

Pesquisa e Ordenação de Dados

Unidade 5.5:

Árvores B

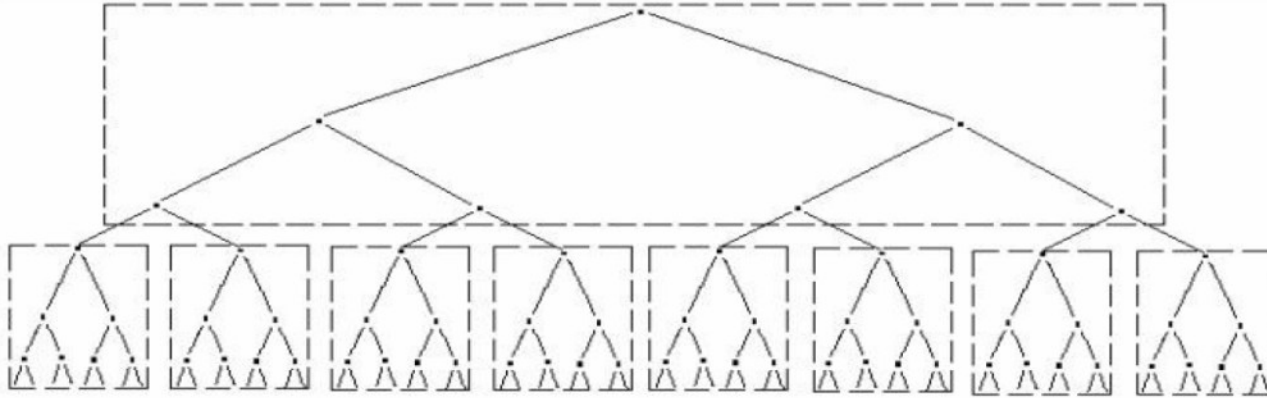


Árvore B



- Rudolf Bayer e Edward Meyers McCreight - 1972
- Generalização das árvores de busca binária
- **BST** (ordem 2):
 - cada nó tem no máximo 2 filhos
 - armazena 1 chave de busca por nó
 - balanceamento requer rotinas adicionais a cada inclusão/exclusão
- **Árvore B** (ordem **m**):
 - generalização da BST
 - cada nó (também chamado de **página**) tem até **m** filhos (**m**-ária / *m*-way)
 - armazena até **m-1** chaves de busca por nó, mantendo-as ordenadas
 - autobalanceada
 - Todas as folhas estão no **mesmo nível**

Árvore B

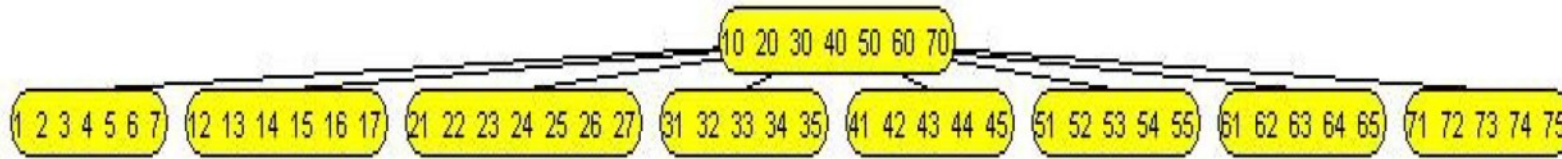


Árvore binária

Nós: 63

Chaves: 63

Altura: 6



Árvore de ordem 8
(até 7 chaves por nó)

Nós: 9

Chaves: até 63

Altura: 2

Motivação

- Alto custo de acesso à memória secundária
 - Um acesso a disco equivale a cerca de 200.000 instruções do computador
 - Solução: tentar ler/escrever o máximo de informações a cada acesso ao disco = menos acessos ao disco
- As árvores B (e sua variante B+) são otimizadas para situações em que parte ou toda a árvore é mantida na memória secundária.
 - arquivos de índices, bancos de dados, sistemas de arquivos
- Os algoritmos de árvores B copiam páginas selecionadas do disco para a memória principal e gravam novamente em disco as páginas que foram alteradas.

Árvore B

- Nó (página)
 - sequência ordenada de chaves
 - conjunto de ponteiros para nós filhos
 - $n^{\circ} \text{ de ponteiros} = n^{\circ} \text{ de chaves} + 1$
- Ordem
 - número máximo de ponteiros que podem ser armazenados em um nó
 - Ex: árvore B de ordem 8
 - máximo de 7 chaves e 8 ponteiros (filhos) por nó

Árvore B de ordem m

Propriedades

- **FILHOS**

- Máximo de filhos por nó: m
- Mínimo de filhos por nó:
 - folhas: 0
 - raiz: 2 (a menos que a raiz também seja uma folha)
 - nós internos: $\lceil m/2 \rceil$

- **CHAVES**

- Máximo de chaves por nó: $m-1$
- Mínimo de chaves por nó:
 - raiz: 1
 - todos os demais nós: $\lceil m/2 \rceil - 1$

taxa de ocupação

- Um nó interno com k filhos (onde $k \leq m$) tem exatamente $k-1$ chaves.

Árvore B de ordem m

Propriedades

- FILHOS

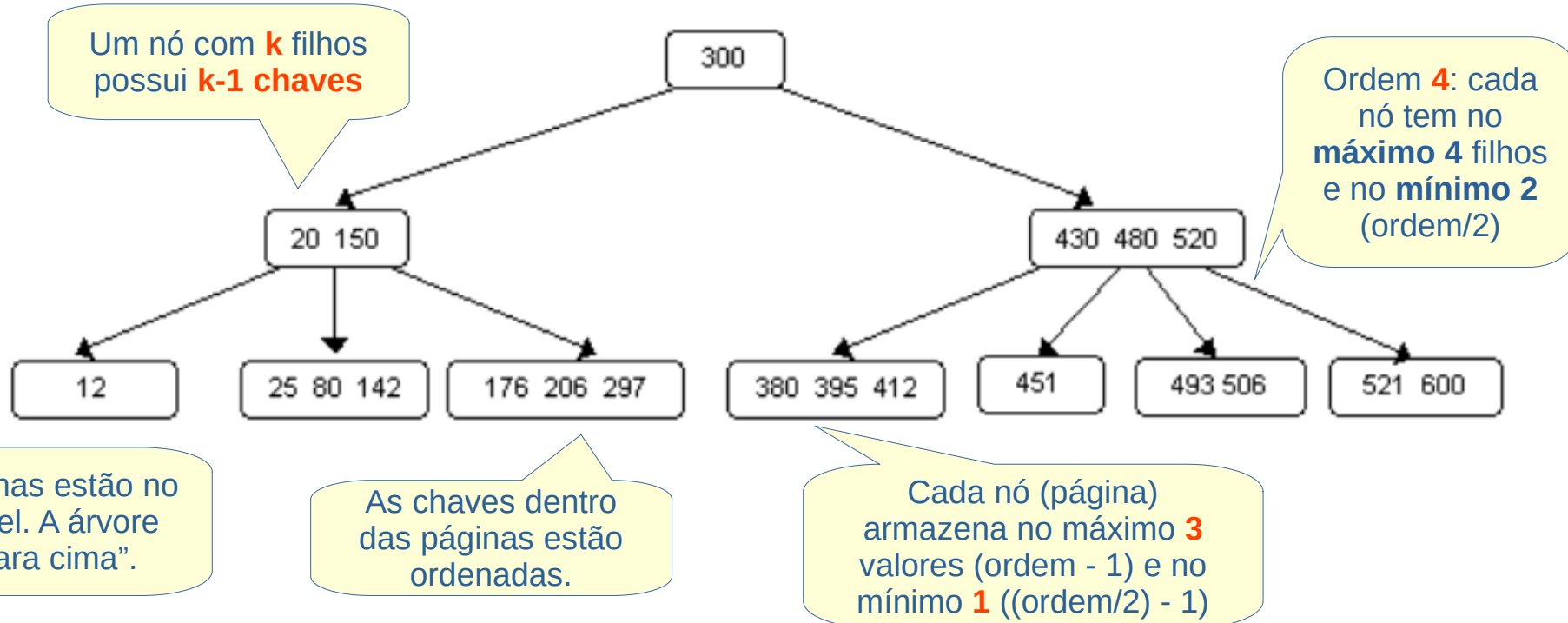
- Máximo de filhos por nó: m
- Mínimo de filhos por nó:
 - folhas: 0
 - raiz: 2 (a menos que a raiz também seja uma folha)
 - nós internos: $\lceil m/2 \rceil$

- CHAVES

Os limites de lotação de cada nó da árvore B garantem que a árvore **não fique alta demais e que não haja desperdício de espaço** pela utilização de páginas quase vazias (FERRAZ, 2003)

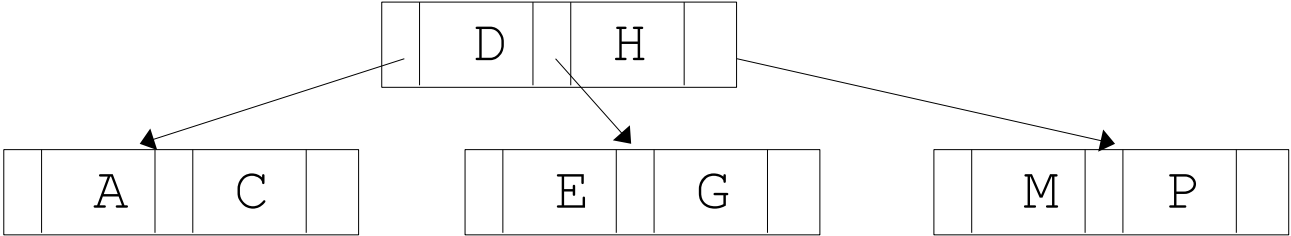
- Um nó interno com k filhos (onde $k \leq m$) tem exatamente $k-1$ chaves.

