

ÁRVORE-B⁺

Organização e Recuperação de Dados
Profa. Valéria

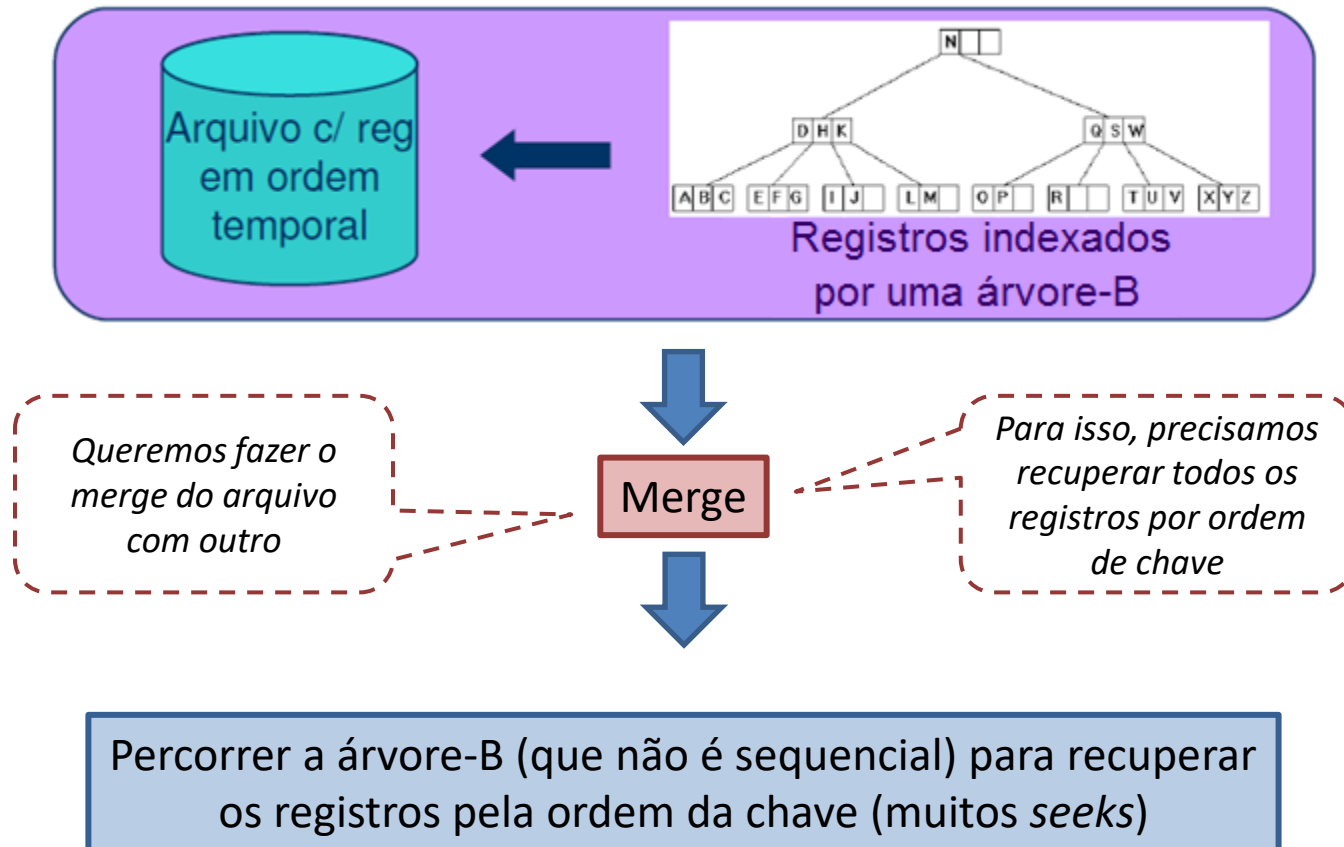
UEM – CTC – DIN

TIPOS DE ACESSO

- Até este ponto, vimos como organizar um arquivo para dois tipos de acesso: **sequencial** e **indexado**
- Um arquivo **sequencial** não utiliza nenhuma estrutura auxiliar
 - Se o arquivo estiver **ordenado**, é possível fazer busca binária → mais rápida que a sequencial, mas impõe restrições ao arquivo
 - O **acesso sequencial ordenado** é rápido, mas o aleatório é lento
- Um arquivo **indexado** utiliza um índice como auxiliar
 - **Árvores-B** são boas para a manutenção de índices em disco
 - As buscas são feitas rapidamente no índice e o acesso no arquivo de registros é direto
 - O **acesso aleatório é rápido**, mas o sequencial ordenado é lento

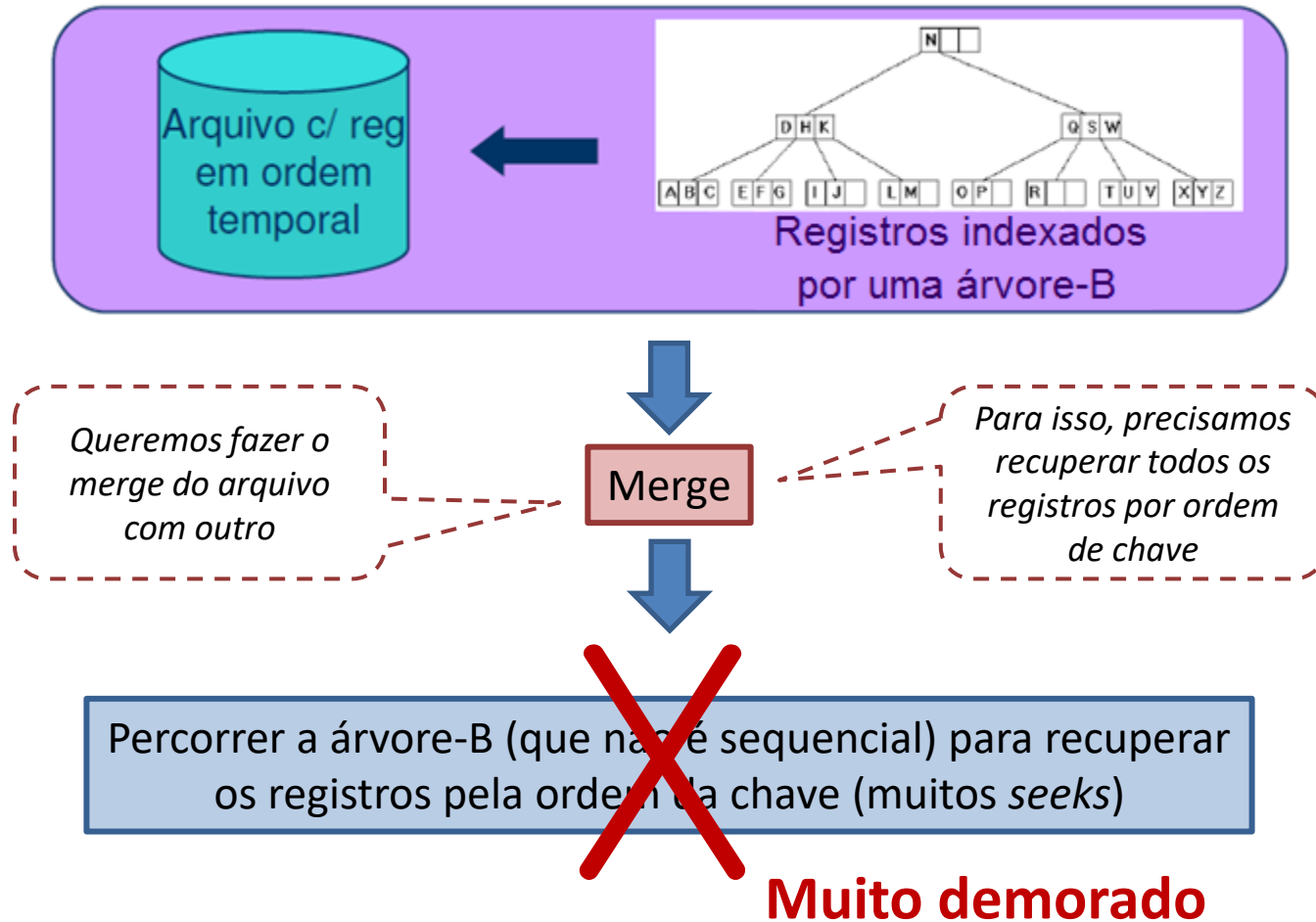
QUAL TIPO UTILIZAR?

■ Situação 1 → Arquivo indexado



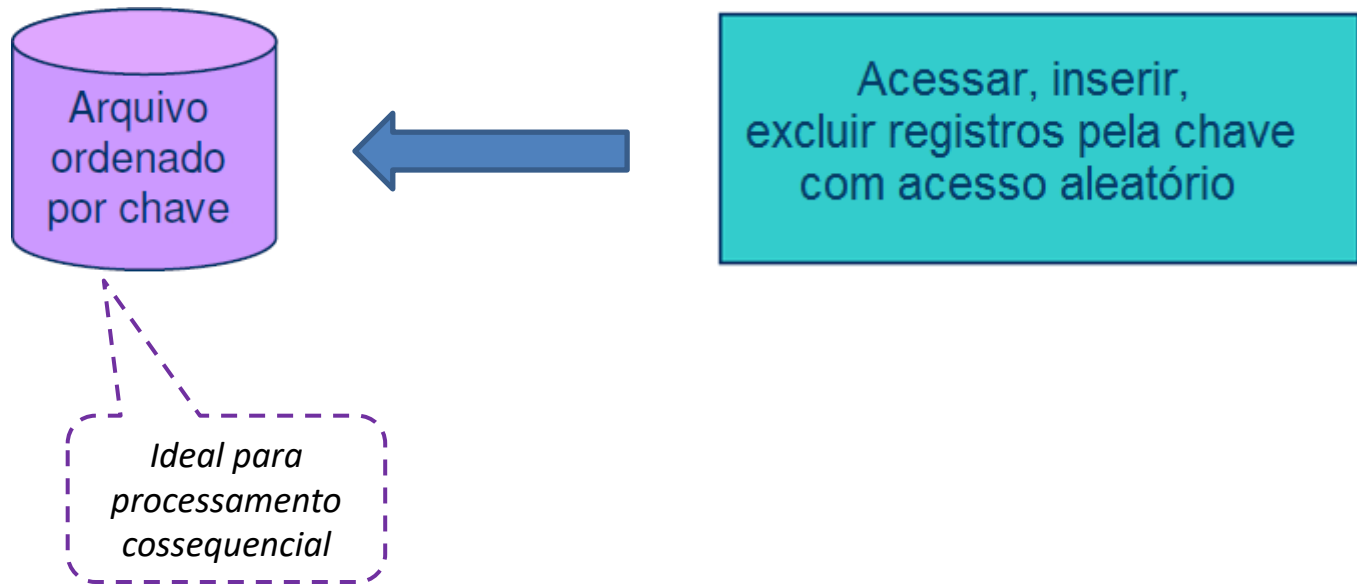
QUAL TIPO UTILIZAR?

■ Situação 1 → Arquivo indexado



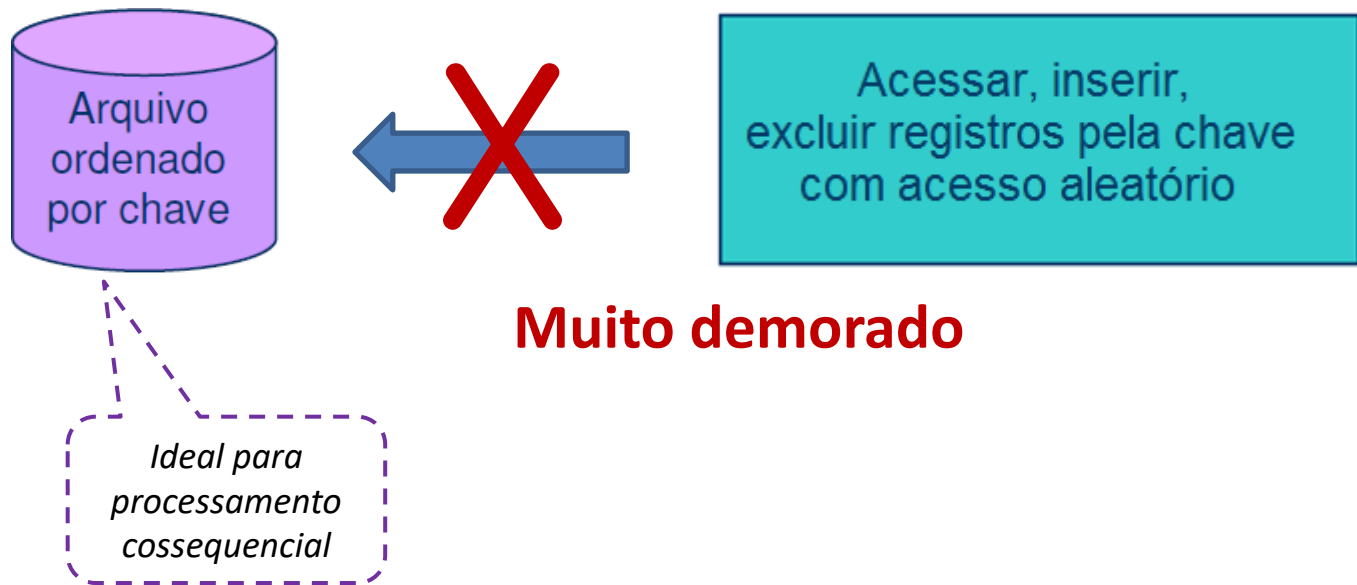
QUAL TIPO UTILIZAR?

■ Situação 2 → Arquivo sequencial ordenado



QUAL TIPO UTILIZAR?

- Situação 2 → Arquivo sequencial ordenado



ACESSO ALEATÓRIO + ACESSO SEQUENCIAL

- Muitas aplicações fazem tanto acesso aleatório quanto sequencial
- Por exemplo:
 - Controle acadêmico em uma universidade
 - Acesso via chave a registros individuais dos alunos
 - Grande quantidade de processamento em lote, por ex., quando os horários são publicados
 - Fatura de cartão de crédito
 - Acesso aleatório para verificar a situação de uma conta específica
 - Processamento em lote quando as faturas são geradas de acordo com as datas de vencimento

Um **arquivo sequencial indexado** busca atender essas necessidades

ACESSO SEQUENCIAL INDEXADO

- Um **arquivo sequencial indexado** contempla duas visões:
 - Arquivo **indexado** → Foco no acesso aleatório
 - Conjunto de registros indexados por uma chave
 - Dada uma chave, o registro pode ser acessado com um único *seek* (seu RRN/*byte-offset* é conhecido)
 - Arquivo **sequencial** → Foco no acesso sequencial ordenado
 - Conjunto de registros fisicamente ordenados por uma chave
 - Quando acessado sequencialmente, retorna os registros por ordem de chave, sem *seeks*

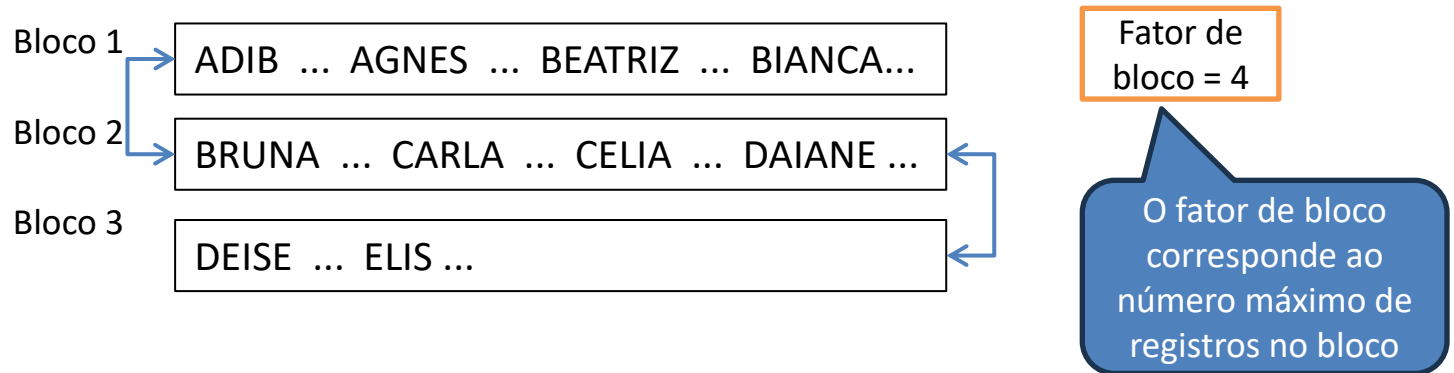
ACESSO SEQUENCIAL INDEXADO

- Como organizar um arquivo sequencial indexado?
 - 1º passo:
 - Criar um **arquivo sequencial em blocos**
 - Cada bloco de registros é fisicamente ordenado
 - Os blocos são duplamente encadeados de acordo com a ordem dos registros
 - Queremos o **acesso sequencial ordenado**, mas com um custo de manutenção baixo
 - 2º passo:
 - **Indexar** os blocos
 - Queremos o **acesso aleatório rápido**, então precisamos de um índice

ACESSO SEQUENCIAL POR BLOCOS

■ Arquivo sequencial em blocos

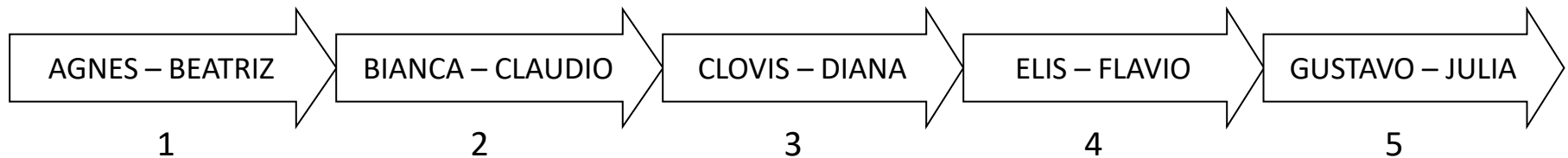
- Blocos ordenados e encadeados, mantendo o arquivo como um todo ordenado



As referências ligando os blocos são necessárias porque blocos logicamente consecutivos não são necessariamente fisicamente adjacentes

ACESSO SEQUENCIAL INDEXADO

- Cada bloco pode ser visto como uma distribuição de registros, com limites inferior e superior
 - Os limites dos blocos são suficientes para saber em qual bloco buscar determinada chave
 - Por ex., não vemos a chave “BRUNO” na representação abaixo, mas sabemos que devemos buscá-la no bloco 2

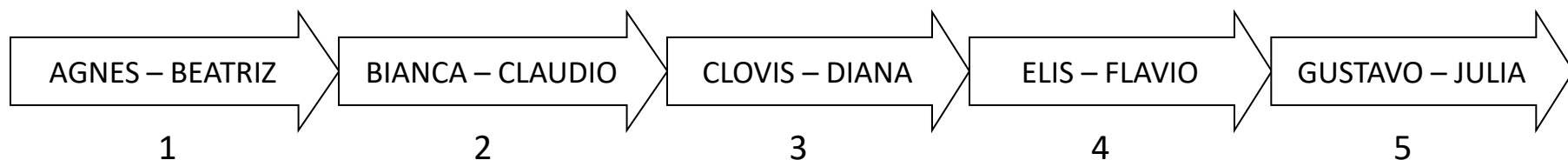


As chaves aparentes em cada bloco são a menor e a maior no bloco (i.e., os limites inferior e superior)

Como localizar o bloco a ser pesquisado sem uma busca sequencial?

