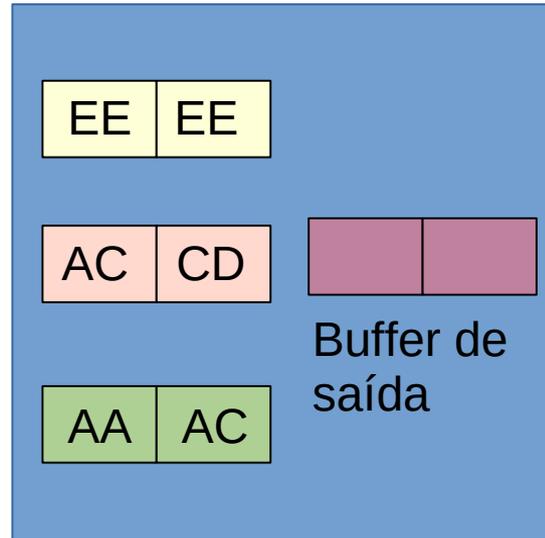
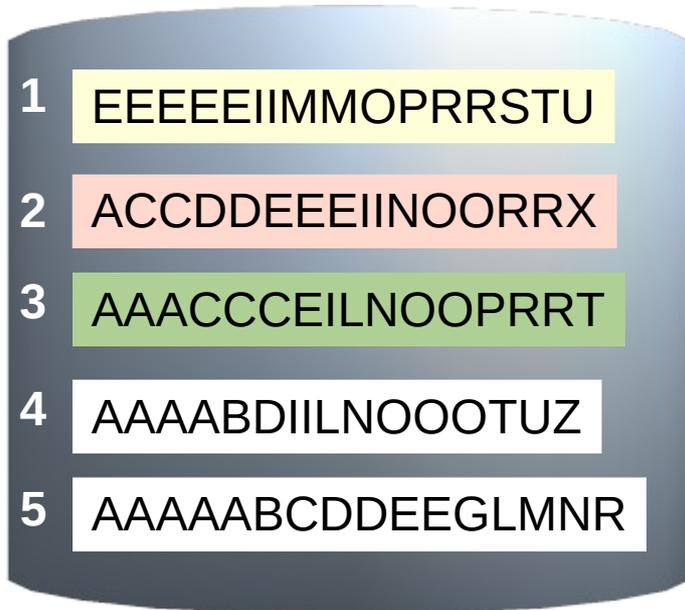


Geração das corridas iniciais



1º Passo de Intercalação

Vai intercalando as 3 corridas até lotar o buffer de saída;
Decarrega o buffer de saída no disco (na nova corrida sendo formada) e continua



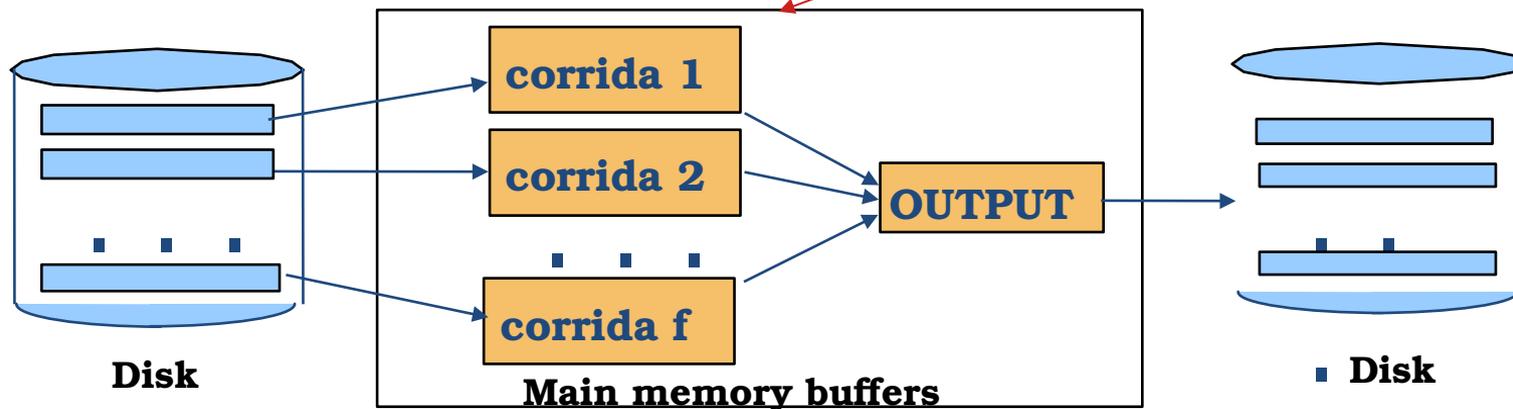
Memória principal (8 blocos)



(Continuar...)