Exemplos: árvore de ordem 4

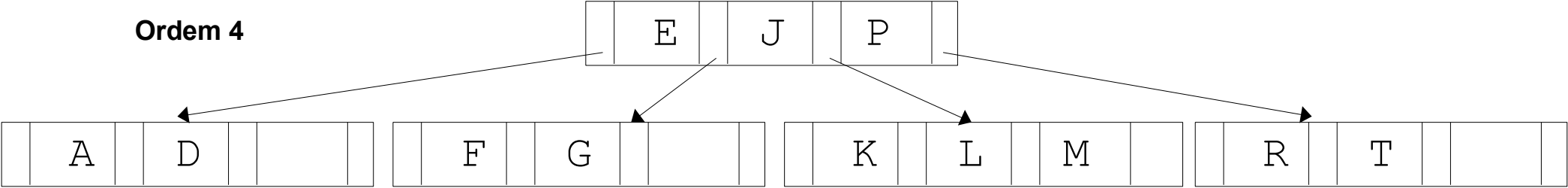


Exemplos

Ordem 3

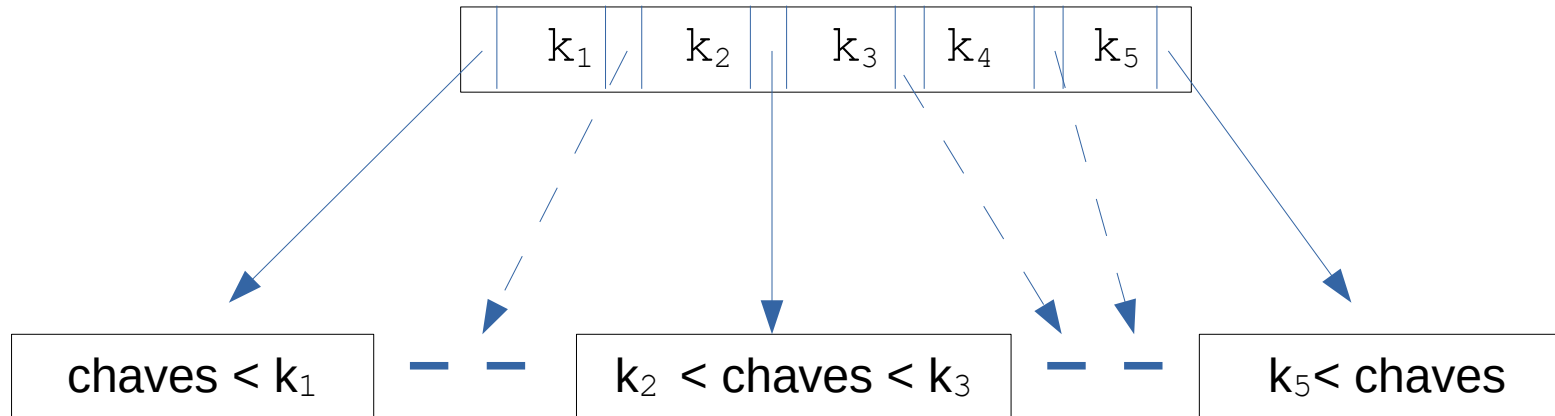


Ordem 4



Árvore B

- As chaves em cada nó estão ordenadas e servem como separadores dos filhos
 - A subárvore à **esquerda** de uma chave **K** contém apenas chaves **menores** do que **K**;
 - A subárvore à **direita** de **K** contém apenas chaves **maiores** do que **K**.



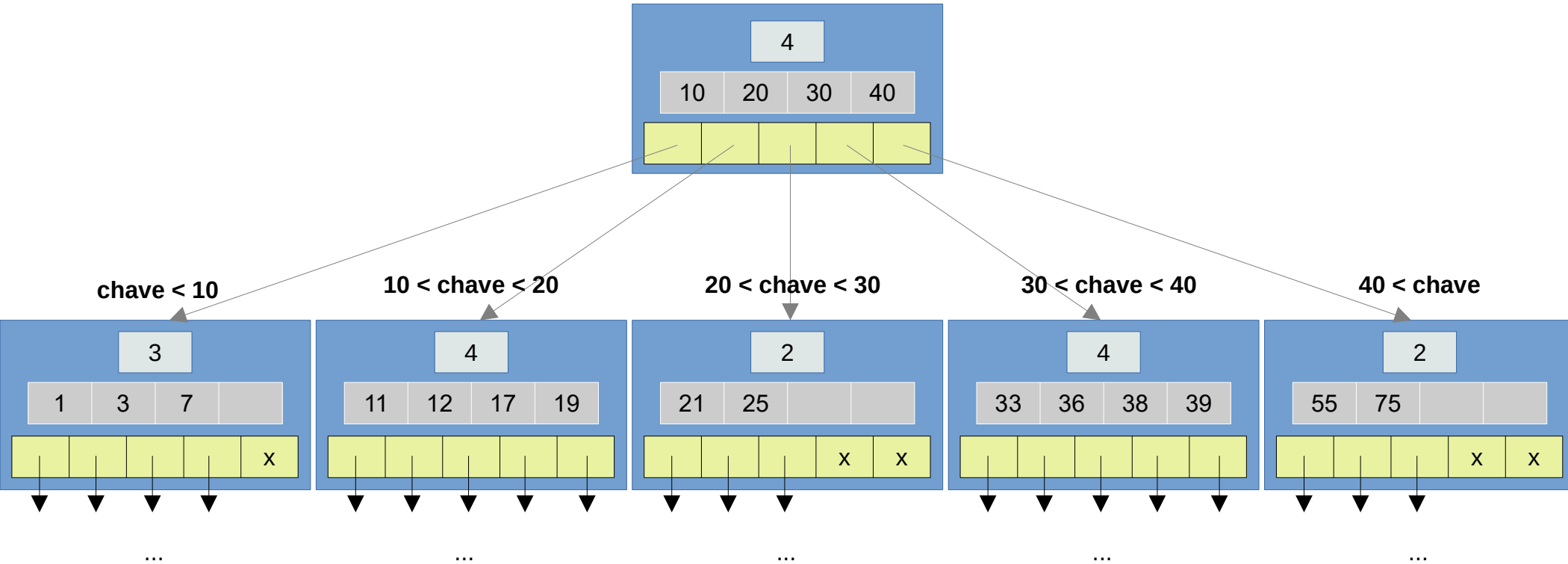
Árvore B

- Estrutura de uma página

```
# define m 5                // árvore de ordem 5

typedef struct pagina {
    short int n;              // numero de chaves inseridas na página
    int chaves[m-1];          // vetor de chaves
    struct pagina filhas[m];  // ponteiros para as páginas filhas
} Pagina;
```

Árvore B



Operações em Árvores B

- **Inserção**

- A inserção sempre ocorre nas folhas.
- Primeiramente, localiza-se a folha onde a inserção deve ocorrer.
- **Caso 1:** se há espaço disponível na página (número de chaves $< m-1$), basta inserir a nova chave, mantendo a ordenação;
- **Caso 2:** se não há espaço disponível na página (*overflow*), esta deve ser dividida em duas (*split*):
 - Encontre o elemento central (mediana) da página – incluindo o elemento que está sendo inserido;
 - Valores menores que a mediana são mantidos na folha à esquerda e valores maiores são colocados em uma nova folha (recém-alocada) à direita;
 - O elemento mediano deve ser promovido (*promote*) para a página pai, levando consigo o ponteiro para a página recém-alocada.
 - Isto pode gerar um novo split no pai, no pai do pai, etc.... podendo se propagar até a raiz.

Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**
 - 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7



Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**
 - 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

• 8 • 29 • 37 • 45 •

Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

não há espaço

mediana = 29

realizar split/promote

• 8 • 29 • 37 • 45 •



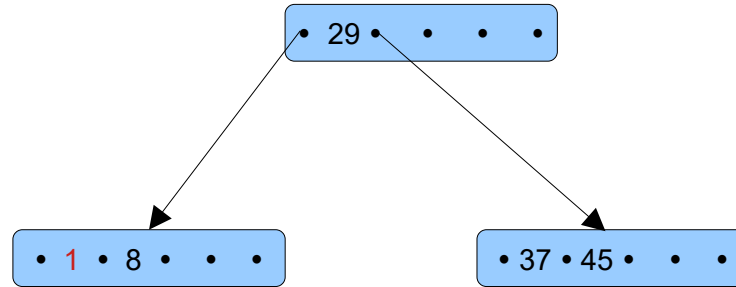
Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, **1**, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