ACESSO SEQUENCIAL INDEXADO

■ Indexação dos blocos



Índice de blocos

Chave	Nº do bloco
BIANCA	1
CLOVIS	2
ELIS	3
GUSTAVO	4
...	5

Para localizar o bloco a ser pesquisado, utilizamos o índice

Neste exemplo, as **chaves no índice** representam o **limite superior** de cada bloco

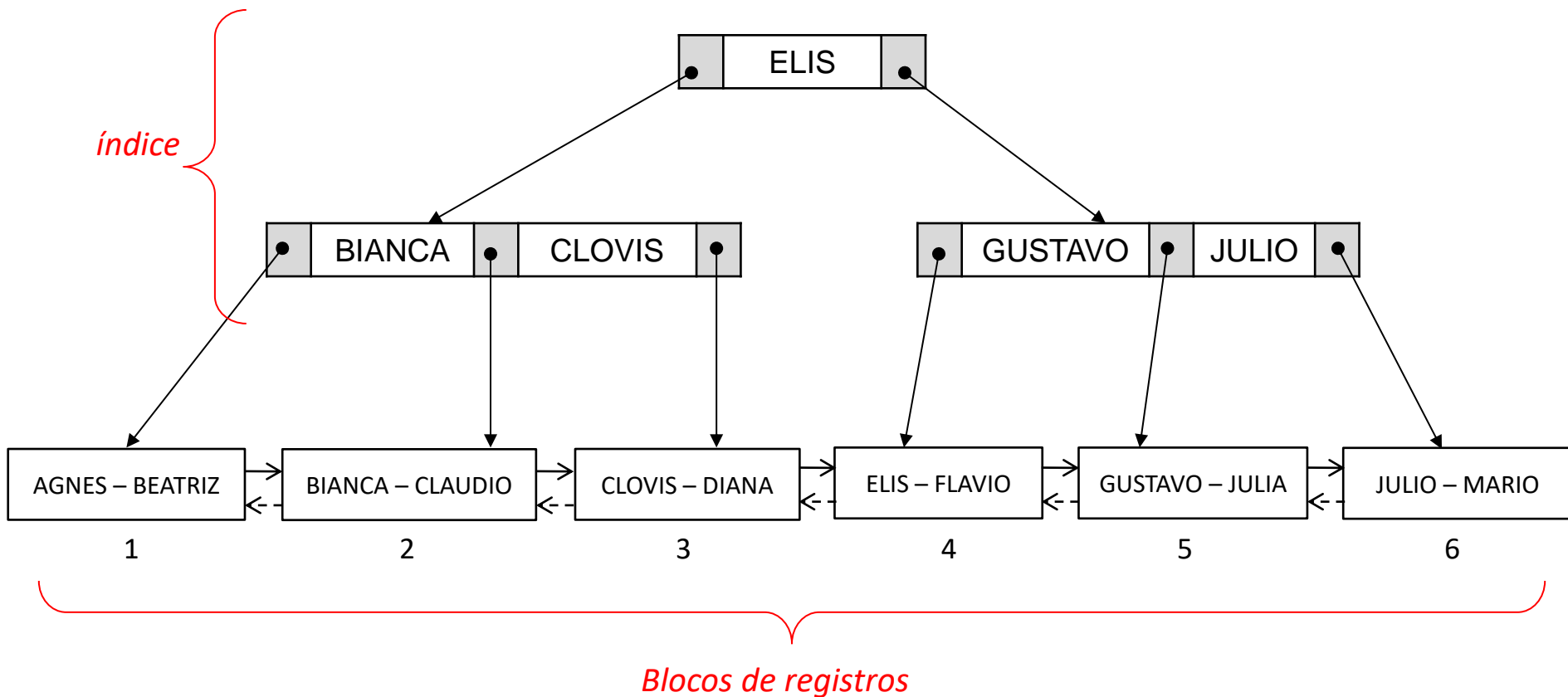
ACESSO SEQUENCIAL INDEXADO

- A combinação de um índice e um arquivo sequencial em blocos implementa o **acesso sequencial indexado**
 - Para o acesso indexado, busca-se no índice pelo bloco correto e, em seguida, busca-se no bloco o registro desejado
 - Para o acesso sequencial, faz-se a leitura dos blocos, a partir do primeiro e seguindo as referências dos próximos
- No caso de arquivos grandes, esses índices podem ser mantidos em disco utilizando uma **árvore-B⁺**
 - Uma **árvore-B⁺** utiliza uma árvore-B para indexar um arquivo sequencial em bloco
 - **As chaves nesse tipo de árvore estão associadas aos blocos de registros** e não a um registro particular

ÁRVORE-B⁺

- Em uma **árvore-B⁺**, as **folhas representam os blocos**, que são duplamente encadeados, e as **páginas internas representam o índice**
- O índice serve como um “mapa” para os blocos
 - Por isso, **as páginas internas da árvore-B⁺ contêm cópias das chaves** que servem para indicar a localização do bloco em que devemos buscar pela chave de interesse
- Note que, por essa definição, a estrutura das folhas (blocos) pode ser diferente da estrutura das páginas

ÁRVORE-B⁺



ÁRVORE-B⁺: MANUTENÇÃO

- As inserções e as remoções são similares às da árvore-B, mas com algumas diferenças

■ Inserção

- O *overflow* em uma folha é tratado com uma divisão, mas em vez de promover a chave mediana, **promovemos uma cópia** dela
 - Isso fará com que a ocupação mínima das folhas fique diferente da ocupação mínima das páginas internas
- No *overflow* de páginas internas, a divisão é igual à da árvore-B

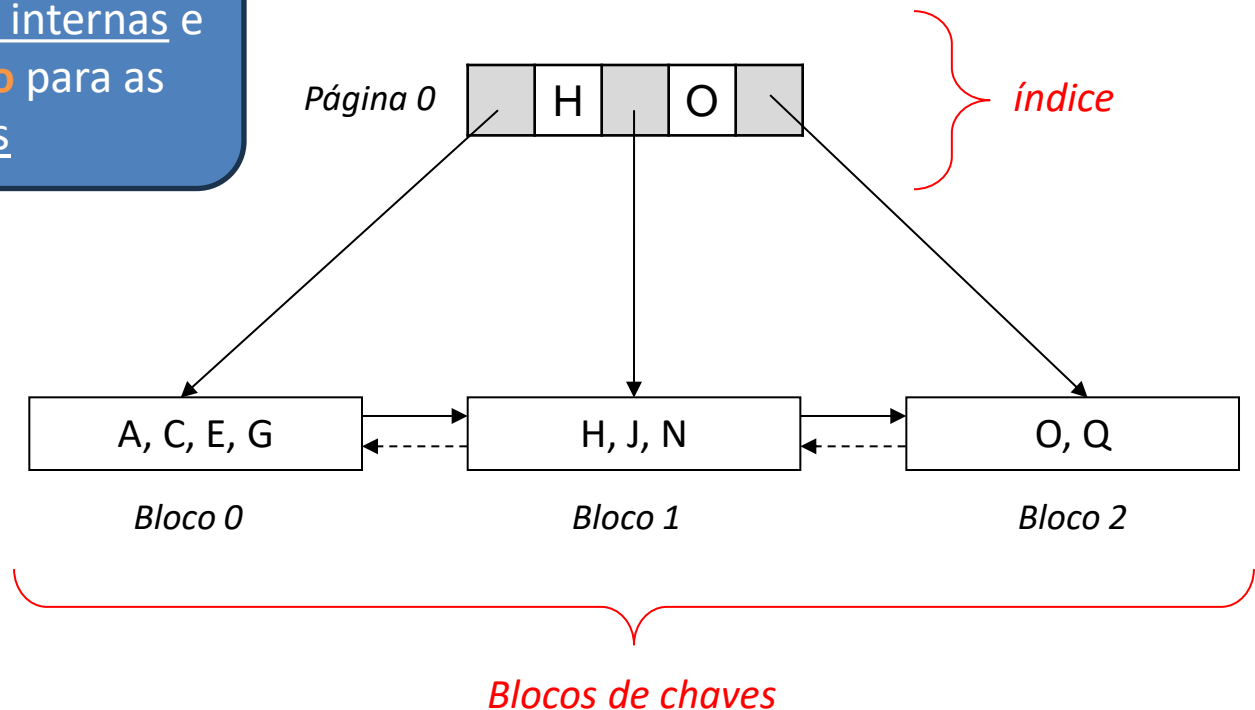
■ Remoção

- Similarmente à árvore-B, no *underflow*, a redistribuição será preferida à concatenação
 - O cálculo da **ocupação mínima será diferente para folhas e páginas internas**, uma vez que não há mais promoção de chaves das folhas

ÁRVORE-B⁺: MANUTENÇÃO

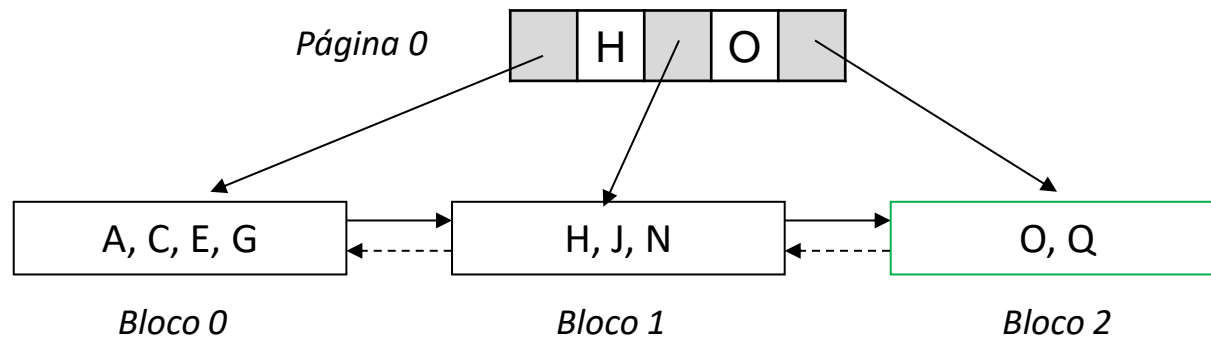
- Folhas \rightarrow fator de bloco 4 \rightarrow **mínimo** = $\lceil fb/2 \rceil = 2$
- Páginas internas \rightarrow ordem 3 \rightarrow **mínimo** = $\lceil m/2 \rceil - 1 = 1$

Usaremos o termo **página** para as páginas internas e o termo **bloco** para as folhas

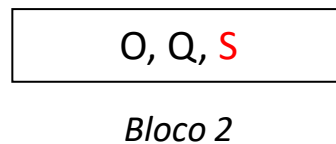


ÁRVORE-B⁺: MANUTENÇÃO

■ Inserir “S”



- Comece pela raiz, vá para a direita de “O” e insira “S” no bloco 2

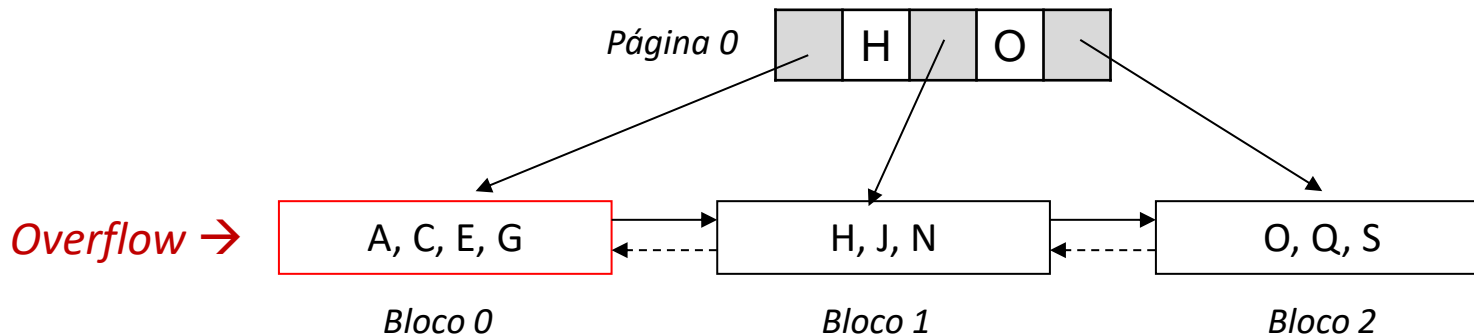


Não há modificação no índice

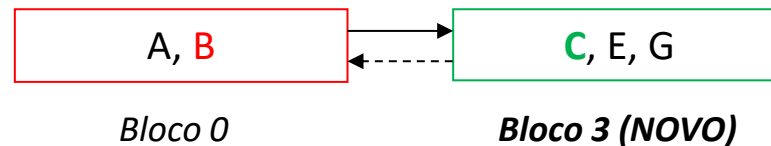
ÁRVORE-B⁺: MANUTENÇÃO

■ Inserir “B”

- Comece pela raiz, vá para a esquerda de “H” e insira “B” no bloco 0

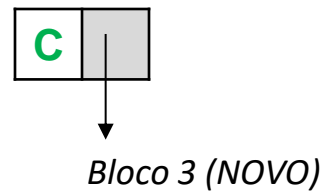


- O bloco 0 é dividido e uma **cópia do separador** é promovido

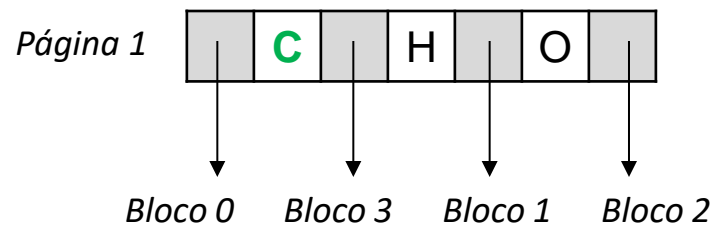


ÁRVORE-B⁺: MANUTENÇÃO

- Promova o novo separador “C” juntamente com o ponteiro para o novo bloco 3



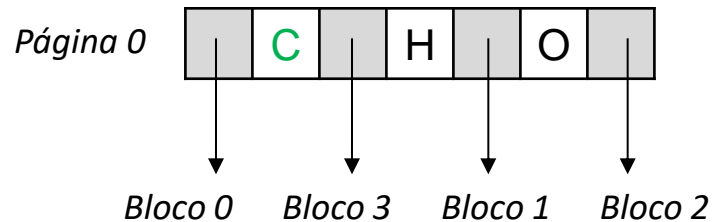
- Idealmente, teríamos:



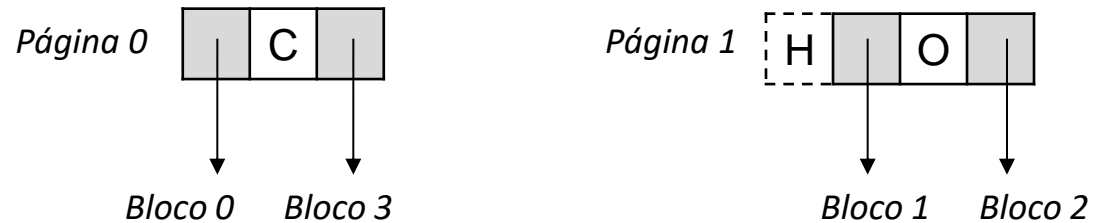
- Mas a ordem do índice é 3 (3 ponteiros, 2 chaves) → *overflow*
 - A página do índice precisa ser dividida

ÁRVORE-B⁺: MANUTENÇÃO

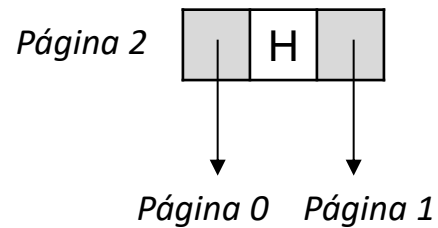
Ideal:



- Divisão da página

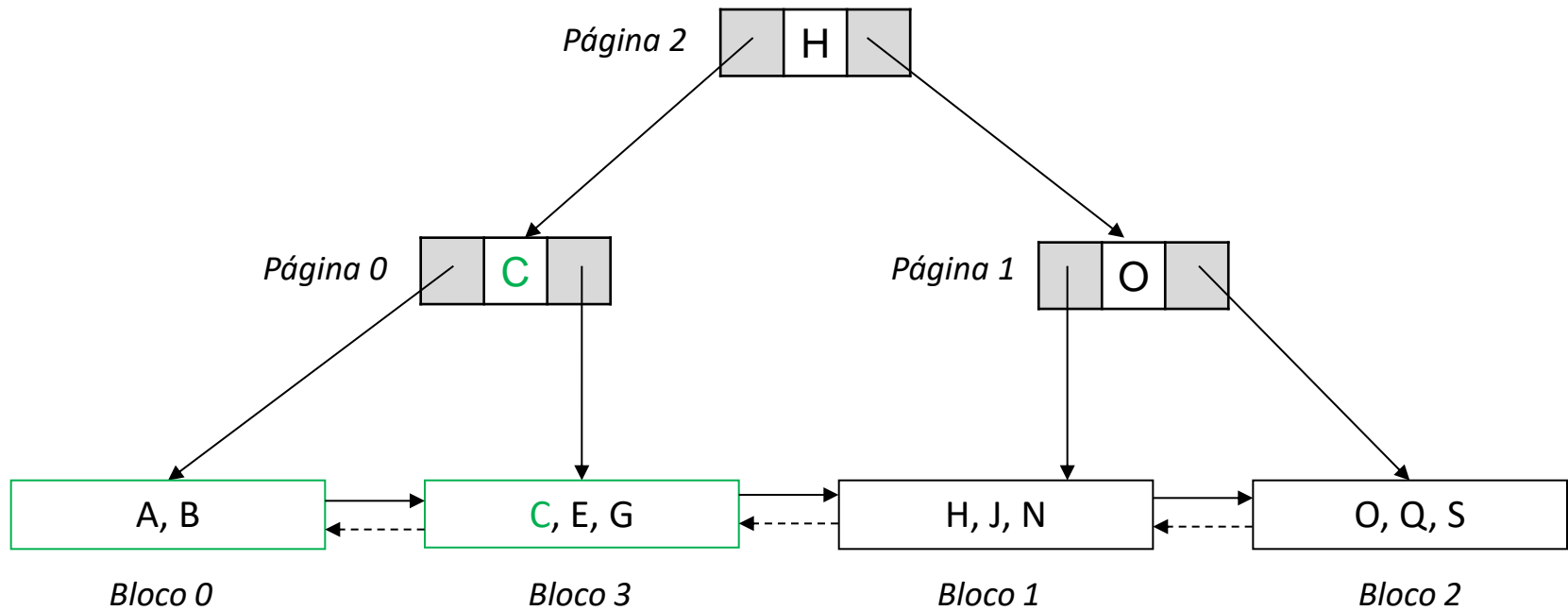


- Crie uma nova raiz para apontar para as páginas 0 e 1



ÁRVORE-B⁺: MANUTENÇÃO

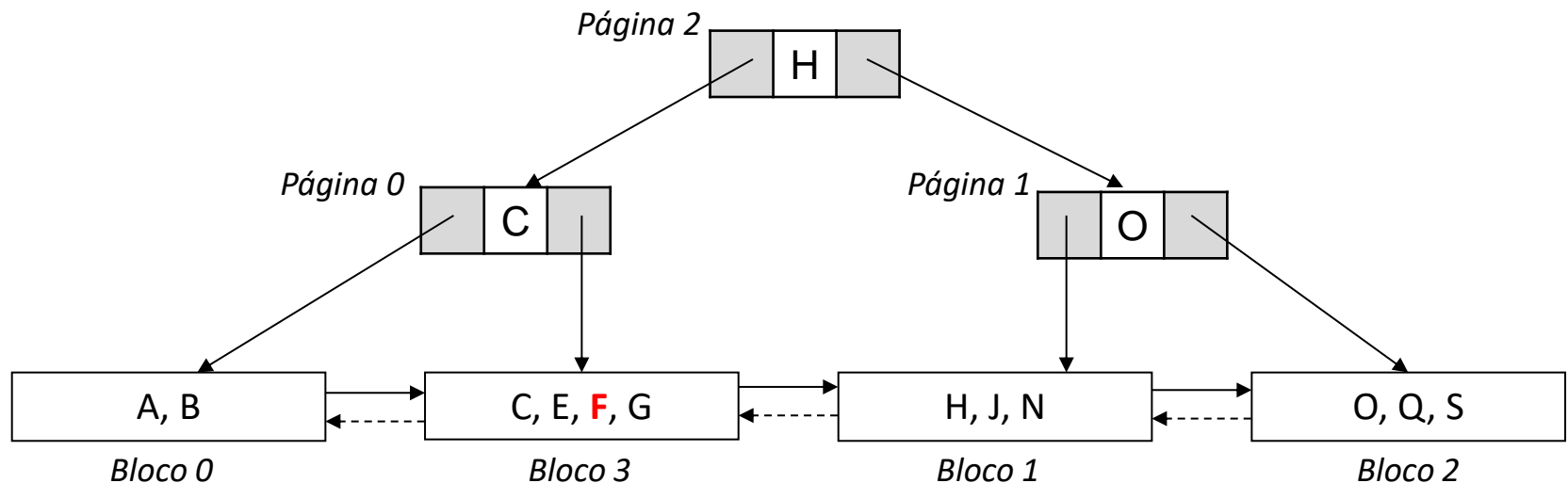
■ Nova árvore-B⁺



ÁRVORE-B⁺: MANUTENÇÃO

■ Inserir “F”

- Comece pela raiz, vá para a esquerda de “H”, depois para a direita de “C” e insira “F” no bloco 3

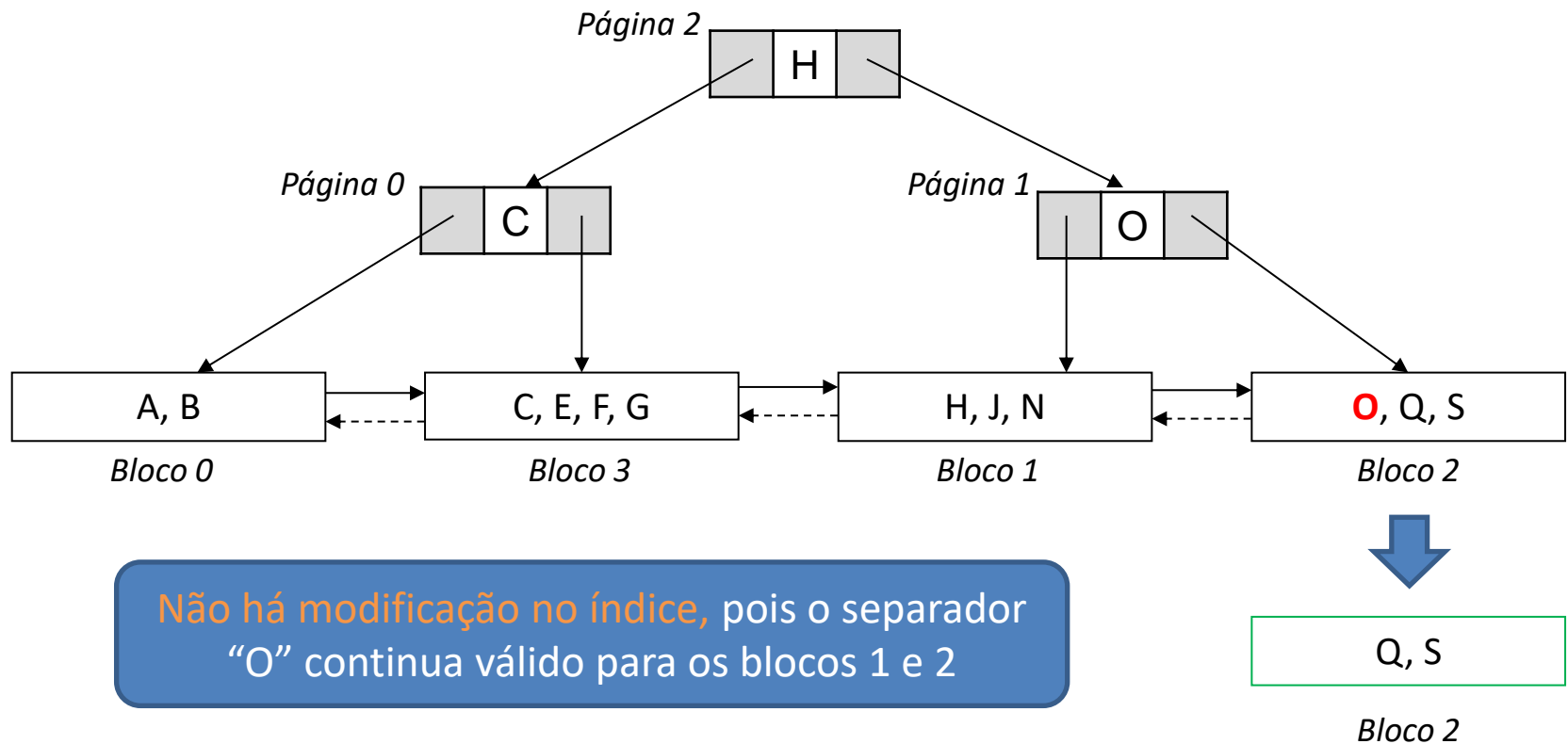


Não há modificação no índice

ÁRVORE-B⁺: MANUTENÇÃO

■ Remover “O”

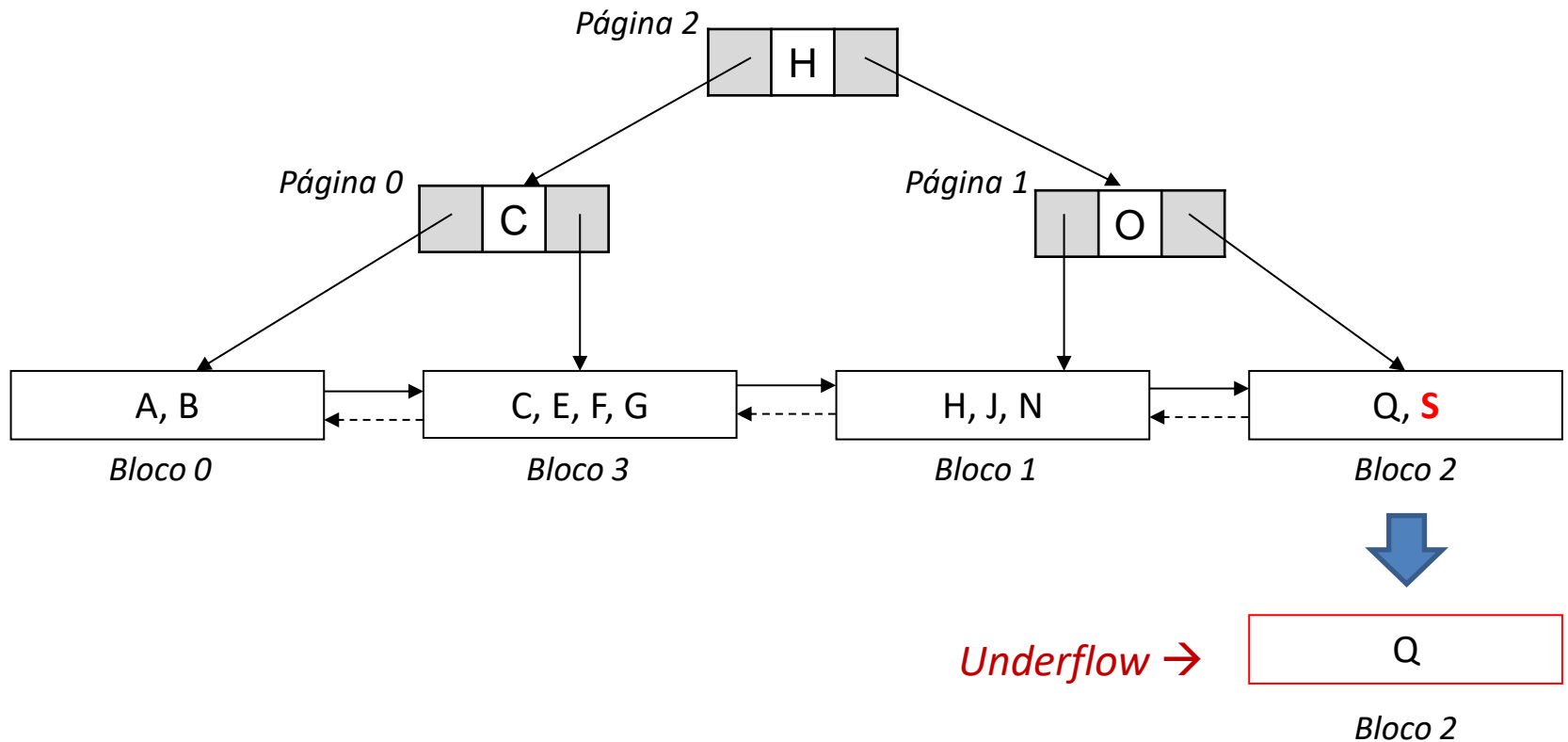
- Comece pela raiz, vá para a direita de “H”, depois para a direita de “O” e remova “O” do bloco 2



ÁRVORE-B⁺: MANUTENÇÃO

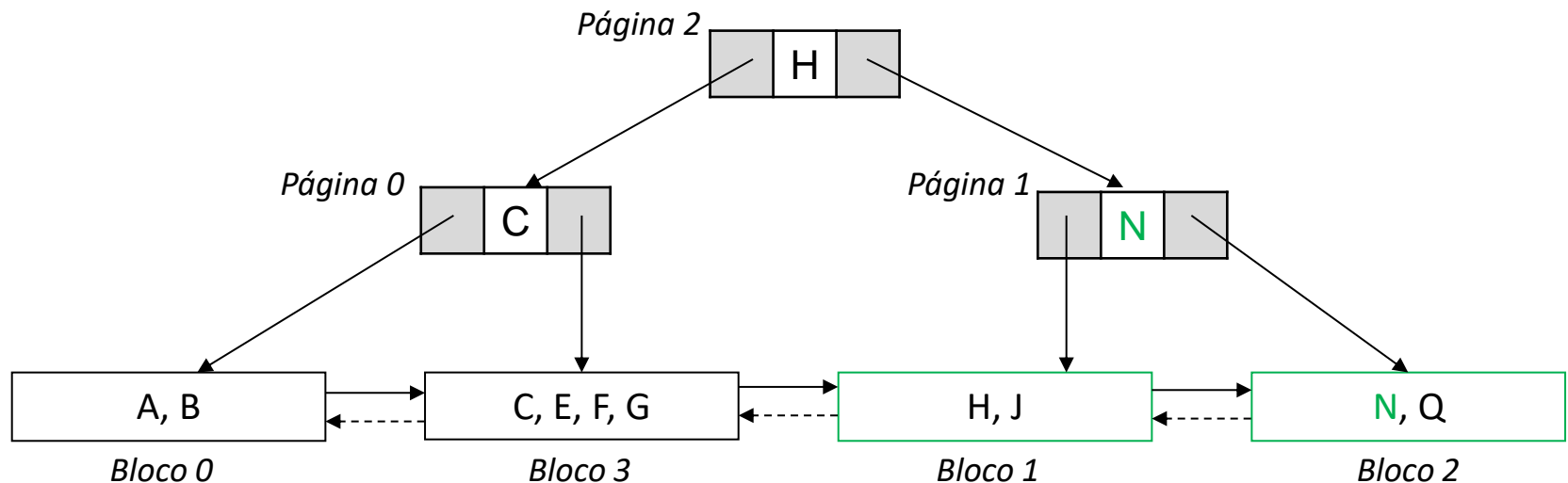
■ Remover “S”