Intercalação em f vias

- Intercala f corridas de cada vez



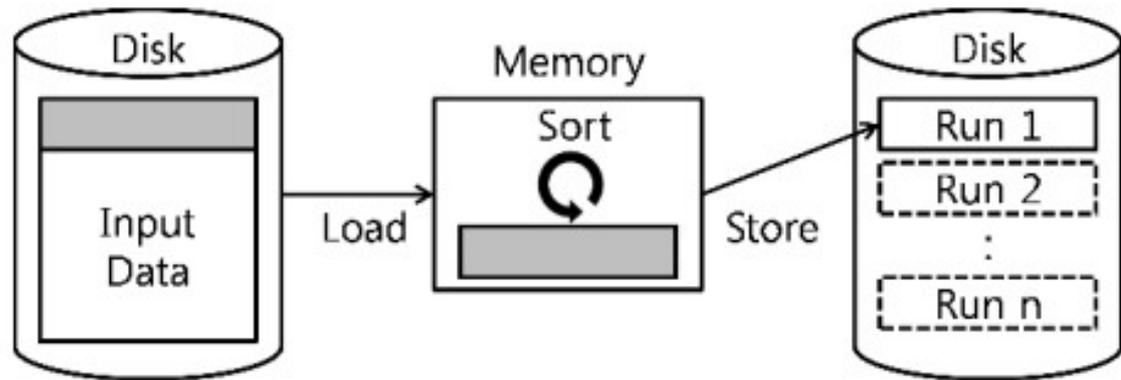
Na verdade, apenas um trecho da corrida i (no mínimo um bloco)

- Para simplificar os cálculos: memória disponível tem $M + 1$ blocos, M sendo usados para as f vias.
- Note que $f = M/k$, k sendo o nr de blocos que vou armazenar em memória de cada corrida ($k = M/f$)

Intercalação em f vias ($f \leq M$)

- Fase de **ordenação** – geração das corridas iniciais (quanto maiores as corridas iniciais, melhor!):
 - lê M blocos do arquivo de cada vez (lota a memória) e ordena formando uma corrida
 - → $\text{ceil}(N/M)$ corridas

Posso usar qualquer algoritmo de ordenação interna?



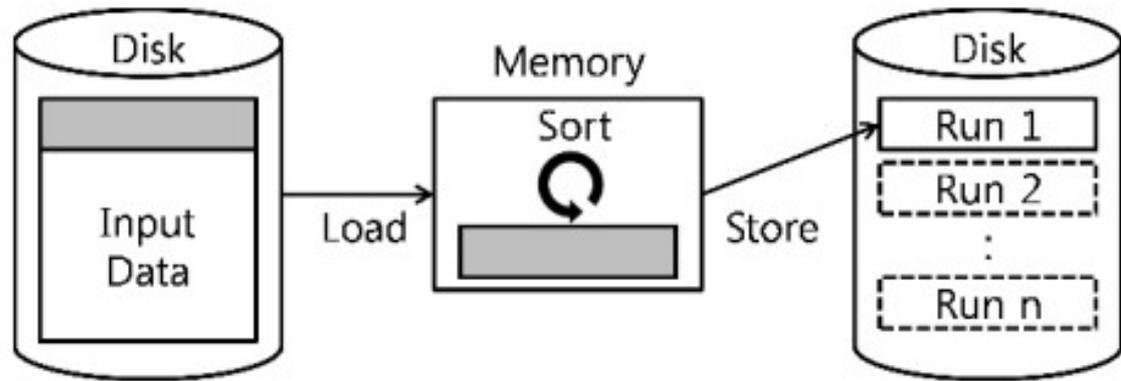
(a) Run formation phase

Lee, Joonhee et al. "External Mergesort for Flash-Based Solid State Drives." IEEE Transactions on Computers 65 (2016): 1518-1527.