árvore cresceu em altura

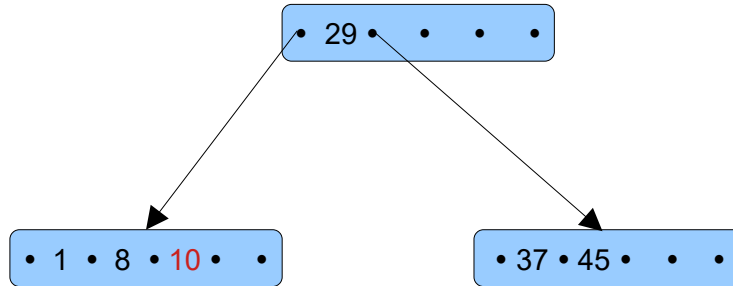


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

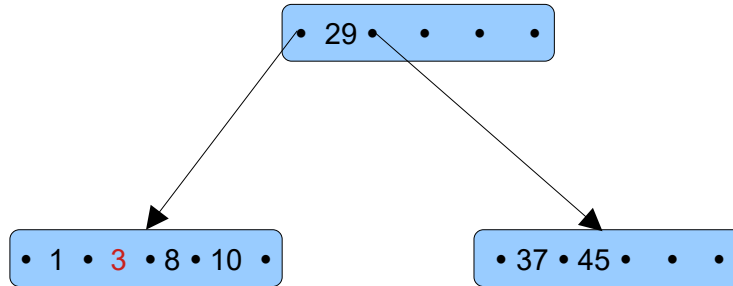


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

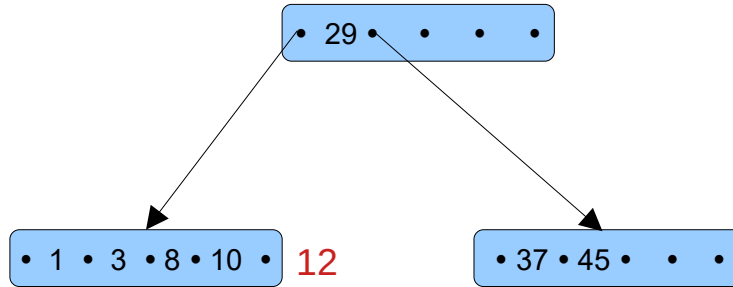


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7



não há espaço

mediana = 8

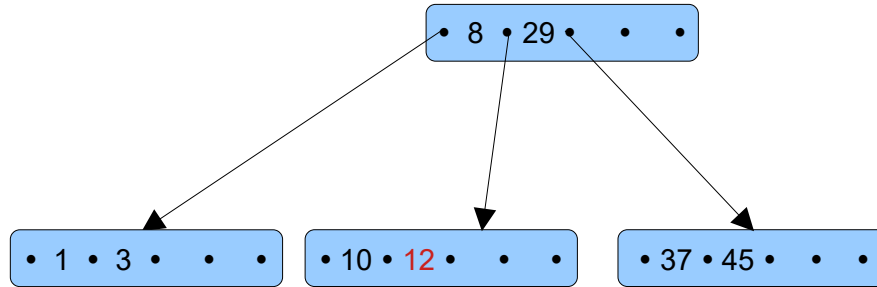
realizar split/promote

Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

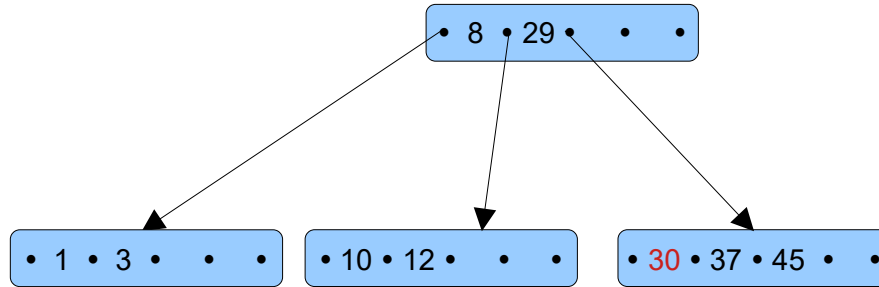


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

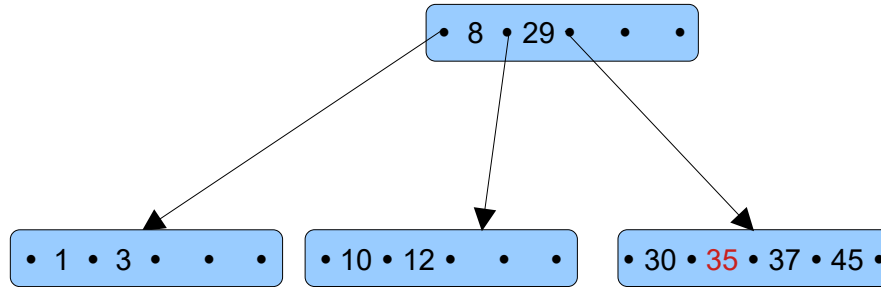


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, **35**, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

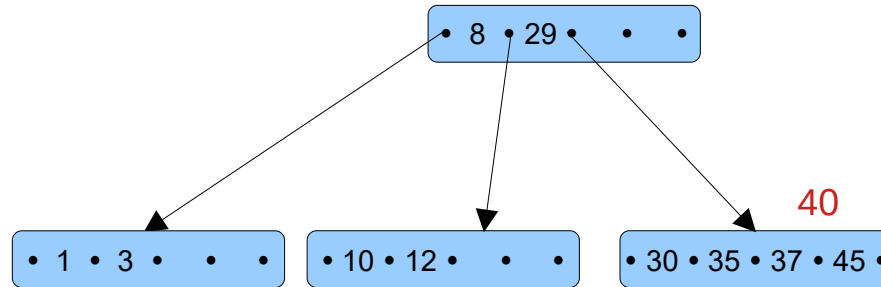


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7



não há espaço

mediana = 37

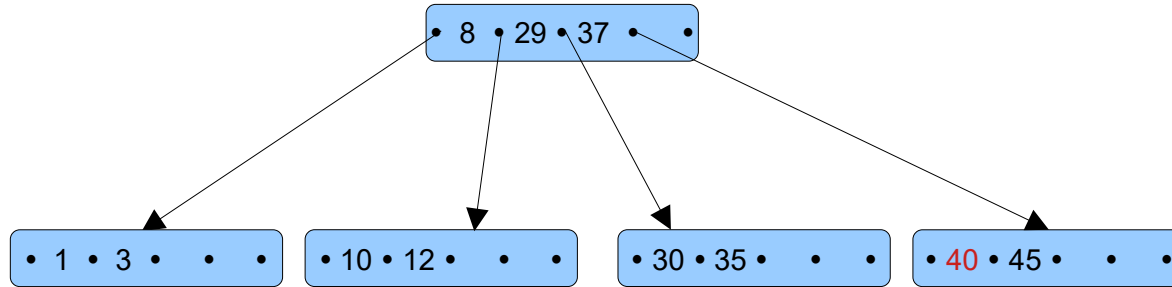
realizar split/promote

Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

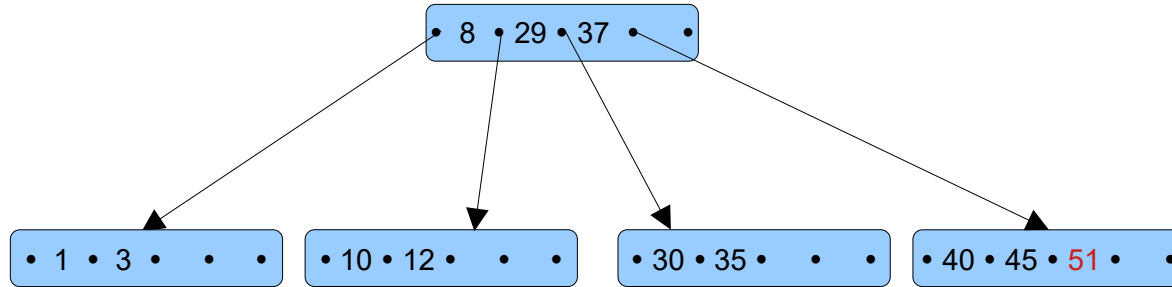
- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7



Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**
 - 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