- *Underflow* no bloco 2



ÁRVORE-B⁺: MANUTENÇÃO

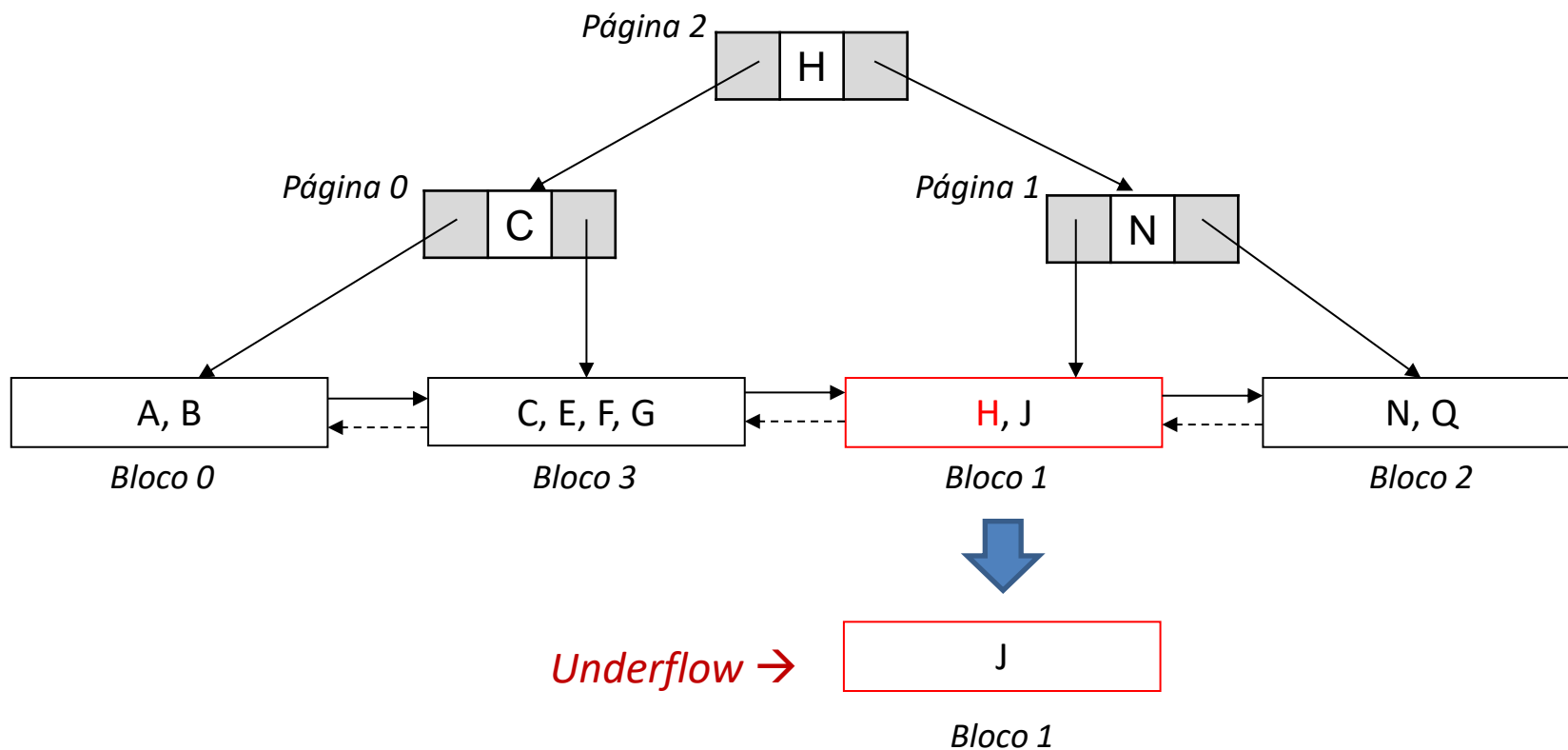
- Uma vez que o bloco 3 está cheio, fazemos redistribuição
 - Devido à redistribuição, atualizamos o separador “O” para “N”



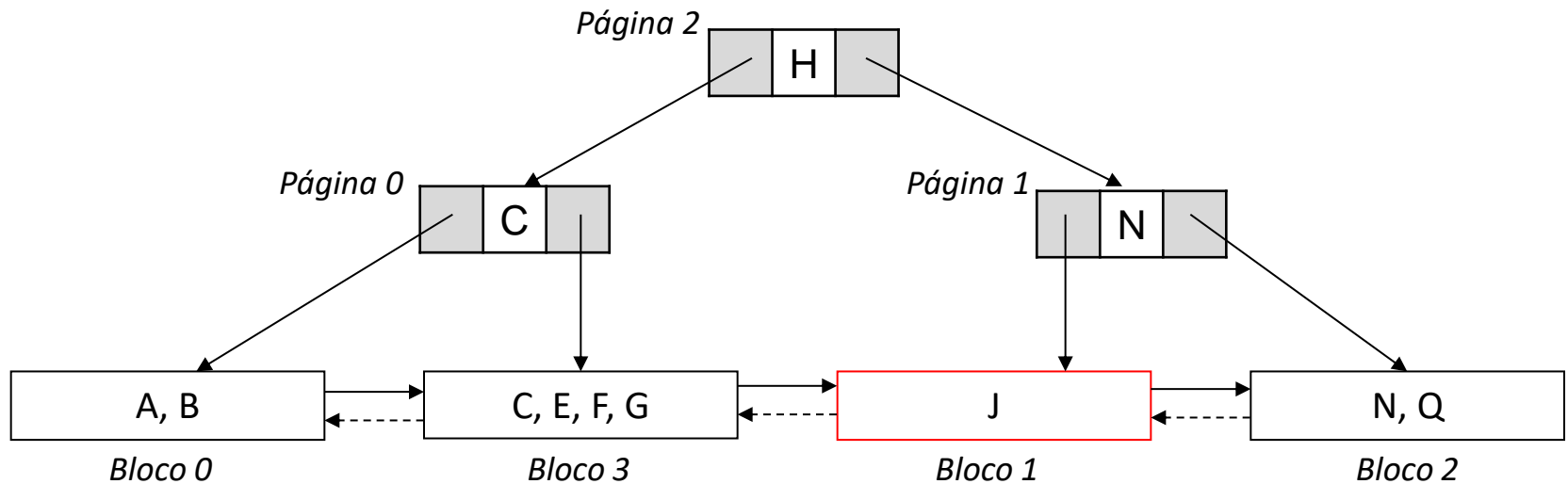
ÁRVORE-B⁺: MANUTENÇÃO

■ Remover “H”

- *Underflow* no bloco 1



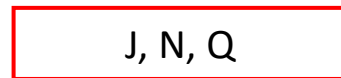
ÁRVORE-B⁺: MANUTENÇÃO



- Não podemos redistribuir entre os blocos 3 e 1, pois **eles não são filhos do mesmo pai**
- Uma vez que o bloco 2 está no limite mínimo, fazemos concatenação

ÁRVORE-B⁺: MANUTENÇÃO

- Concatenamos os blocos 1 e 2
- O bloco 2 é enviado para a PED



Bloco 1



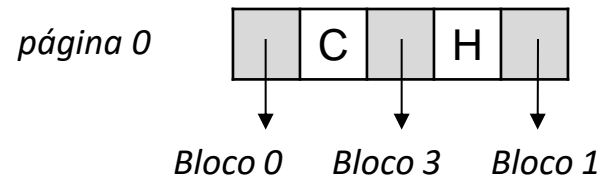
Bloco 2

- É necessário remover o separador “N” e o ponteiro para o bloco 2 da página 1, o que causa underflow

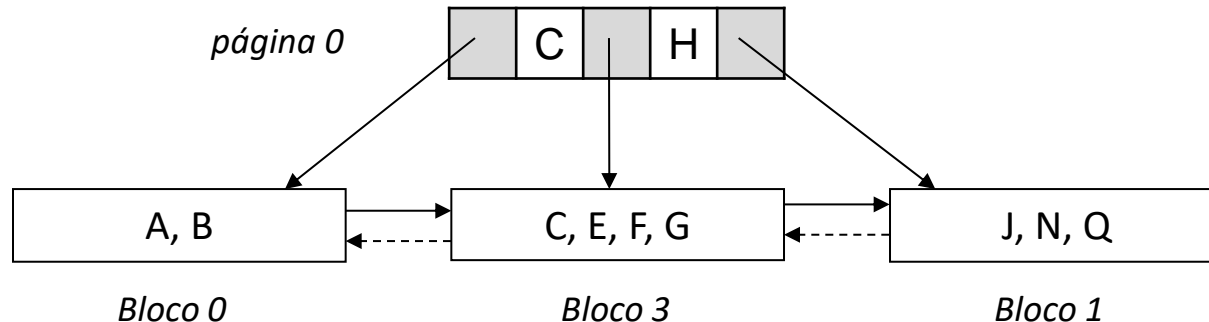


ÁRVORE-B⁺: MANUTENÇÃO

- Concatenamos a página 0 com a página 1



- O “H” desce da raiz, o que causa *underflow* na raiz
- A raiz é removida e a página 0 volta a ser a nova raiz



Árvore-B⁺ final

OBSERVAÇÕES FINAIS

- Geralmente, a árvore- B^+ fica armazenada em um arquivo único
 - É comum que os blocos tenham a mesma estrutura das páginas internas, pois isso facilita a manutenção do arquivo
 - Entretanto, nada impede que os blocos sequenciais (folhas) fiquem em um arquivo e o índice (páginas internas) em outro
- Árvores- B , B^* e B^+ são estruturas de arquivos poderosas e flexíveis, mas é um erro achar que elas são a solução para todos os problemas
 - Por exemplo, índices lineares mantidos em RAM são mais eficientes e devem ser utilizados para arquivos pequenos

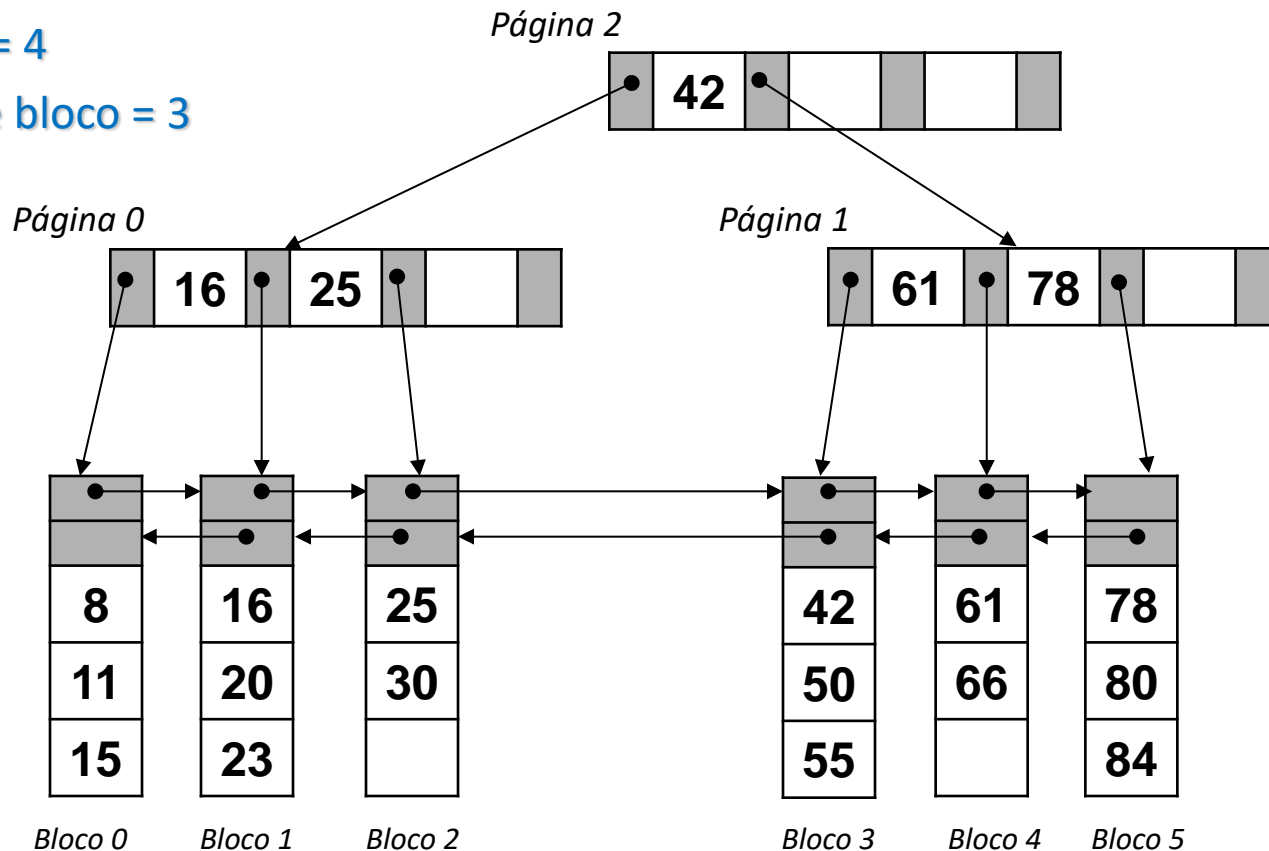
EXERCÍCIO 1

Em caso de
overflow, divida!

- Dada a árvore-B⁺ abaixo, indique o seu estado após cada uma das seguintes operações de inserção, nesta ordem: **89, 65, 53, 54, 27, 7, 35**