Intercalação em f vias ($f \leq M$)

- Fase de **ordenação** – geração das corridas iniciais (quanto maiores as corridas iniciais, melhor!):
 - lê M blocos do arquivo de cada vez (lota a memória) e ordena formando uma corrida
 - → $\text{ceil}(N/M)$ corridas

Obs: necessário usar um algoritmo de ordenação interna que ordene *in loco* (sem usar vetor auxiliar)



(a) Run formation phase

Lee, Joonhee et al. "External Mergesort for Flash-Based Solid State Drives." IEEE Transactions on Computers 65 (2016): 1518-1527.

Intercalação em f vias (f ≤ M)

- Fase de **ordenação** – geração das corridas iniciais (quanto maiores as corridas iniciais, melhor!):

- lê M blocos do arquivo de cada vez (lota a memória) e ordena formando uma corrida
- → $\text{ceil}(N/M)$ corridas

Obs: necessário usar um algoritmo de ordenação interna que ordene *in loco* (sem usar vetor auxiliar)

- Fase de intercalação:

- M blocos de memória precisam ser divididos para as f corridas → cada via conterá M/f blocos (ou seja, 1/f da corrida original) → cada via deverá ser lida f vezes (**posteriormente t vezes**, dependendo do tamanho da corrida)

- Cada passo (passada sobre todo o arquivo):

Apenas f seeks se cada corrida estivesse em um disco

- Leitura: necessários $f * t$ seeks (f vias (corridas), t vezes cada, sendo cada vez uma leitura sequencial de blocos)
- CPU: Cada intercalação - O(N) no total
- Escrita: N/b seeks (b = número de blocos do buffer de saída)

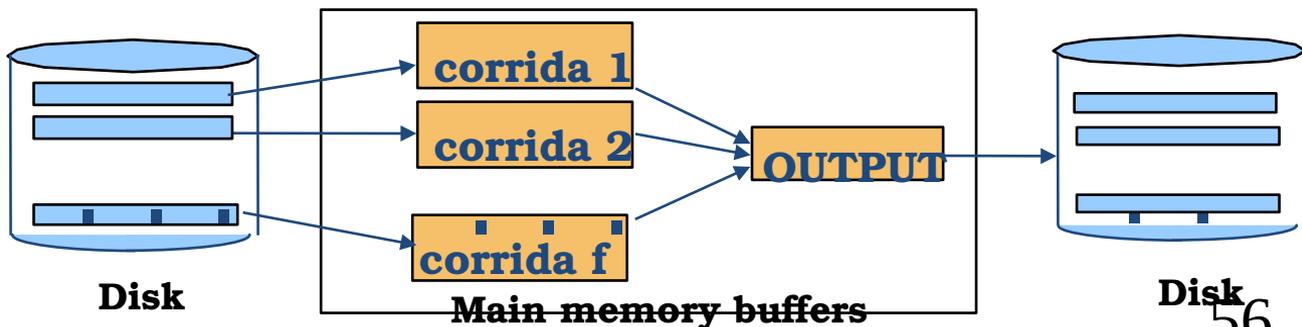
- Nr de passos (incluindo geração inicial) = $1 + P(N) = 1 + \text{ceil}(\log_f \text{ceil}(N/M))$

- Aumentar o f diminui o nr de passos mas aumenta o nr de seeks de leituras

- f = M → não há leitura sequencial da corrida se ela não estiver em um disco dedicado a ela

* **Custo = $2 * N * (1 + P(N))$ seeks**

(leitura e escrita de cada bloco P(N)+1 vezes)



Exemplo: nr de passos P de intercalações em f vias

		f (nr de corridas trabalhadas em memória)					
		2	4	8	16	128	256
N/M (nr de corridas iniciais)	100	7	4	3	2	1	1
	1.000	10	5	4	3	2	2
	10.000	14	7	5	4	2	2
	100.000	17	9	6	5	3	3
	1.000.000	20	10	7	5	3	3
	10.000.000	24	12	8	6	4	3
	100.000.000	27	14	9	7	4	4
	1.000.000.000	30	15	10	8	5	4

Referências

- Cap 4 do livro do Ziviani (parte final)
- ELMASRI, R.; NAVATHE, S. B. **Fundamentals of Database Systems**. 4 ed. Ed Pearson/Addisonn-Wesley. Seção 15.2
- RAMAKRISHNAN, R.; GEHRKE, J. **Database Management Systems**. 3 ed. Ed. McGraw-Hill. Cap 13