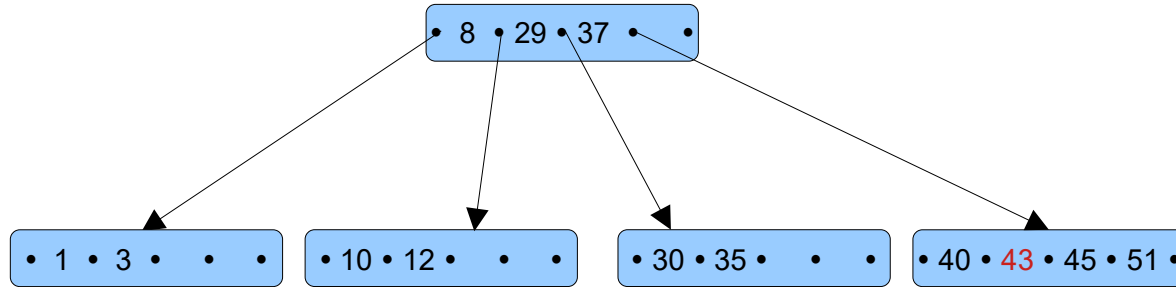


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

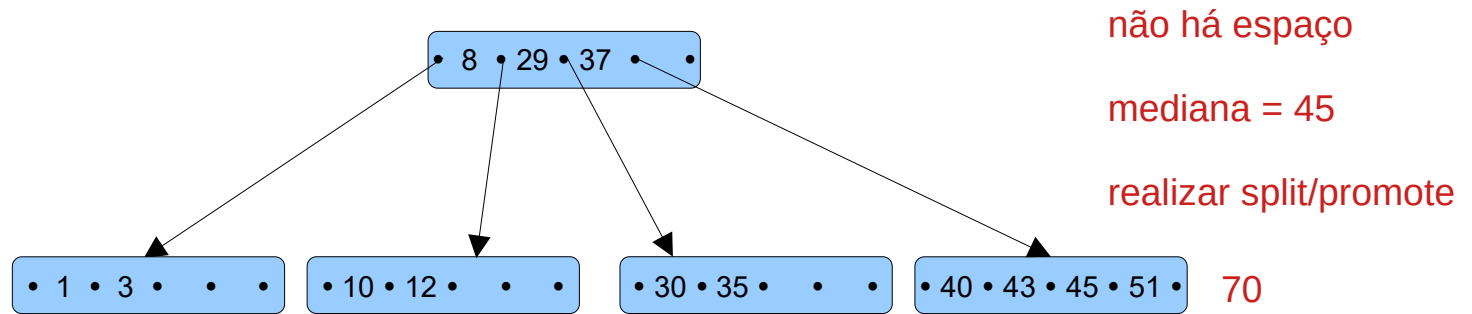


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

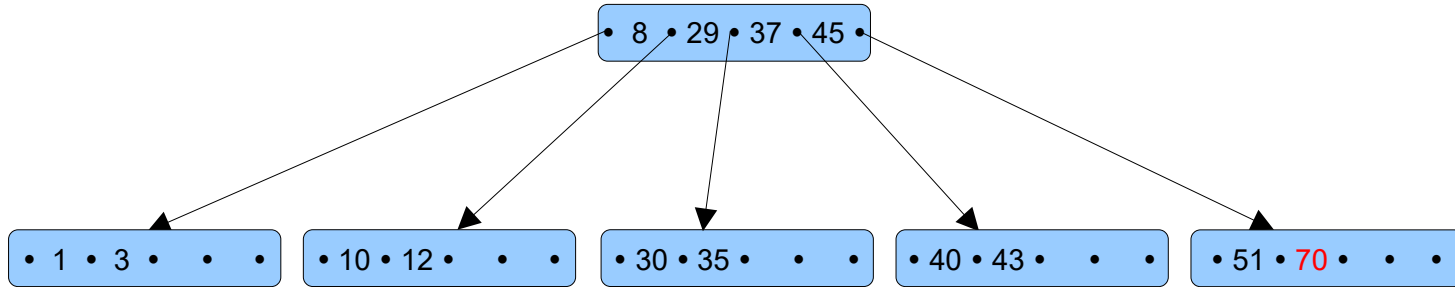


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

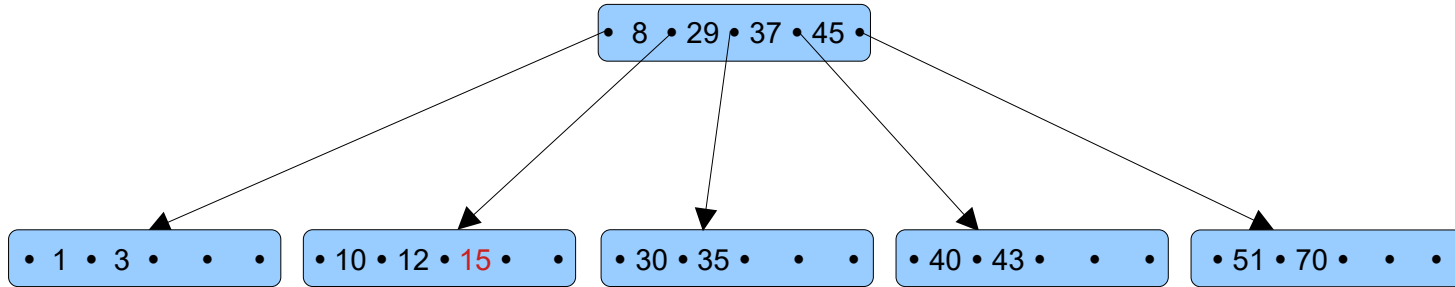


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

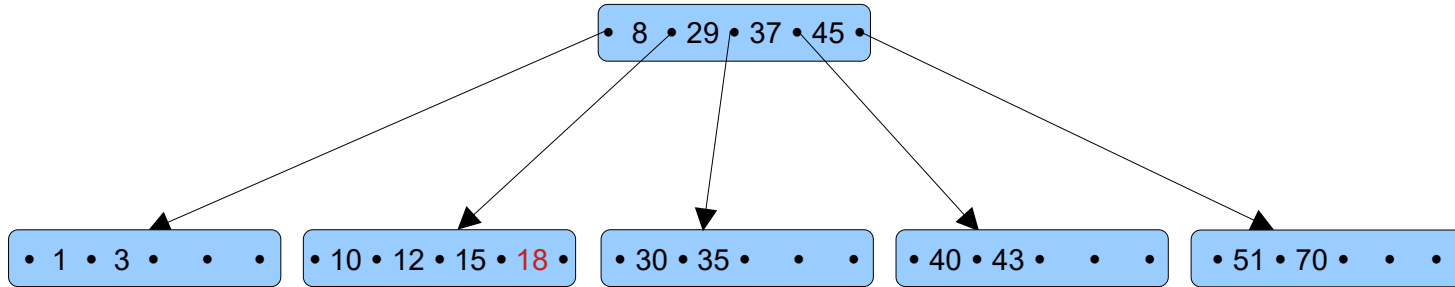
- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7



Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**
 - 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, **18**, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

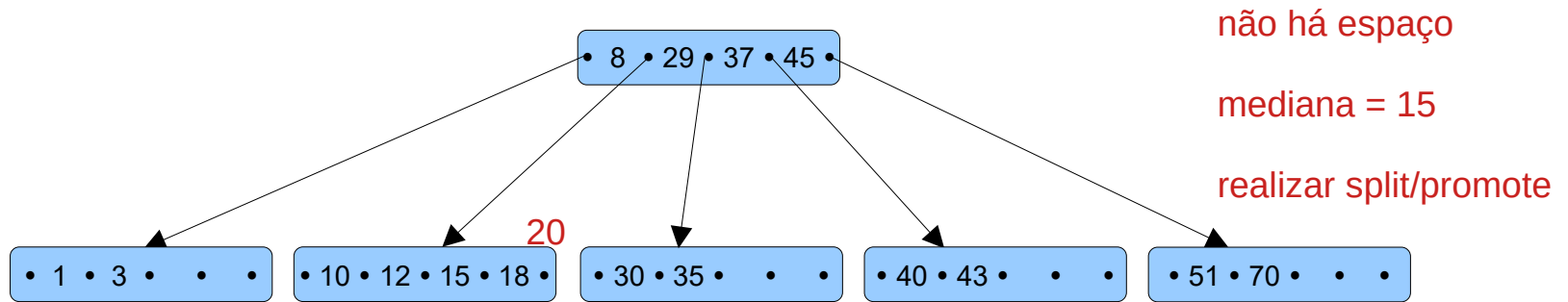


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7



Árvore B

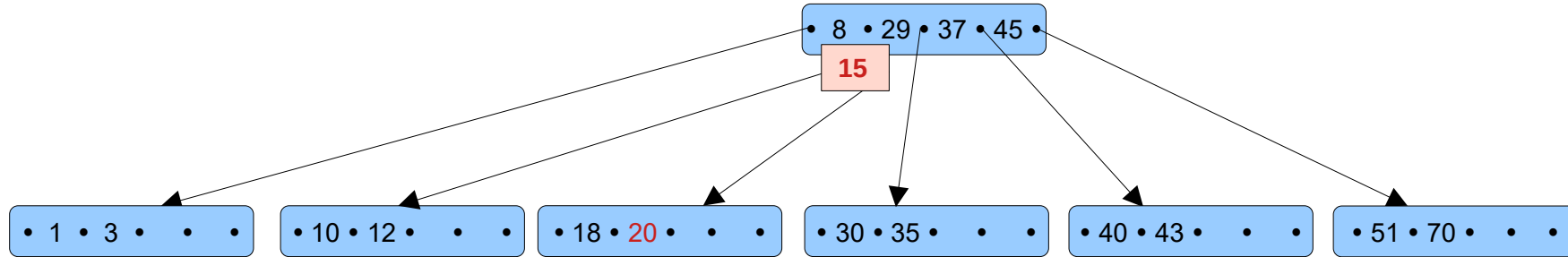
Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

overflow na raiz!