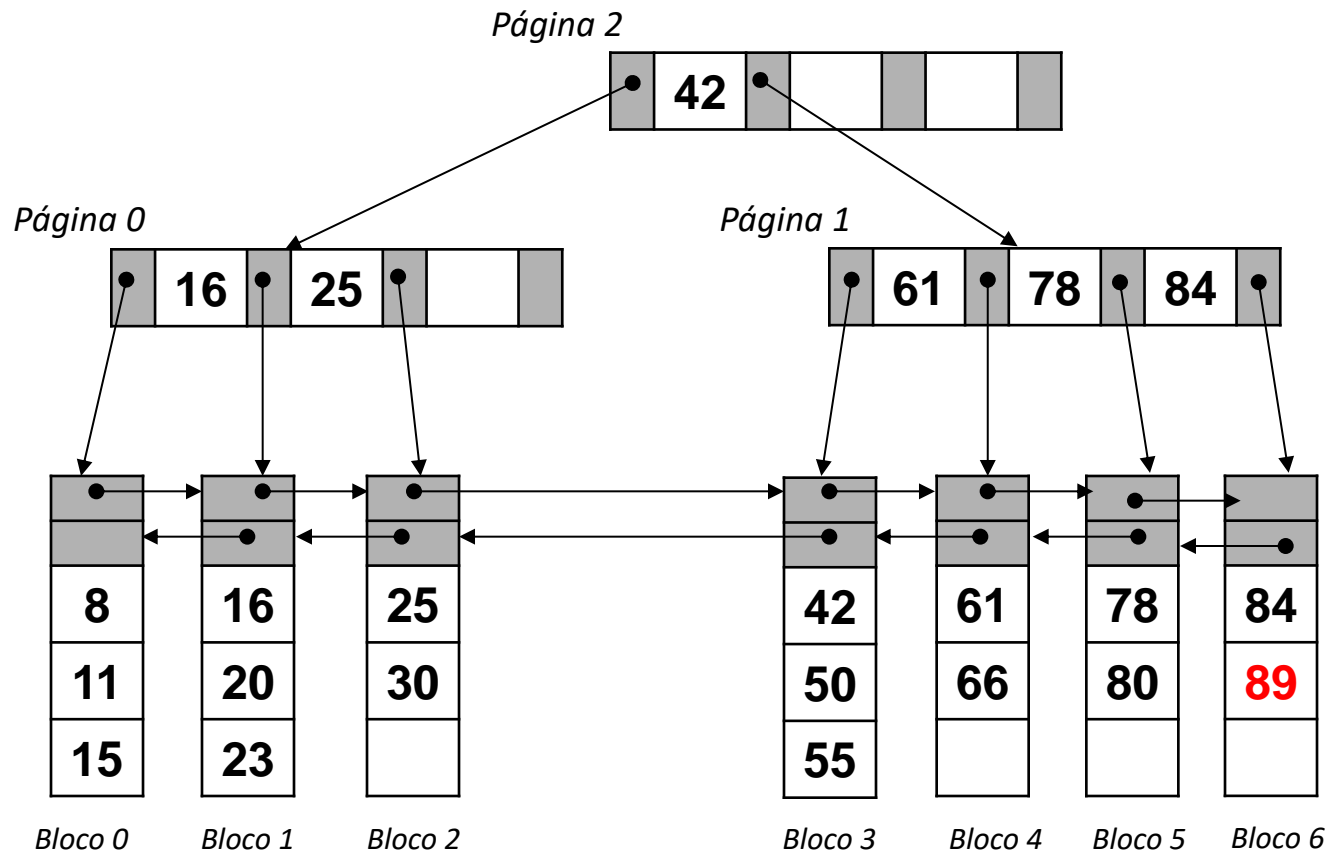
Ordem = 4

Fator de bloco = 3



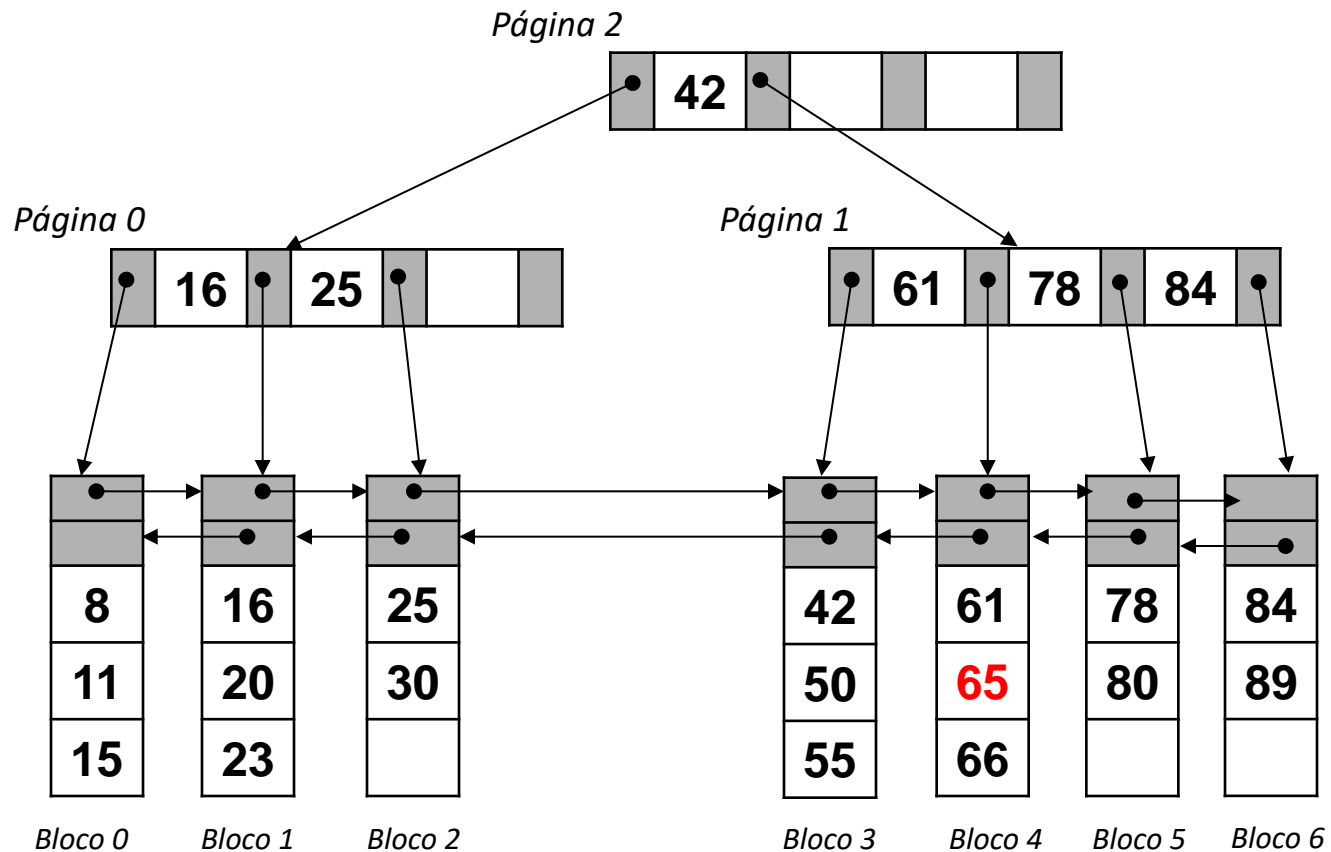
EXERCÍCIO 1

■ 89, 65, 53, 54, 27, 7, 35



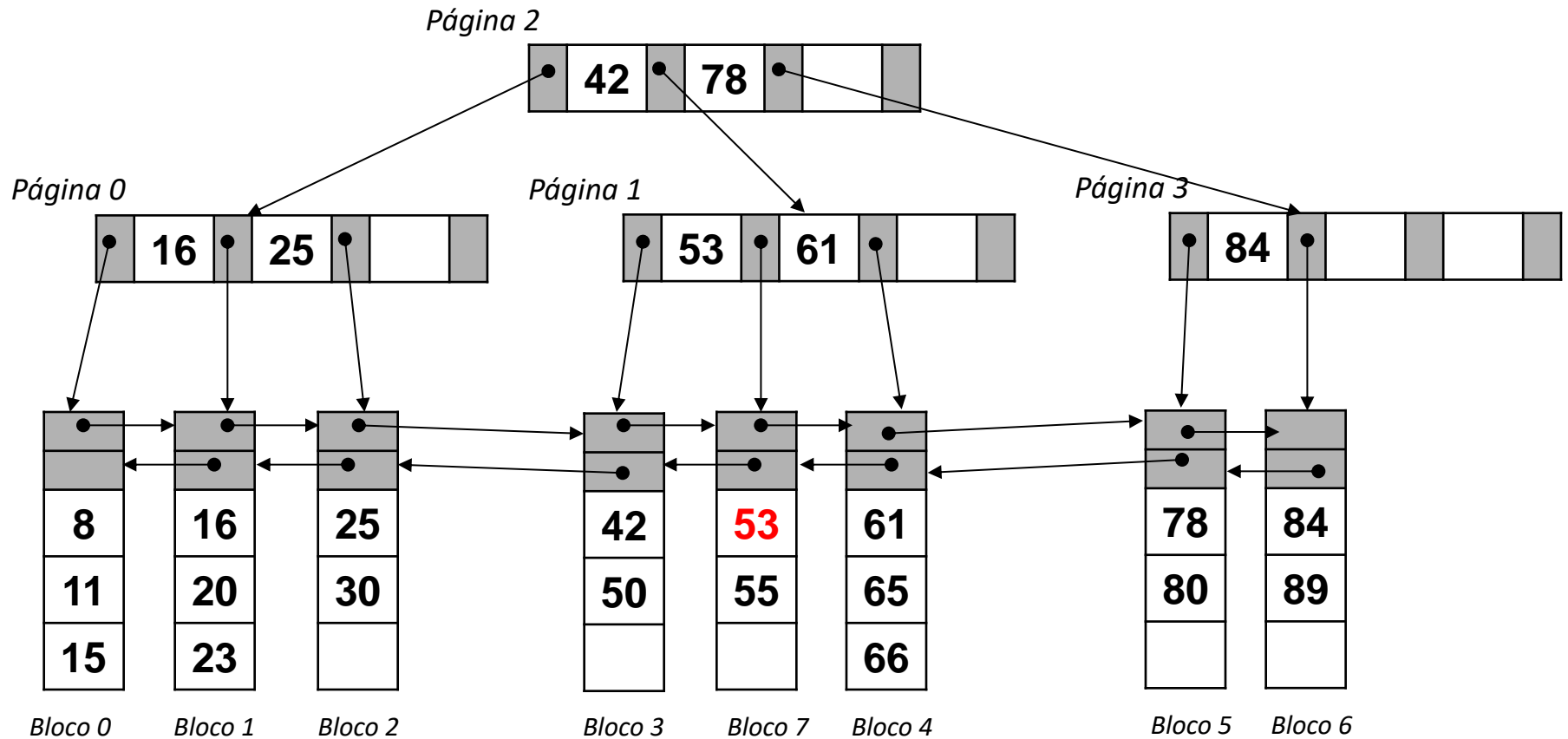
EXERCÍCIO 1

■ 65, 53, 54, 27, 7, 35



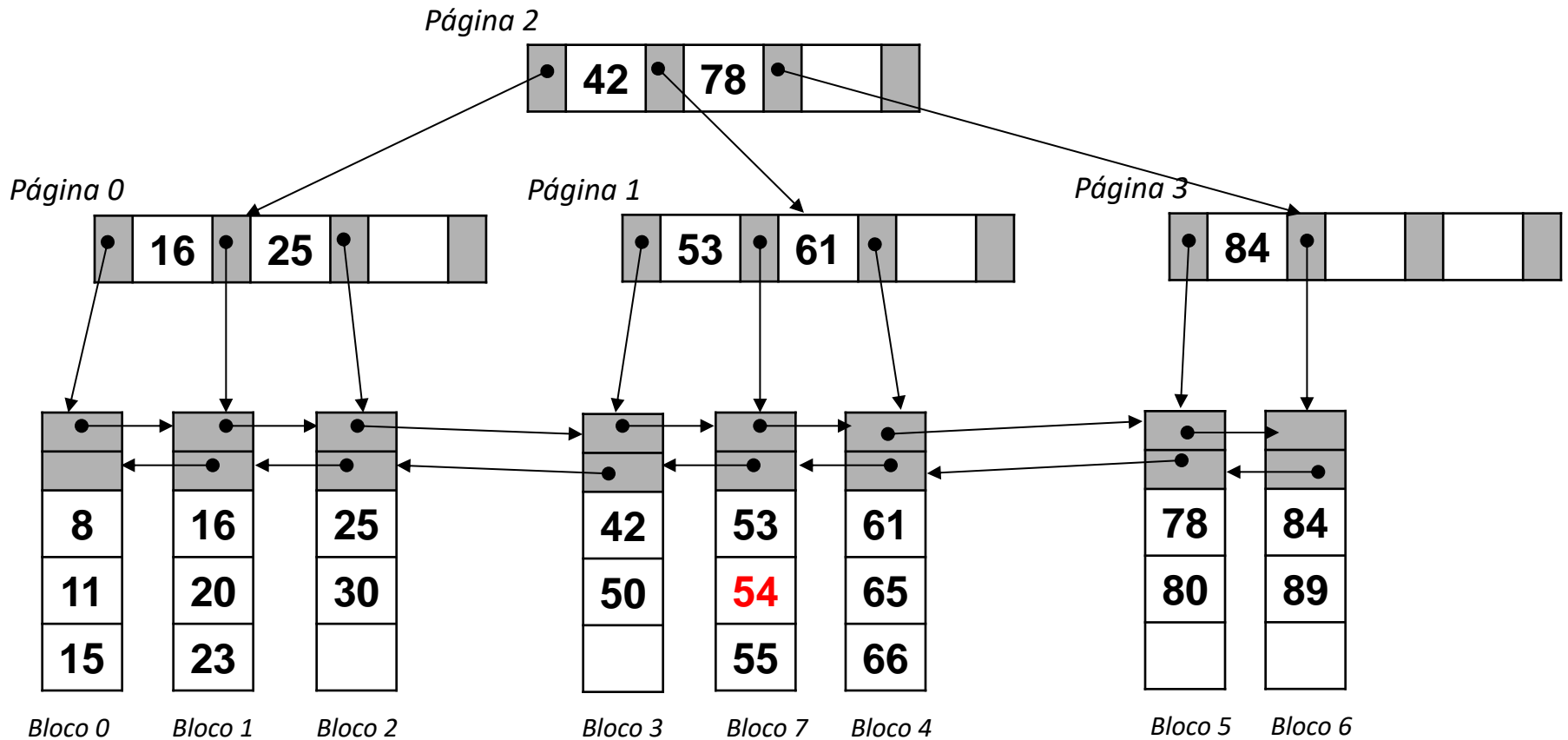
EXERCÍCIO 1

■ **53**, 54, 27, 7, 35



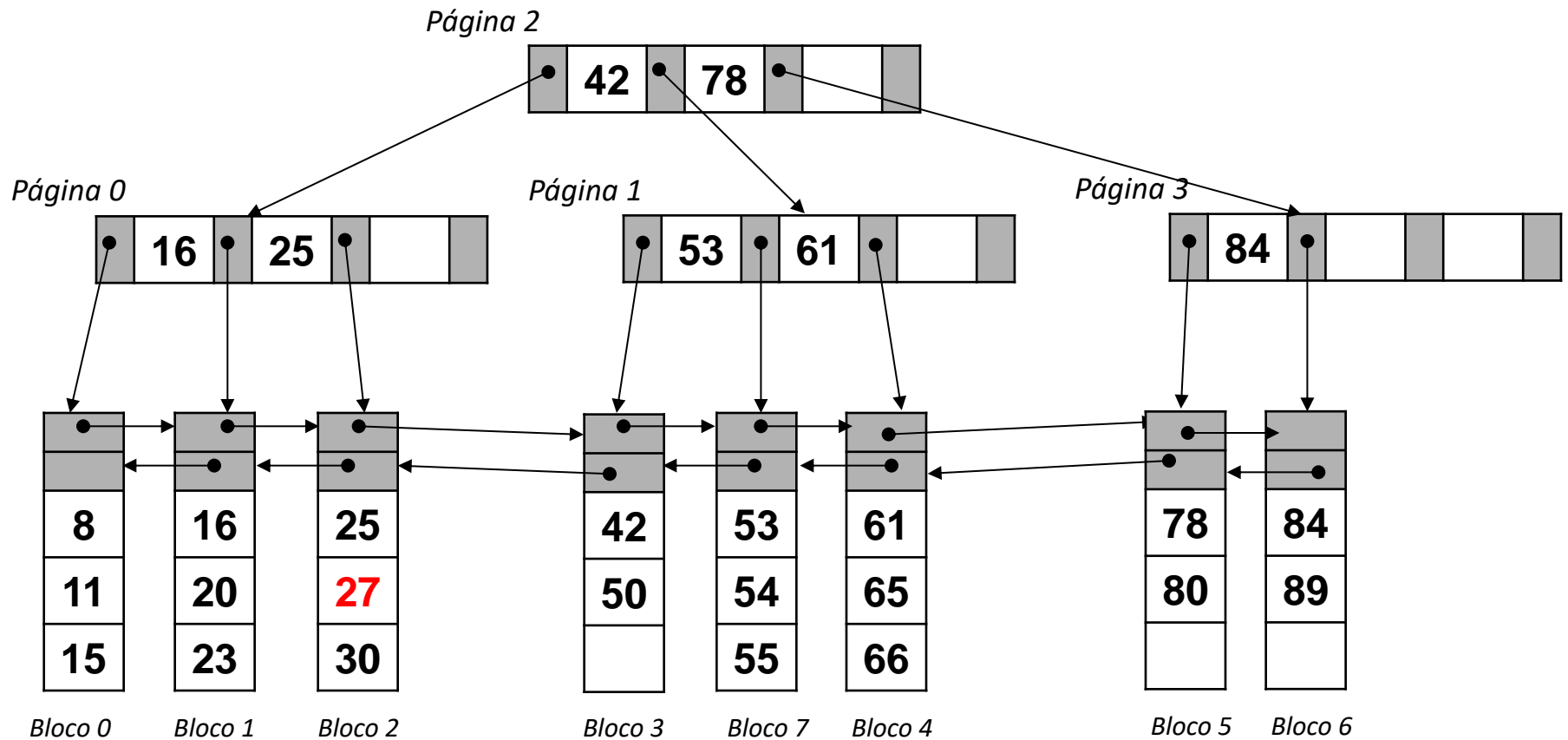
EXERCÍCIO 1

■ 54, 27, 7, 35



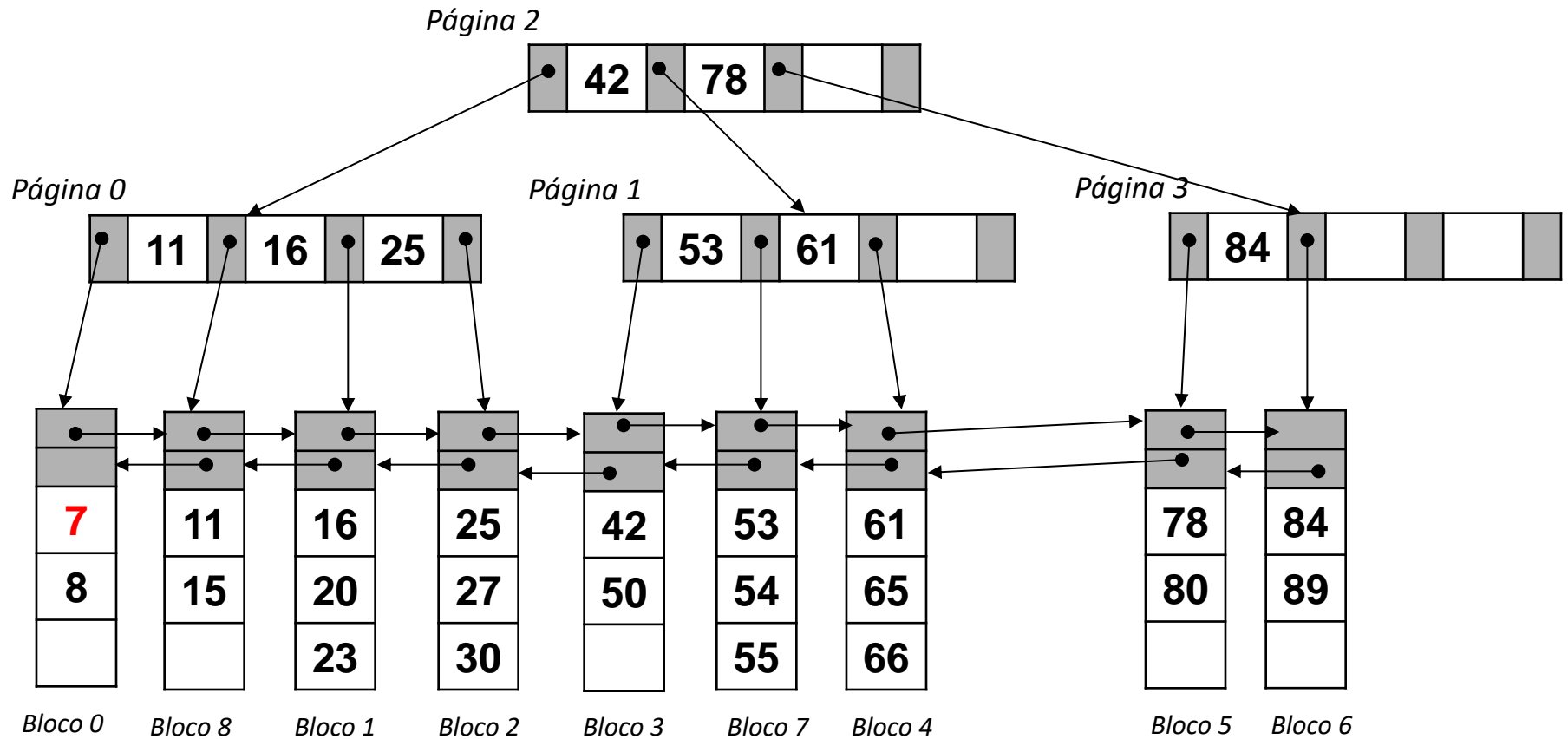
EXERCÍCIO 1

■ 27, 7, 35



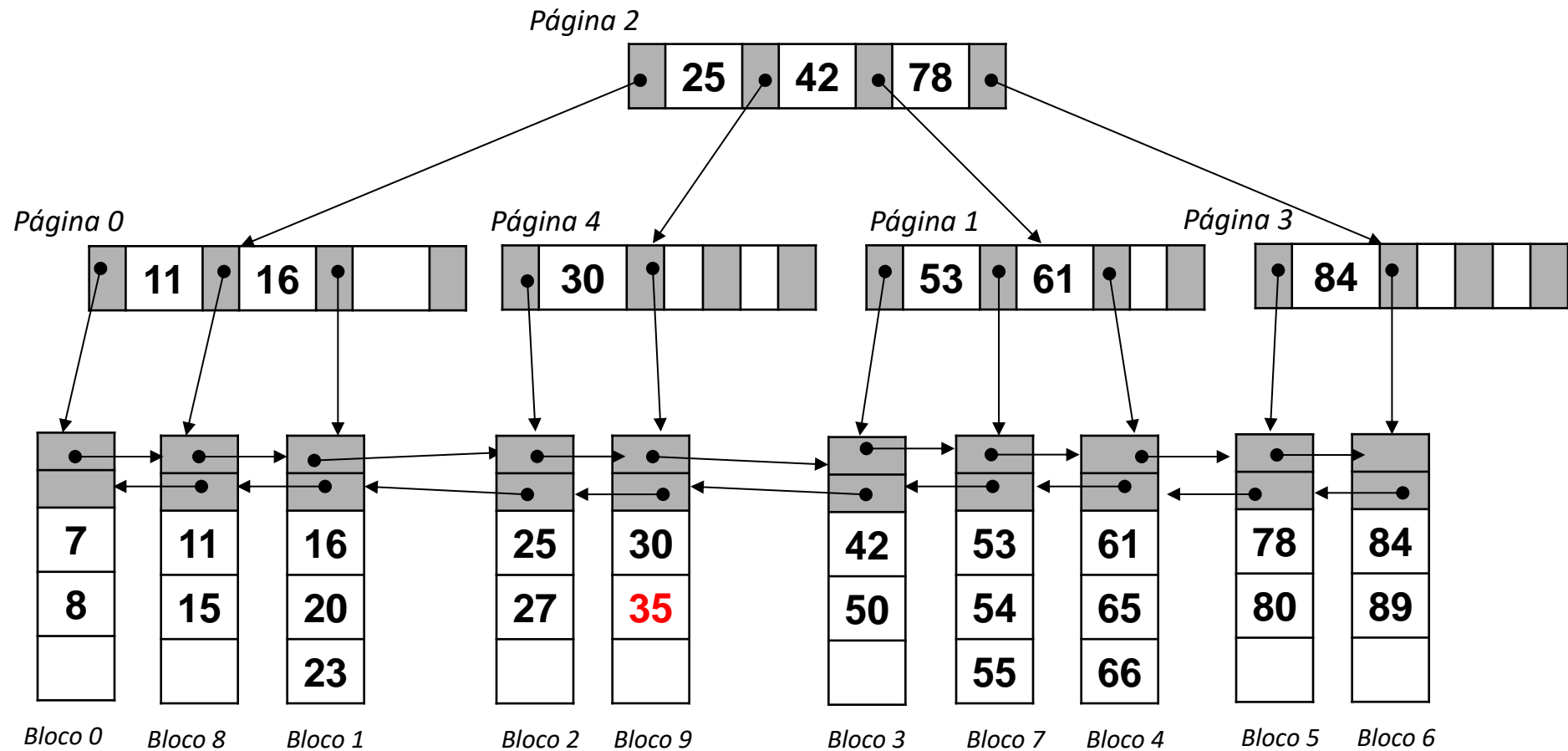