novo split/promote

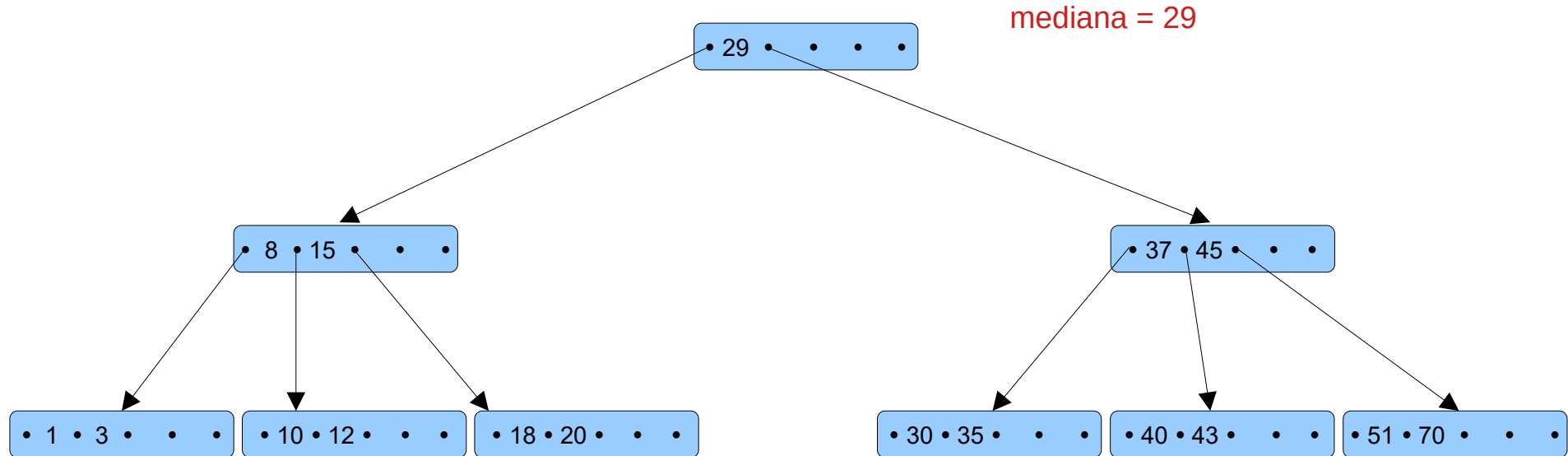


Árvore B

Inserção

- Exemplo – Árvore de ordem 5

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

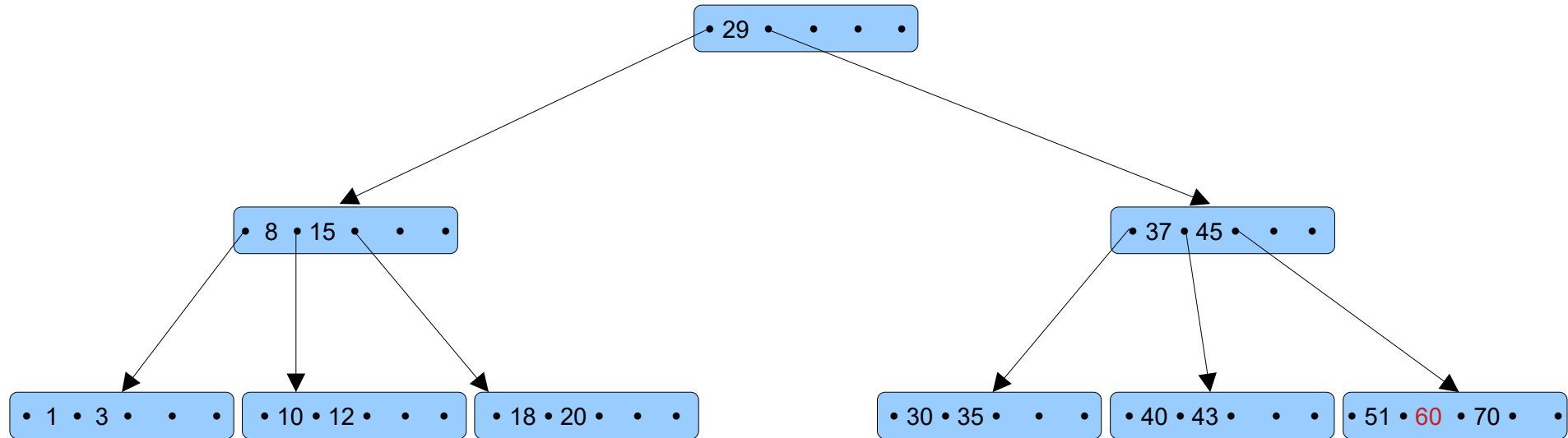


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

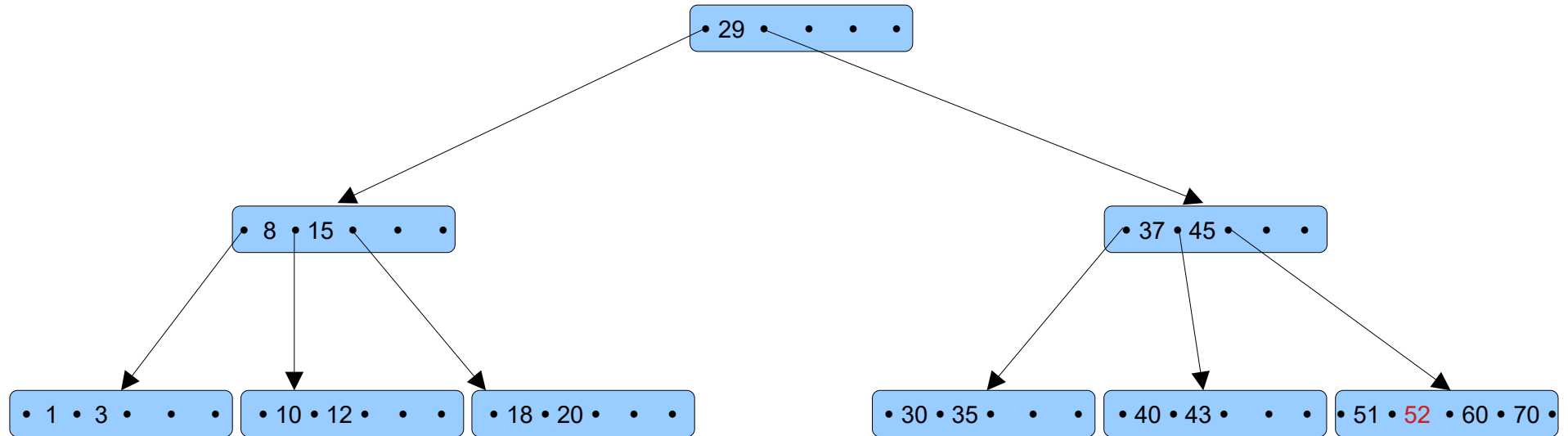


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, **52**, 77, 25, 14, 4, 41, 13, 7

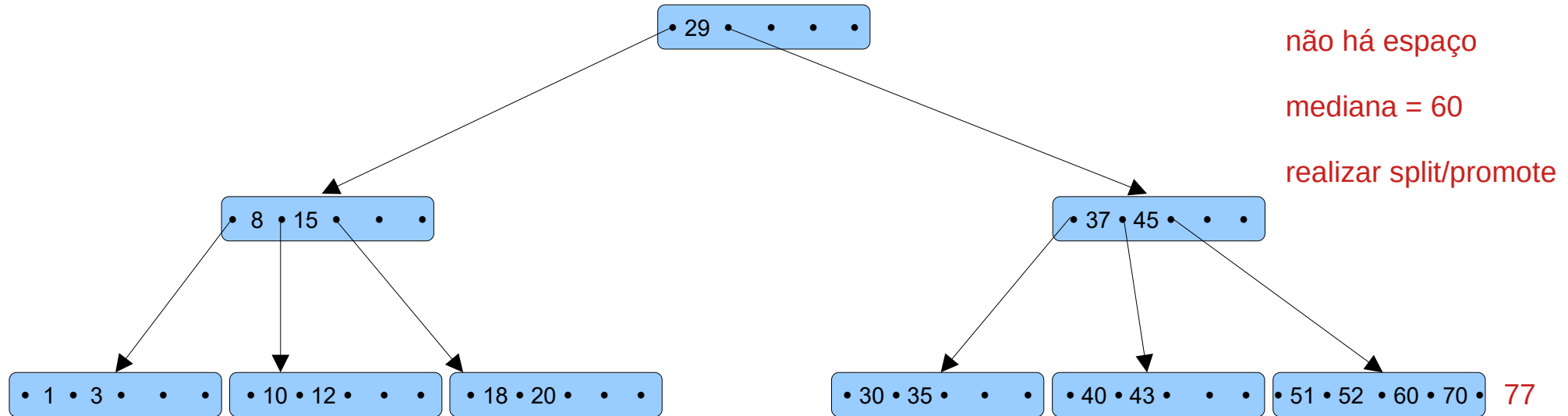


Árvore B

Inserção

- Exemplo – Árvore de ordem 5

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, **77**, 25, 14, 4, 41, 13, 7

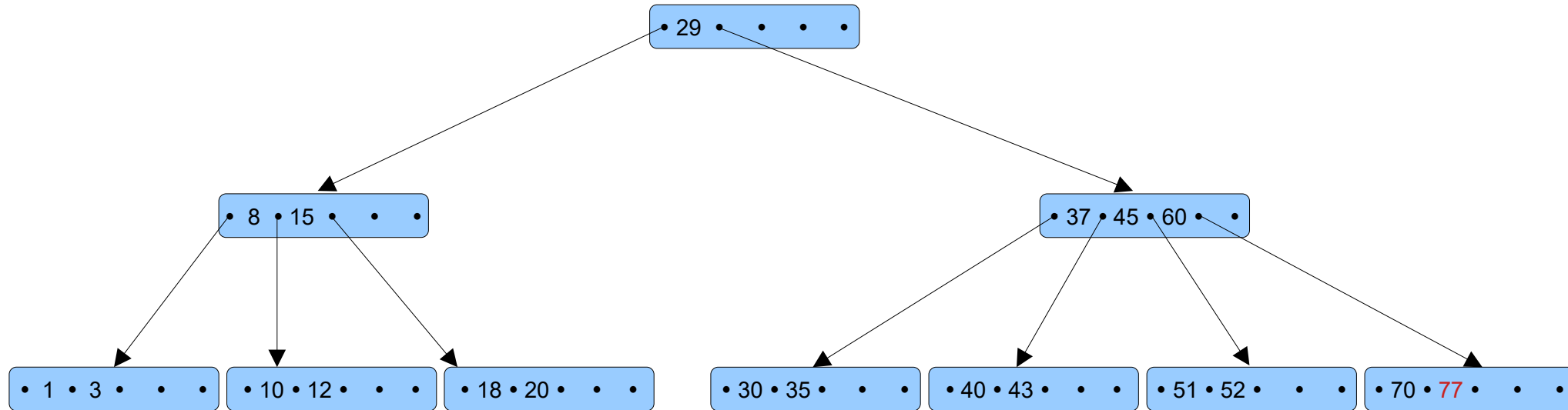


Árvore B

Inserção

- Exemplo – Árvore de ordem 5

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, **77**, 25, 14, 4, 41, 13, 7

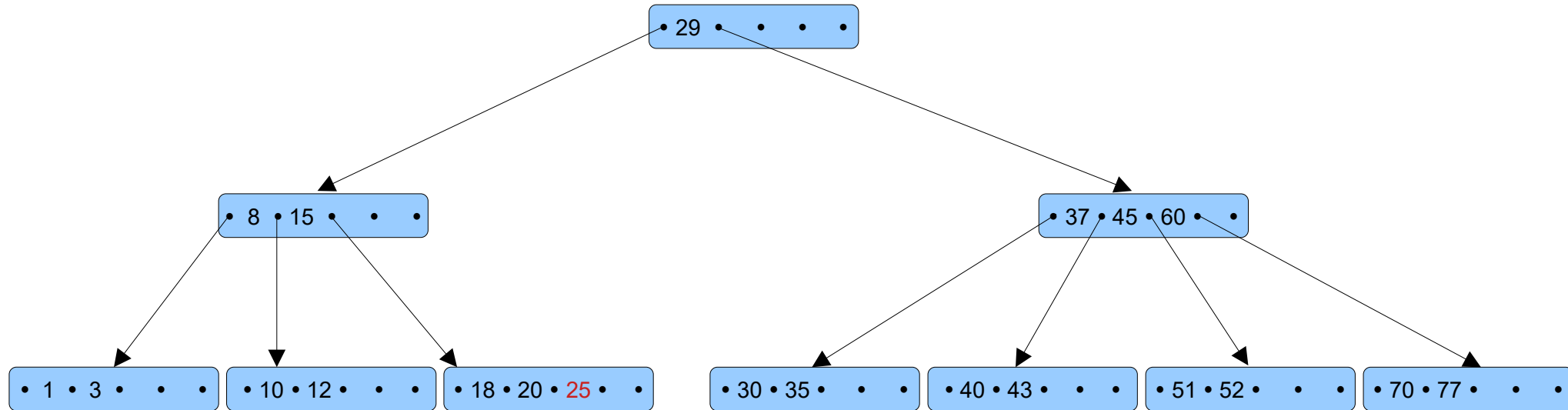


Árvore B

Inserção

- Exemplo – Árvore de ordem 5

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77,
25, 14, 4, 41, 13, 7

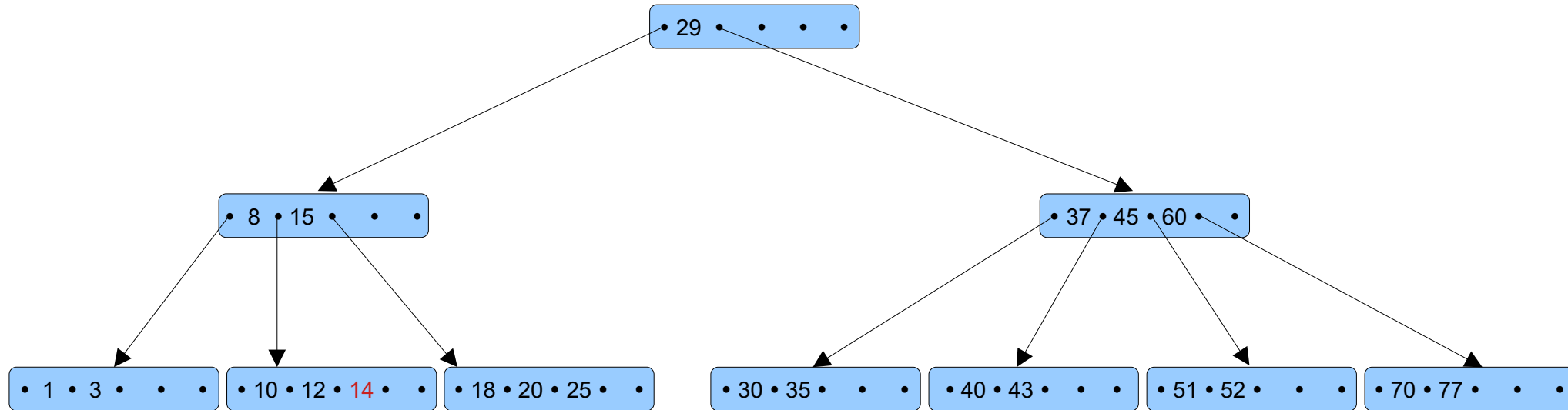


Árvore B

Inserção

- Exemplo – Árvore de ordem 5

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

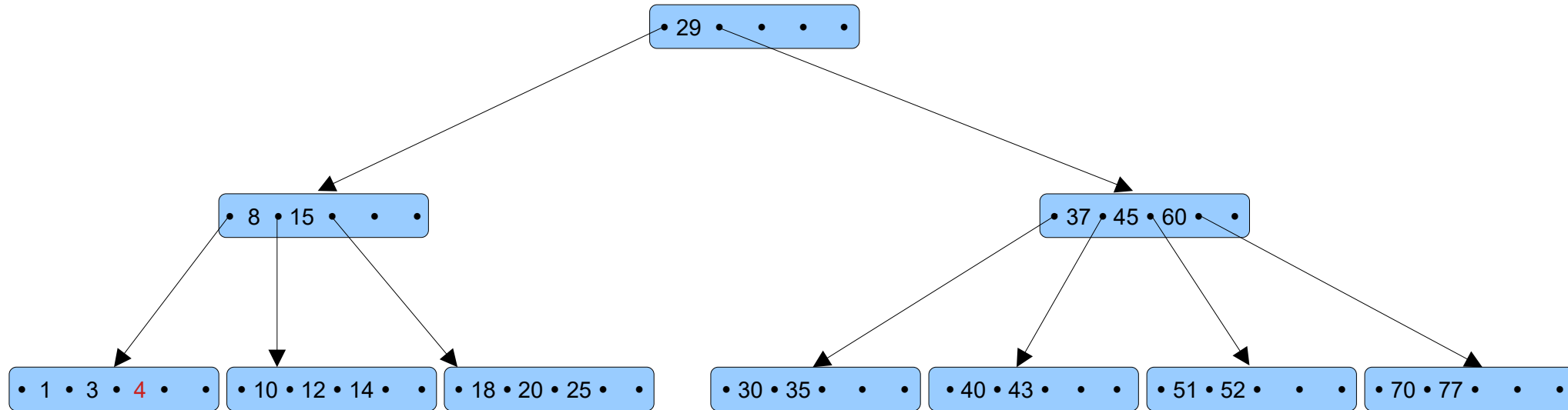


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

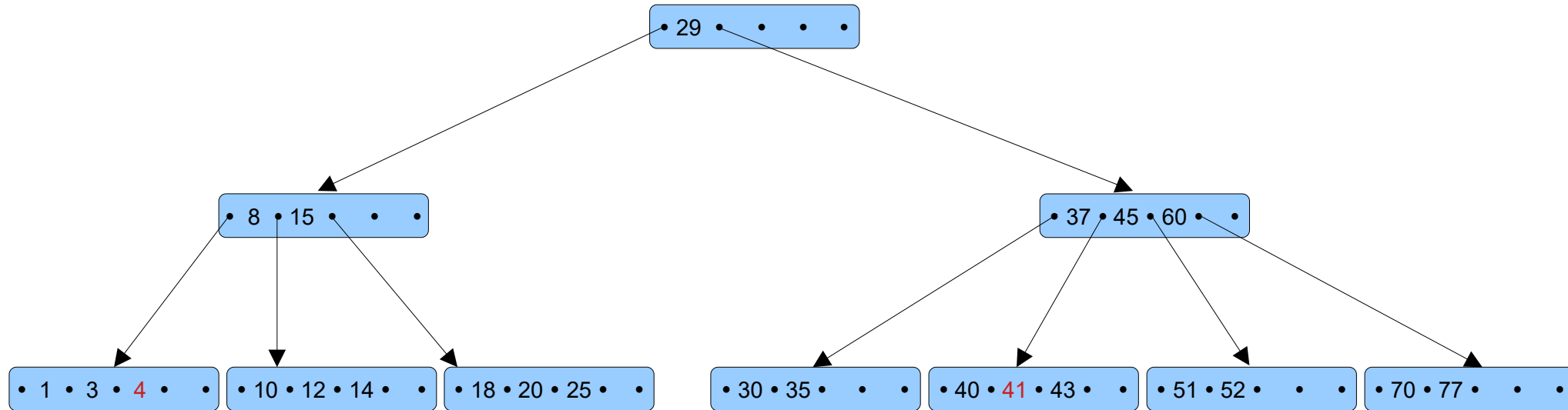


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, **41**, 13, 7

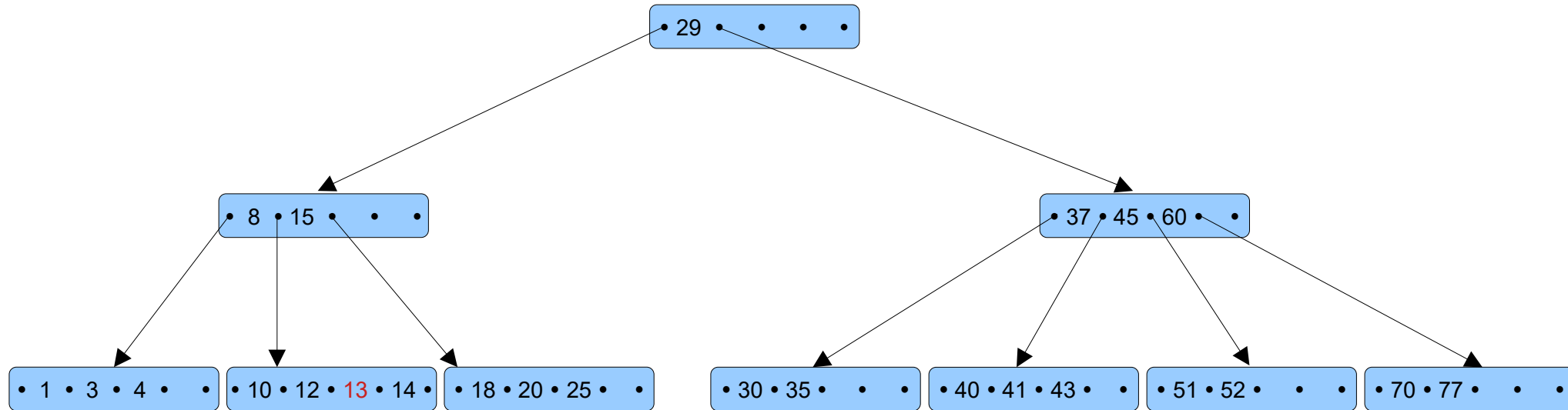


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, **13**, 7

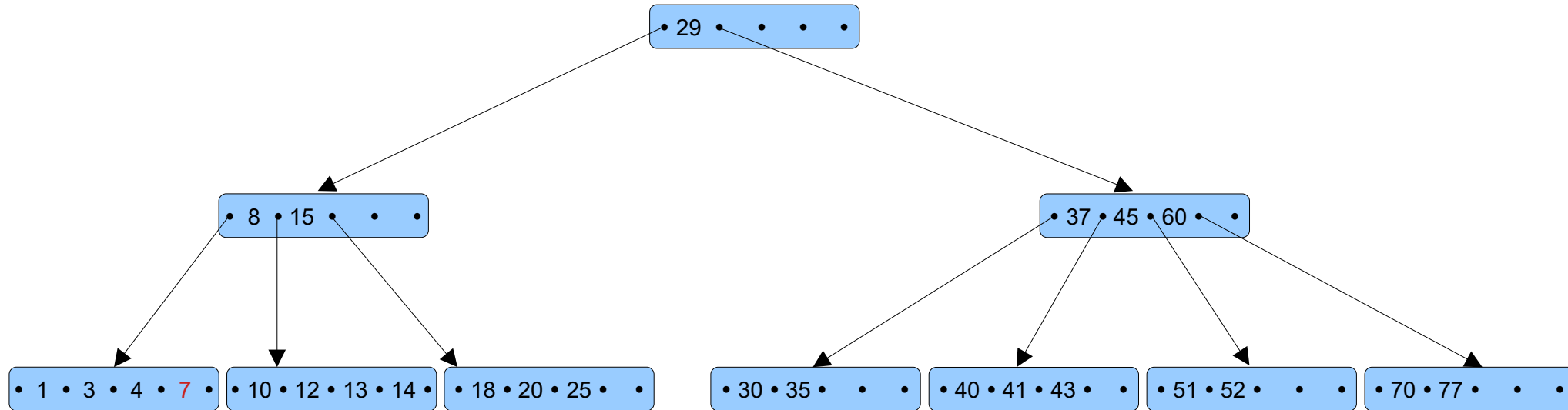


Árvore B

Inserção

- Exemplo – Árvore de ordem 5