EXERCÍCIO 1

■ 7, 35



EXERCÍCIO 1

■ 35



Ocupação mínima:

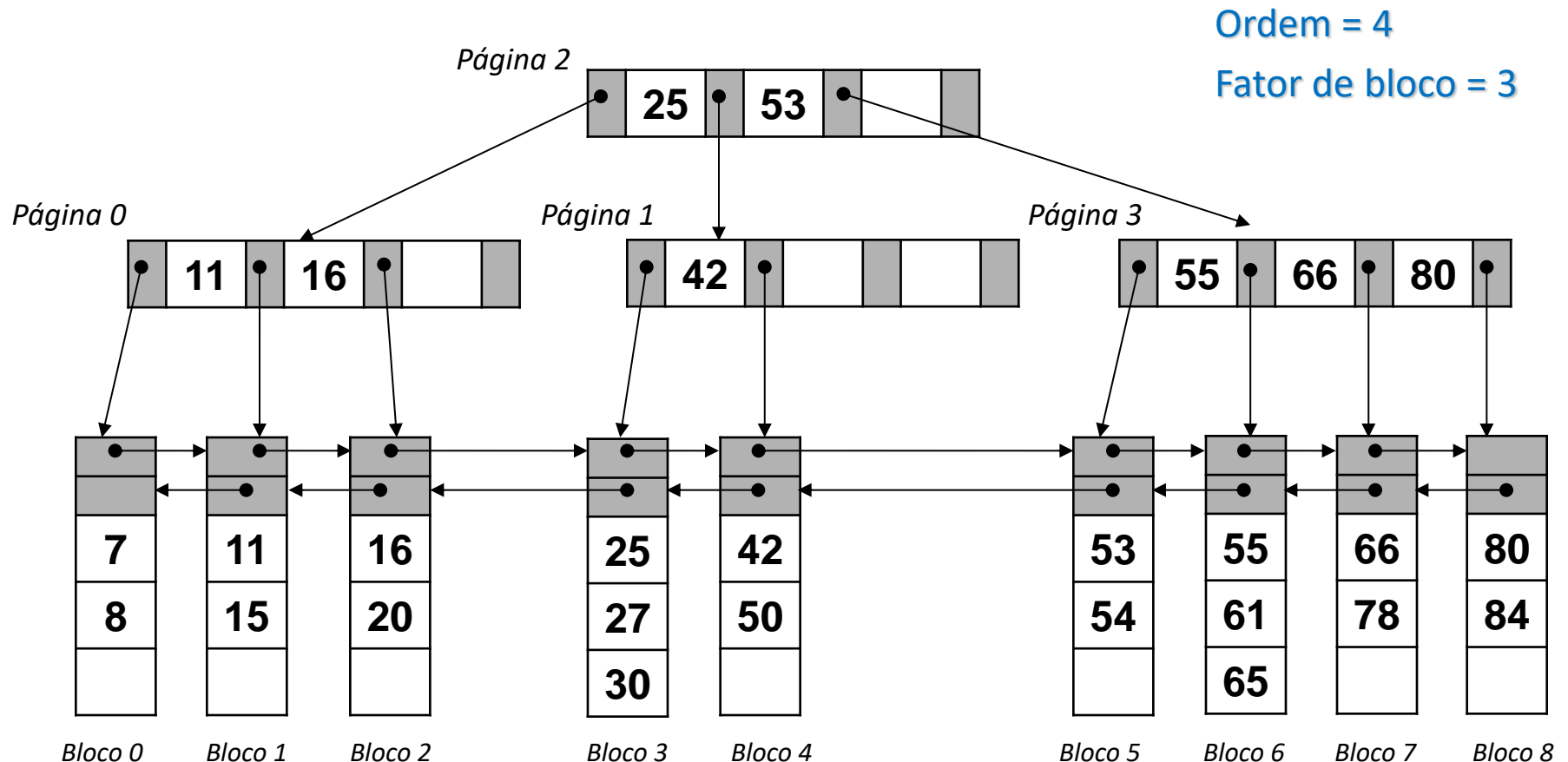
$$\text{Bloco} = \lceil \text{FB}/2 \rceil$$

$$\text{Página} = \lceil m/2 \rceil - 1$$

EXERCÍCIO 2

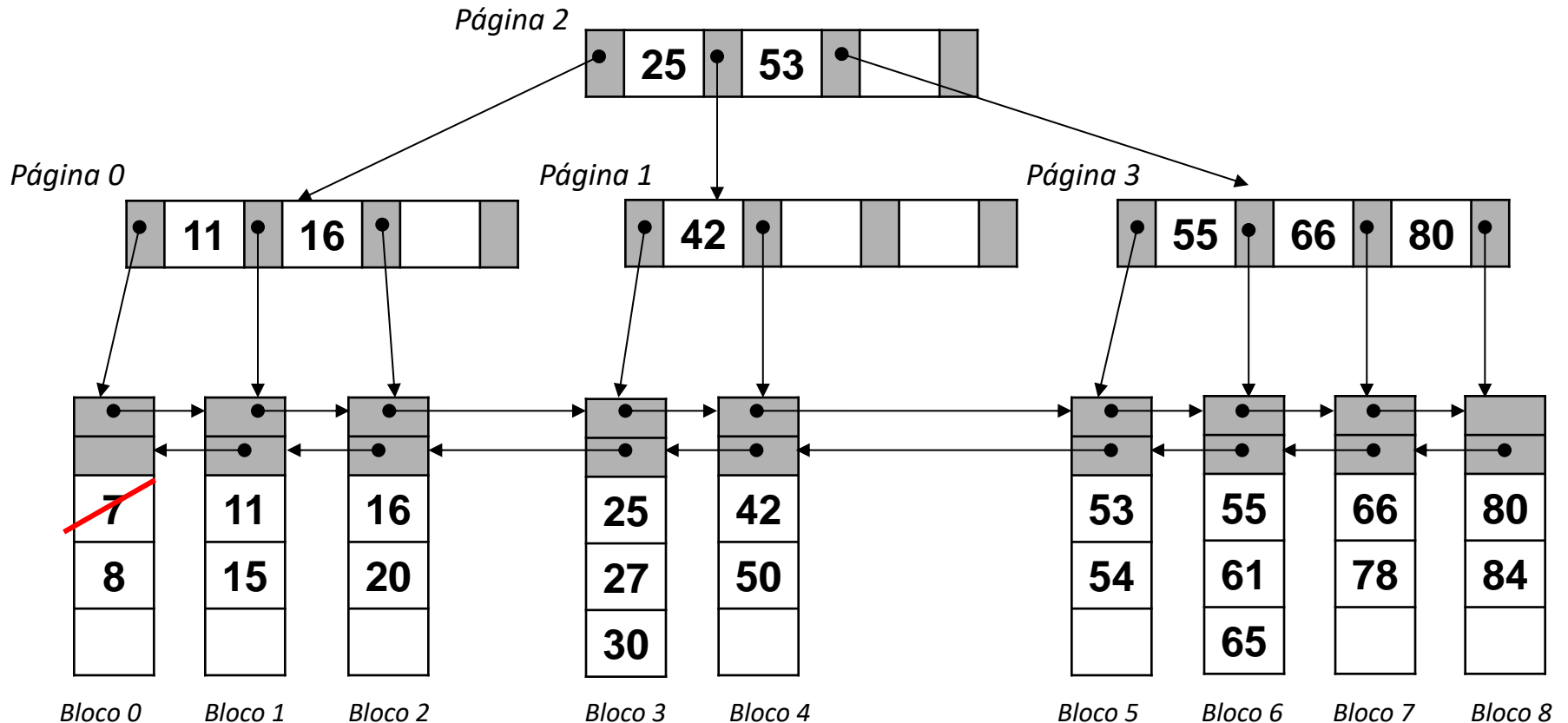
Em caso de *underflow*,
redistribua ou concatene!

- Dada a árvore-B⁺ abaixo, indique o seu estado após cada uma das seguintes operações de **remoção**, nesta ordem: **7, 27, 54, 53, 65, 84, 42, 55, 50, 20, 16, 80**