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7

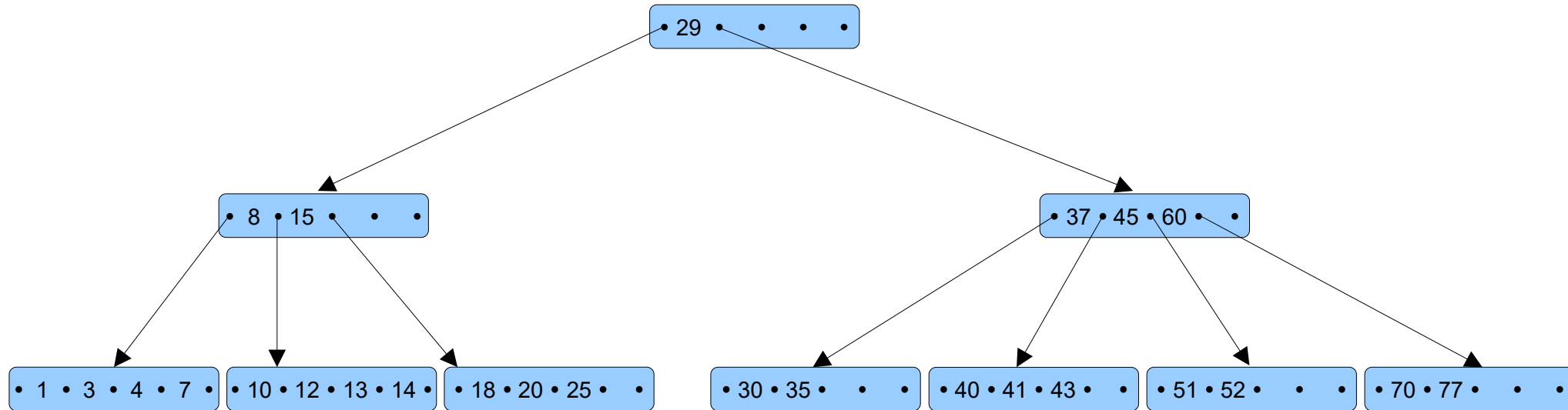


Árvore B

Inserção

- **Exemplo – Árvore de ordem 5**

- 37, 29, 8, 45, 1, 10, 3, 12, 30, 35, 40, 51, 43, 70, 15, 18, 20, 60, 52, 77, 25, 14, 4, 41, 13, 7



Operações em Árvores B

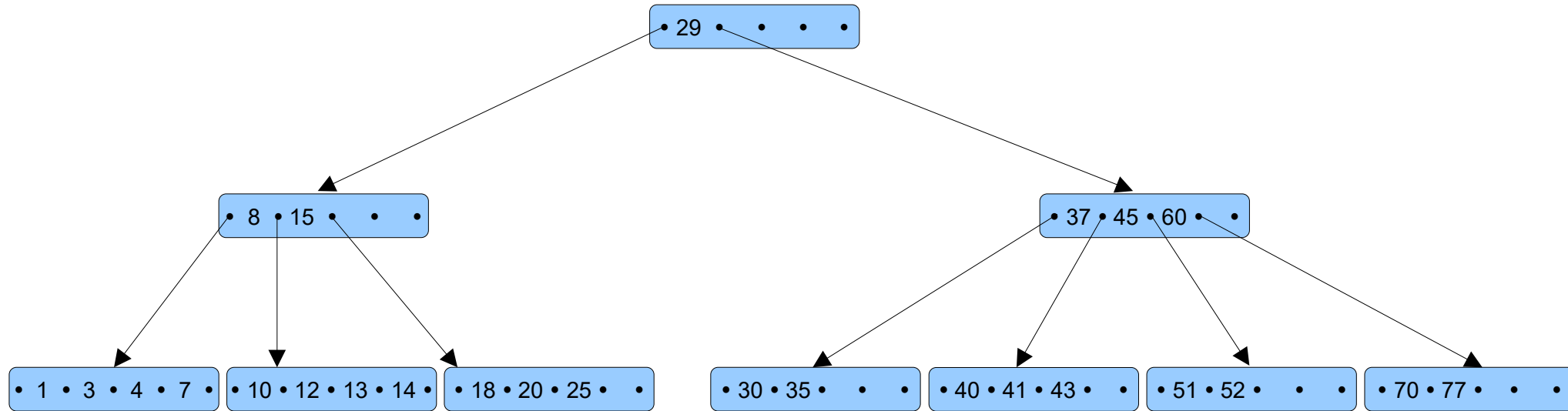
- **Busca**

- similar à pesquisa em BST, com a diferença de que um nó pode ter mais de uma chave
- Começando pela raiz:
 - Verificar se o elemento está dentro da página (pesquisa linear)
 - Se não estiver, procurar na subárvore cujos separadores são k_1 e k_2 tal que
$$k_1 < \text{valor procurado} < k_2.$$
 - Repetir até que o elemento seja encontrado (busca bem-sucedida) ou que o elemento não tenha sido encontrado em uma página folha (busca malsucedida).