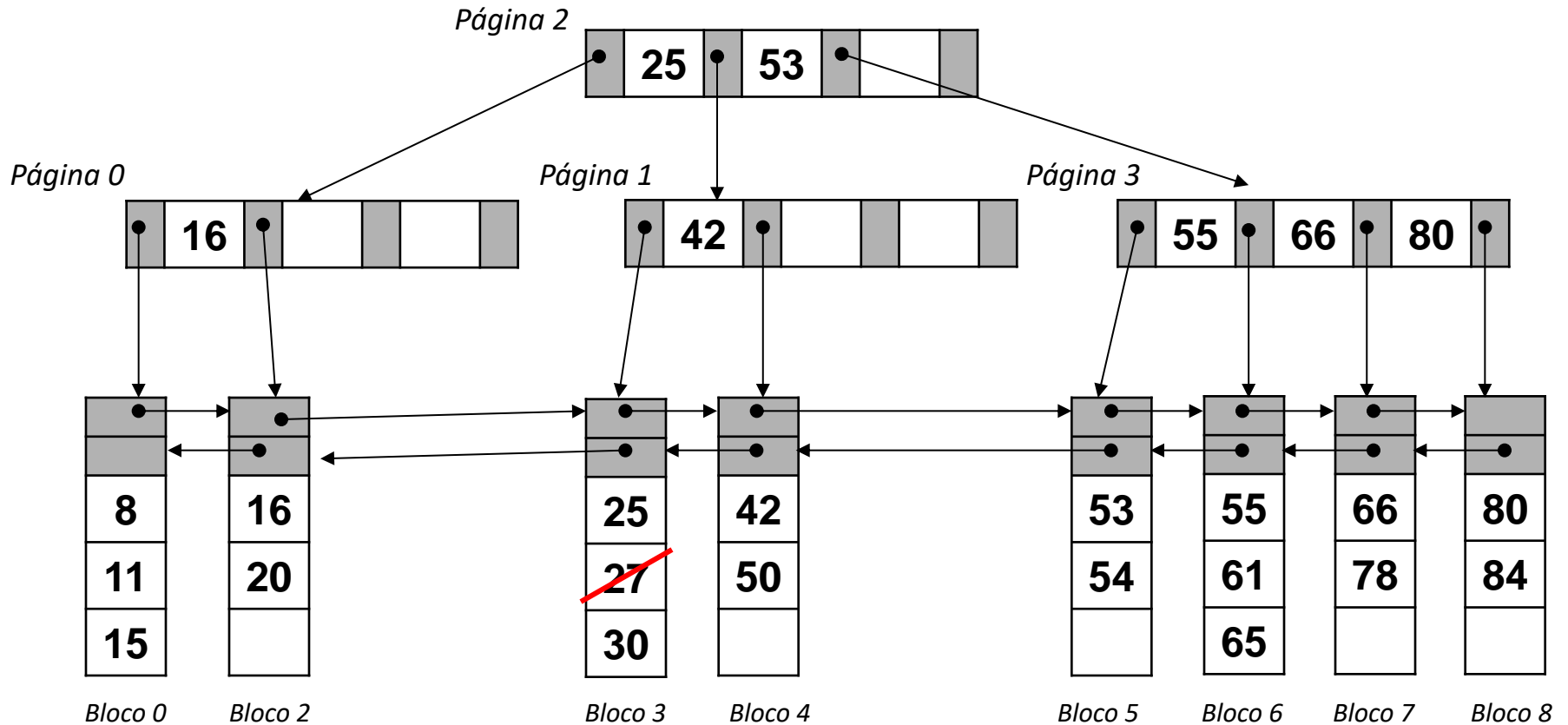
EXERCÍCIO 2

■ 7, 27, 54, 53, 65, 84, 42, 55, 50, 20, 16, 80



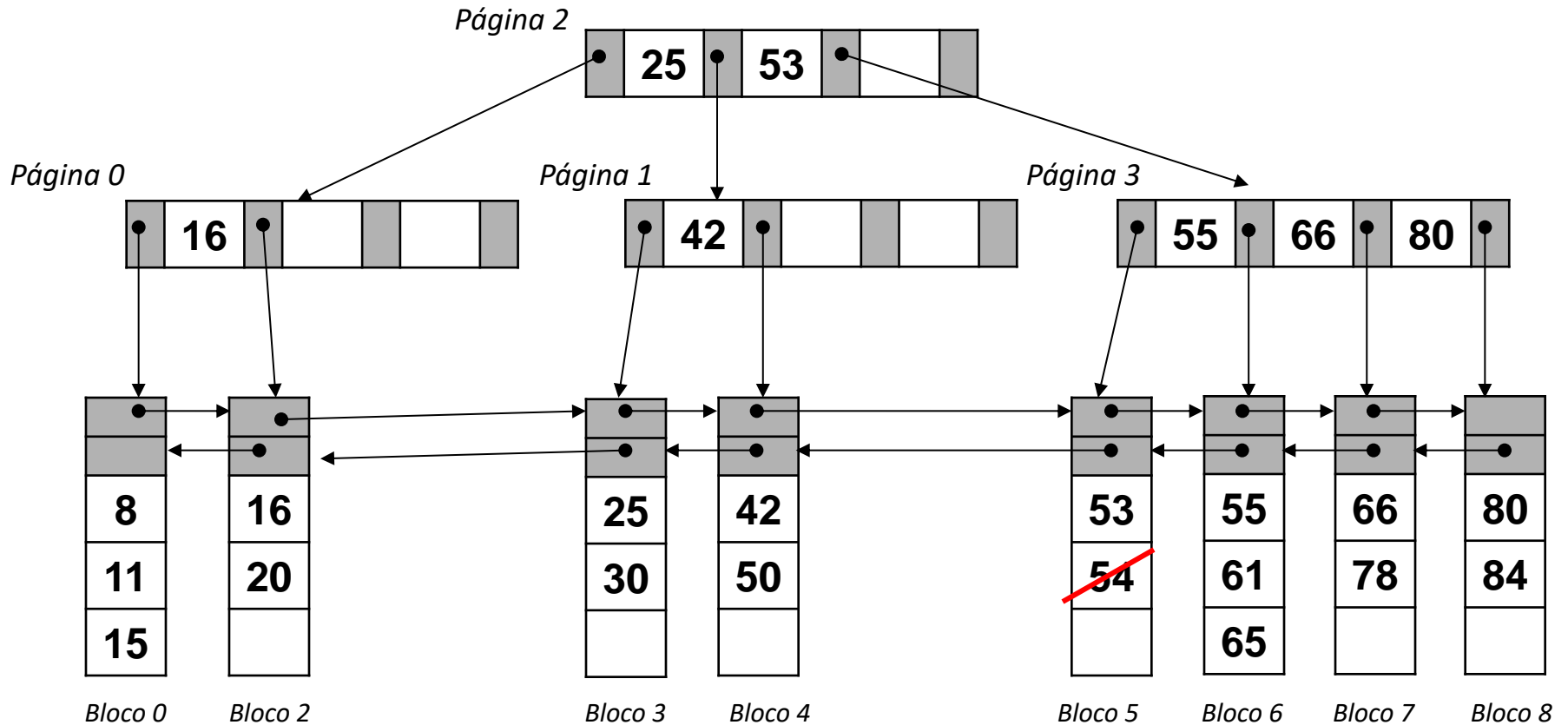
EXERCÍCIO 2

■ **7**, 27, 54, 53, 65, 84, 42, 55, 50, 20, 16, 80



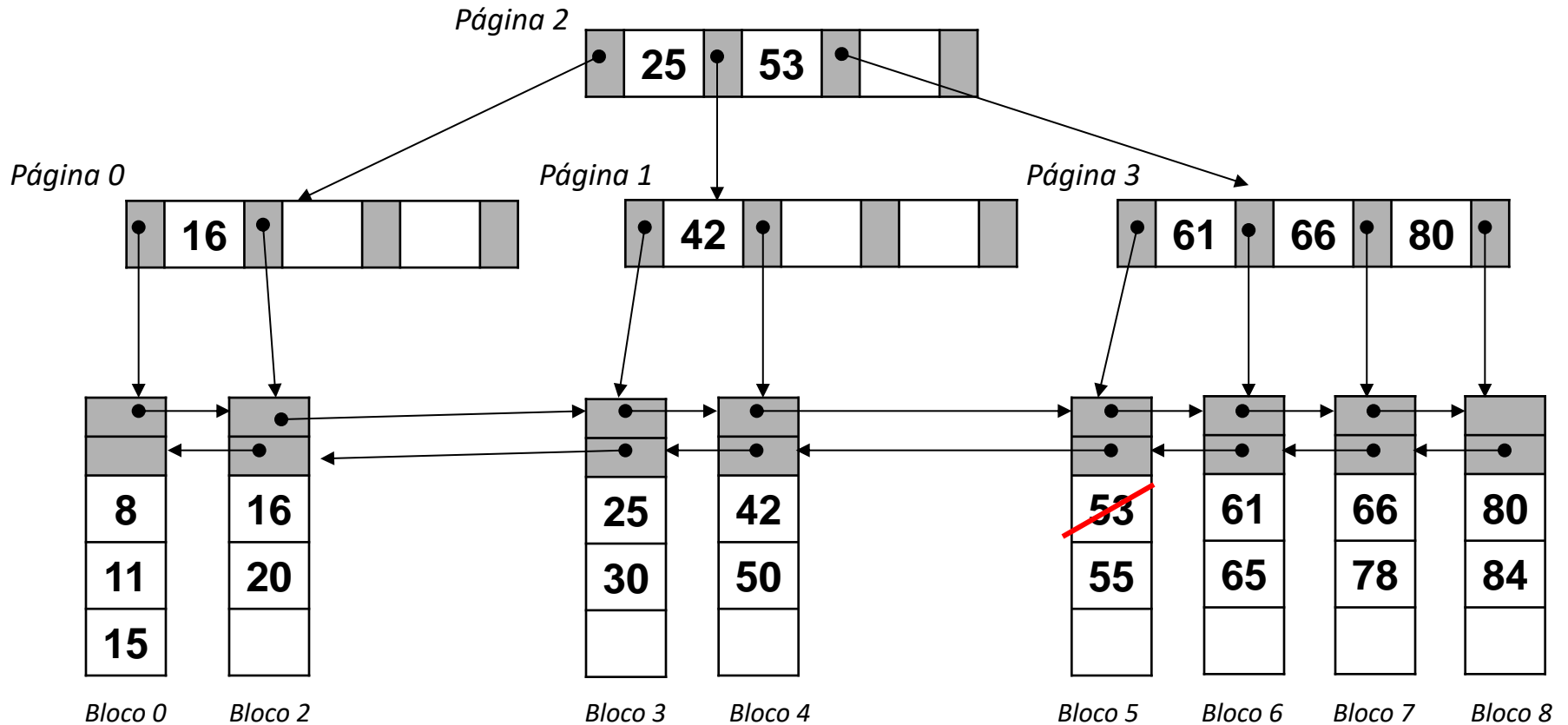
EXERCÍCIO 2

■ **27**, 54, 53, 65, 84, 42, 55, 50, 20, 16, 80



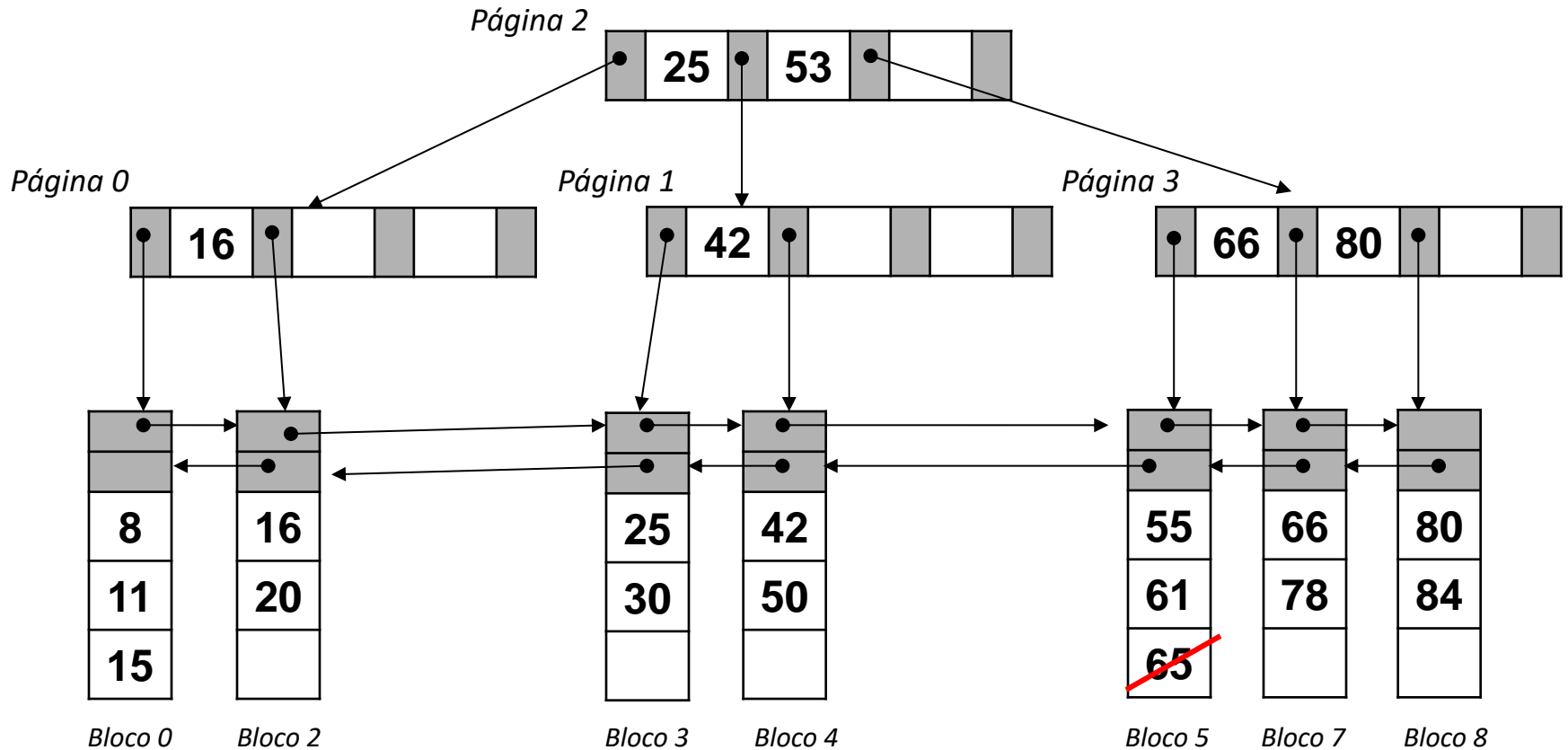
EXERCÍCIO 2

■ **54**, 53, 65, 84, 42, 55, 50, 20, 16, 80



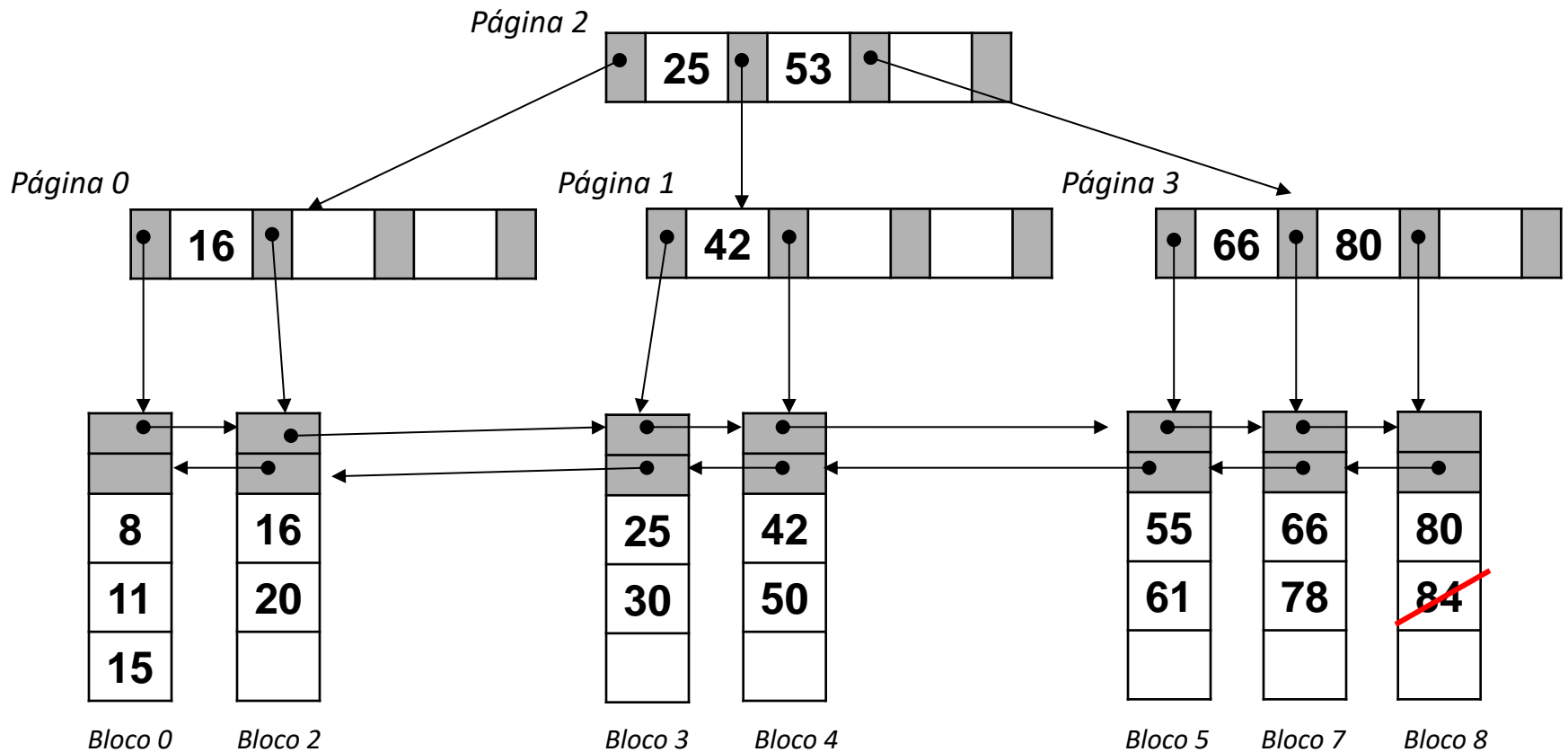
EXERCÍCIO 2

■ **53**, 65, 84, 42, 55, 50, 20, 16, 80



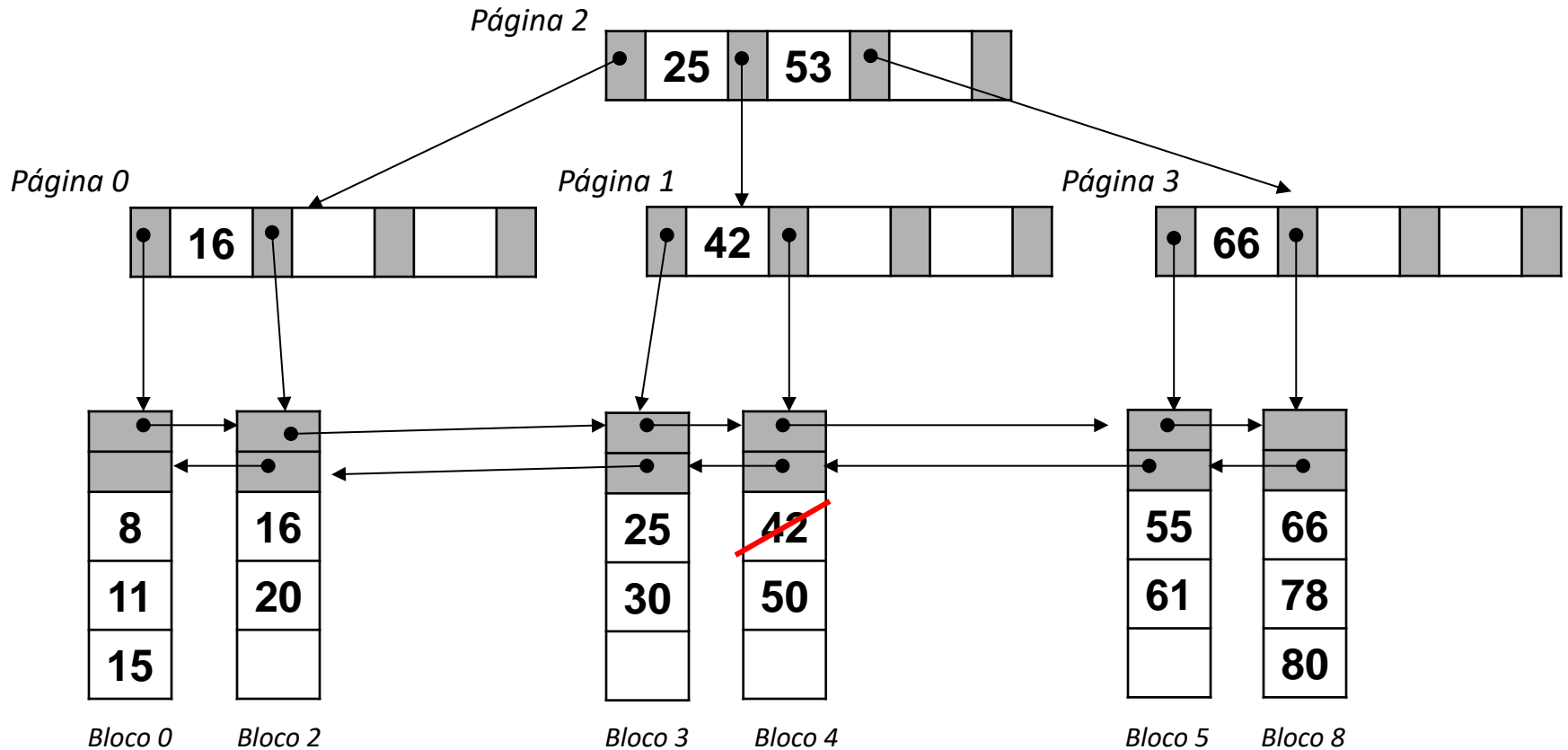
EXERCÍCIO 2

■ **65**, 84, 42, 55, 50, 20, 16, 80



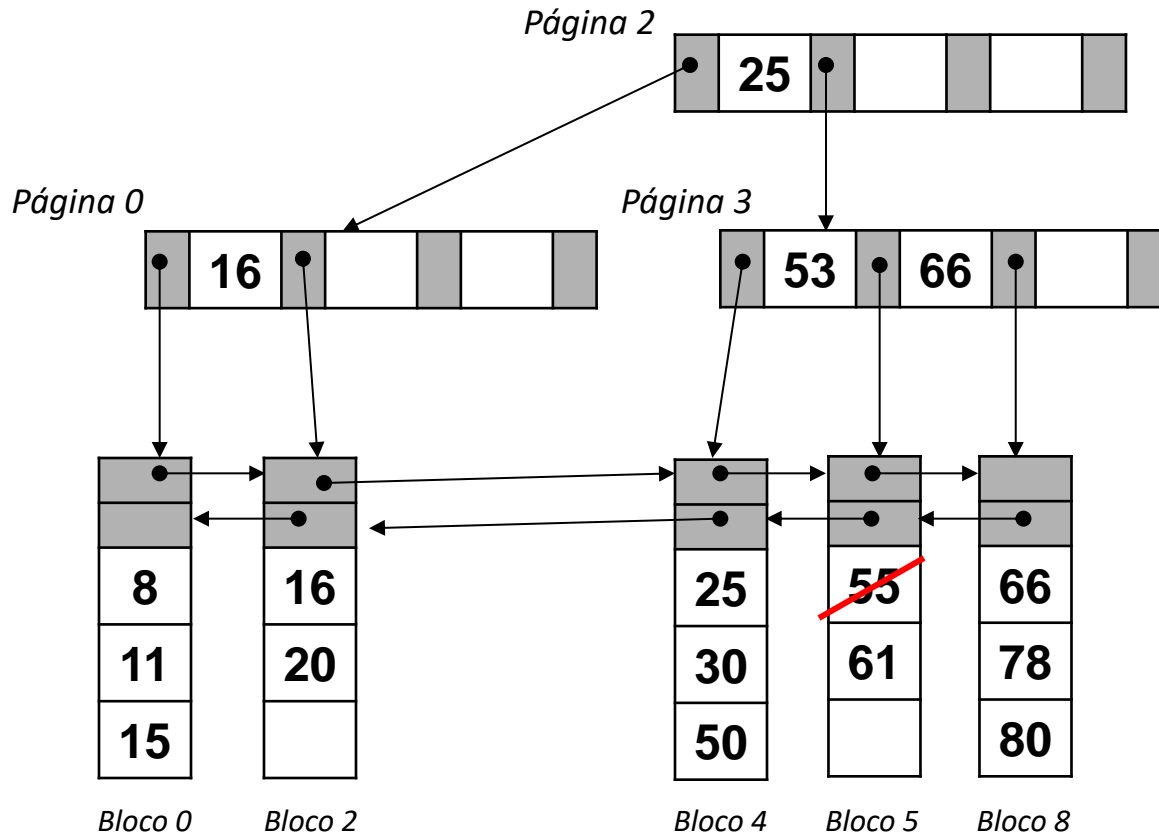
EXERCÍCIO 2

■ **84**, 42, 55, 50, 20, 16, 80



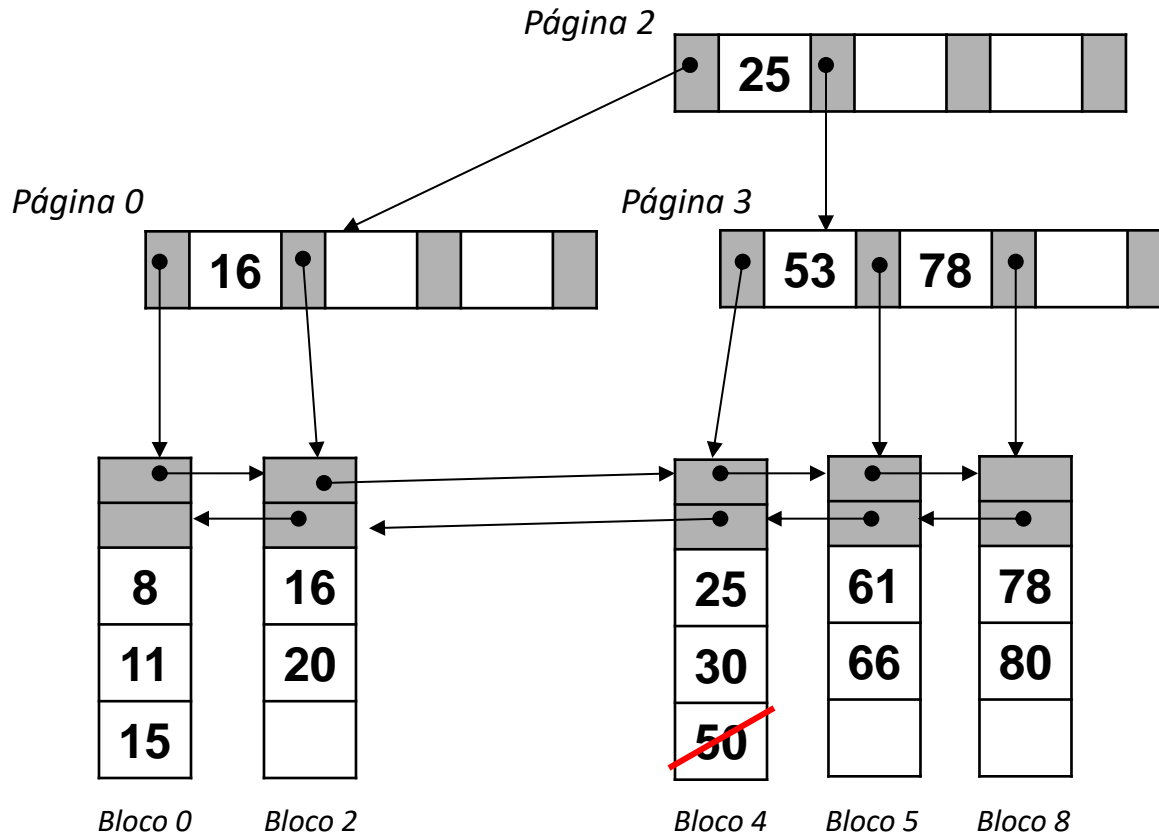
EXERCÍCIO 2

■ **42**, 55, 50, 20, 16, 80



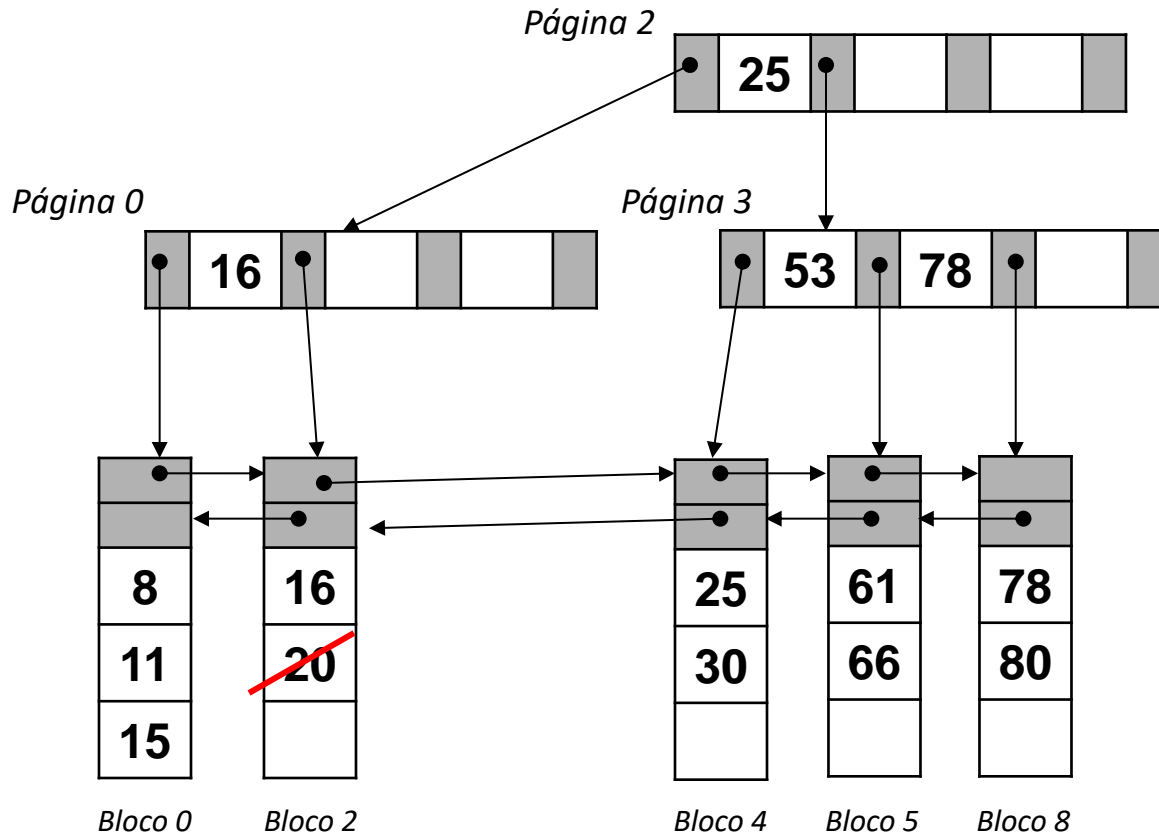
EXERCÍCIO 2

■ **55**, 50, 20, 16, 80



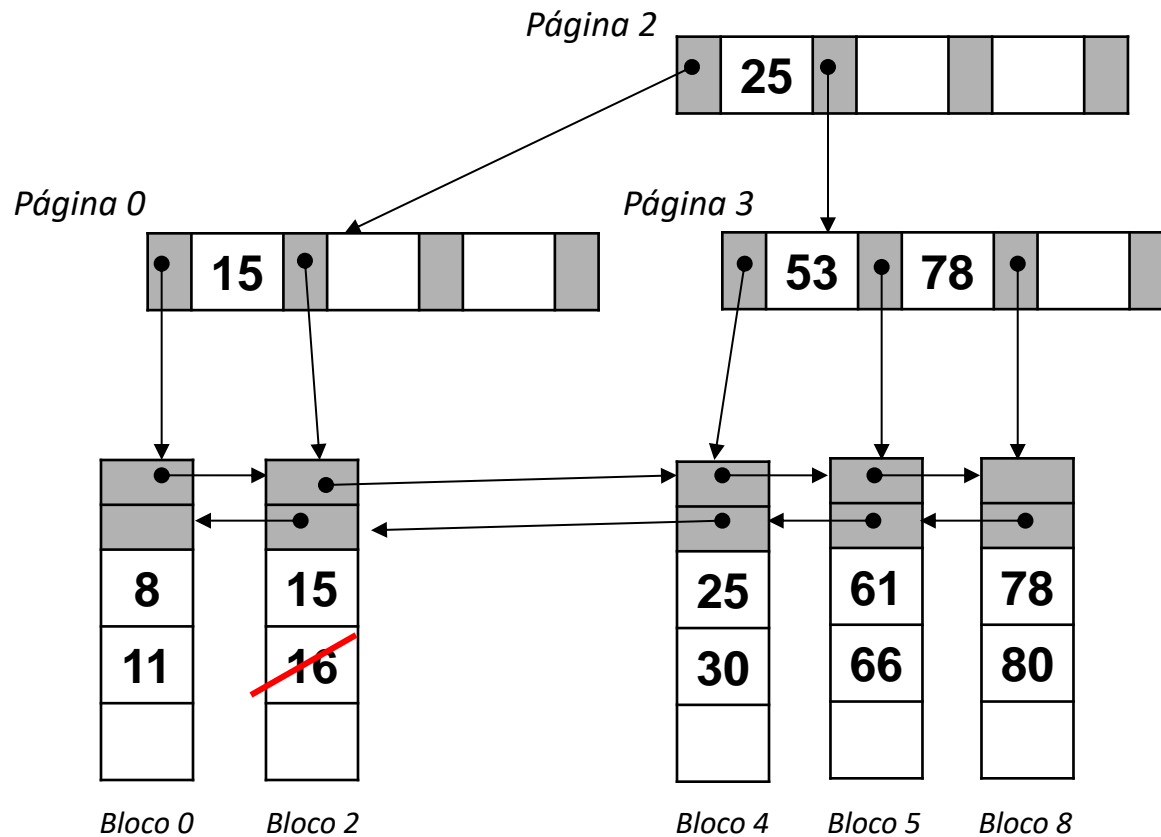
EXERCÍCIO 2

■ **50**, 20, 16, 80



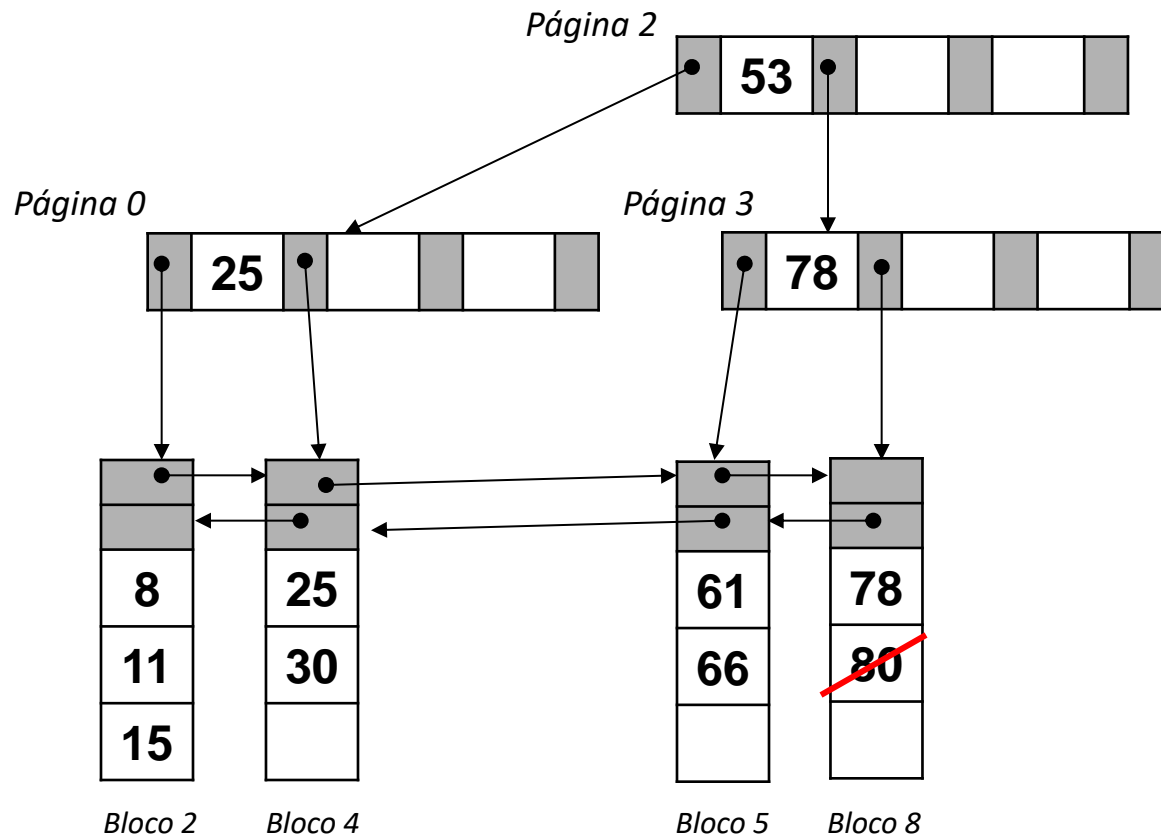
EXERCÍCIO 2

■ **20**, 16, 80



EXERCÍCIO 2

■ **16, 80**



EXERCÍCIO 2

■ 80

Página 0