Árvore B

Busca

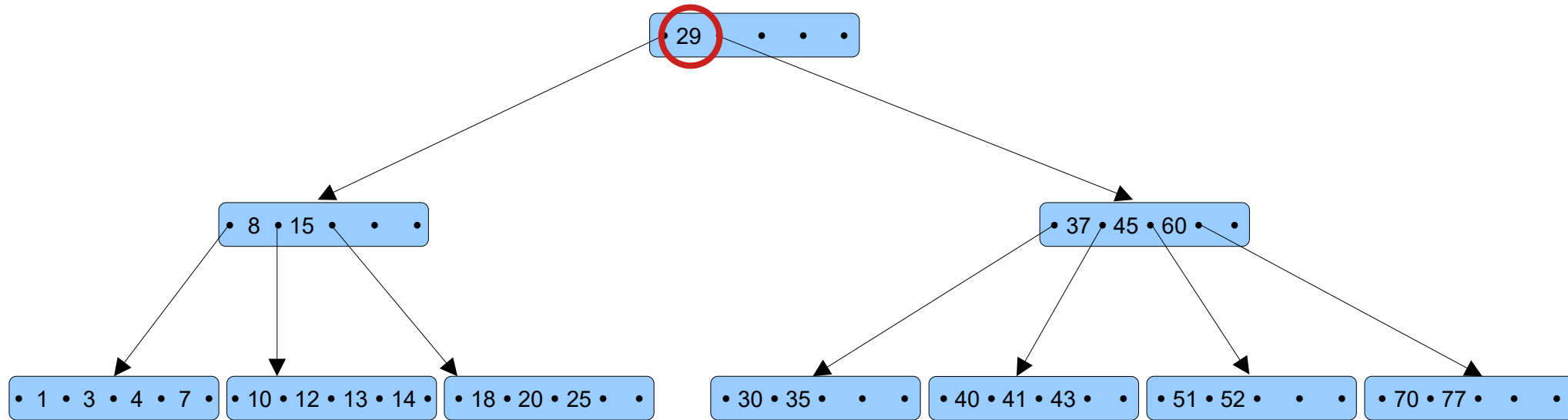
- Exemplo – Buscar a chave **25**



Árvore B

Busca

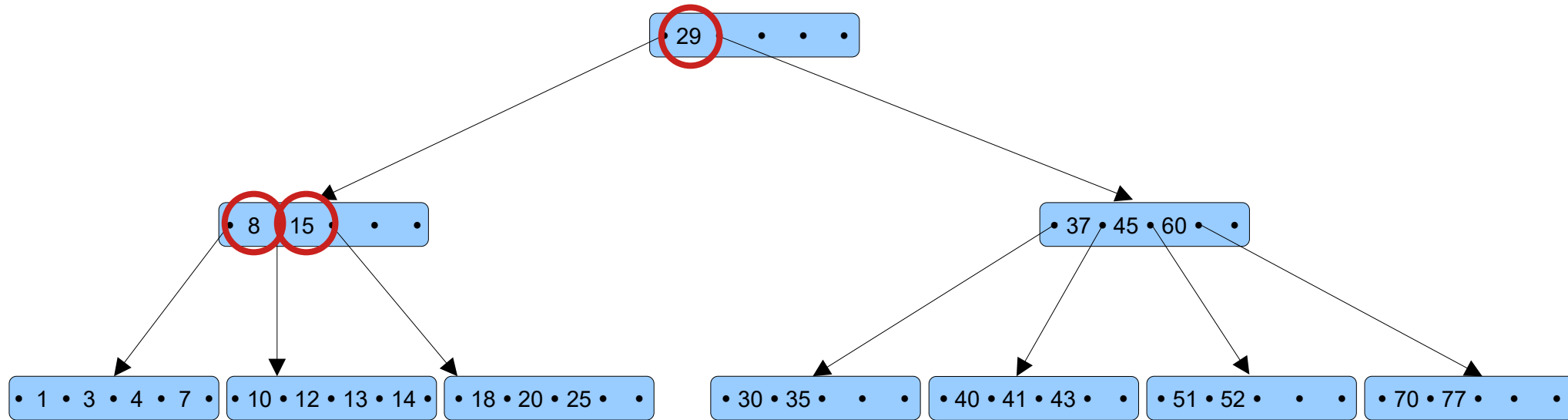
- Exemplo – Buscar a chave **25**



Árvore B

Busca

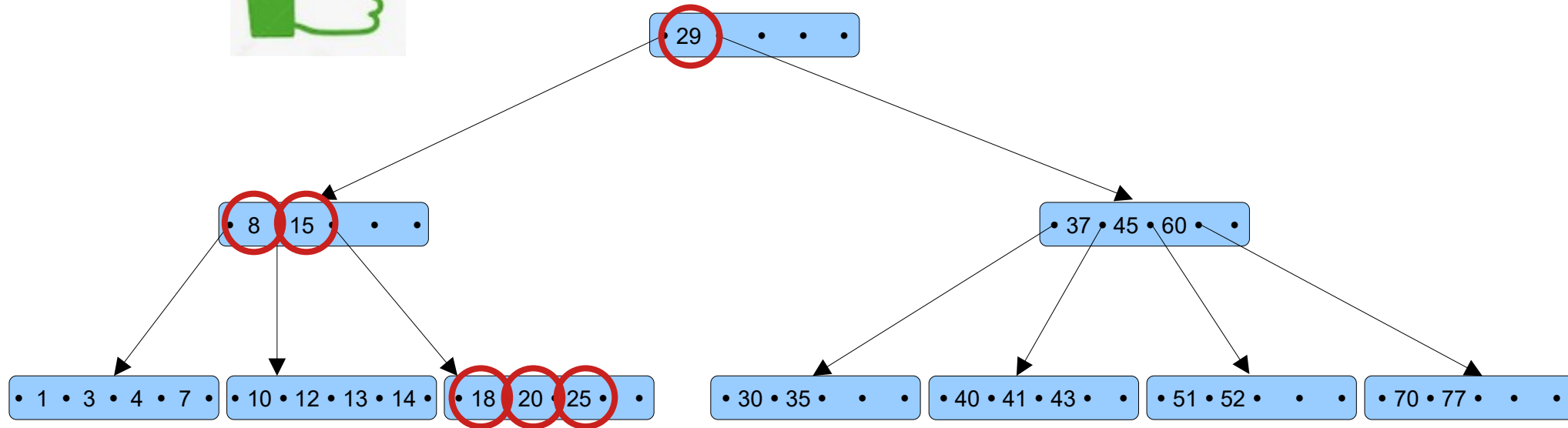
- Exemplo – Buscar a chave **25**



Árvore B

Busca

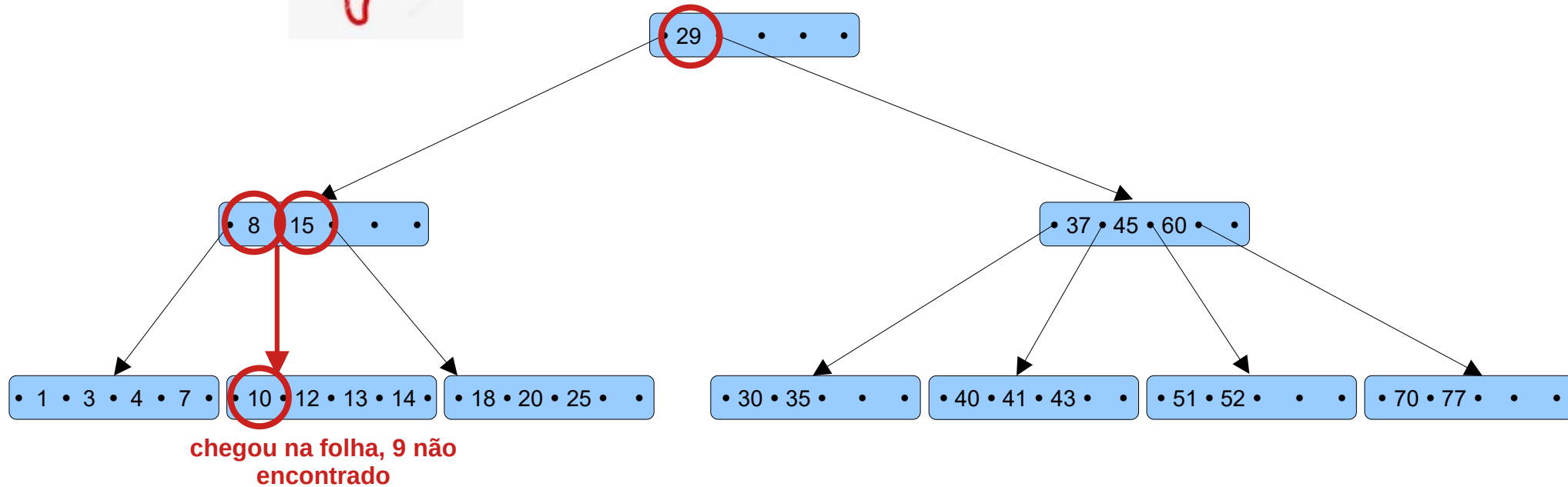
- Exemplo – Buscar a chave **25**



Árvore B

Busca

- Exemplo – Buscar a chave 9



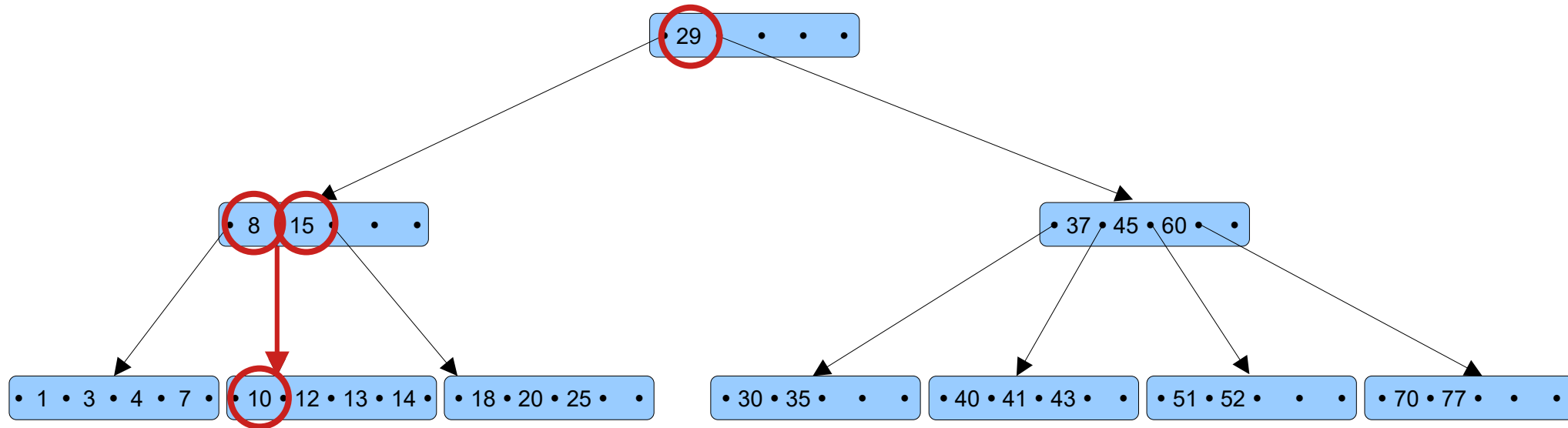
Operações em Árvores B

- **Exclusão: Caso 1**
 - Eliminação de uma chave em uma página folha, sendo que o número mínimo de chaves na página é respeitado.
 - 50% de ocupação
 - **Solução:** a chave é retirada e os registros internos à página são reorganizados.

Árvore B

Exclusão

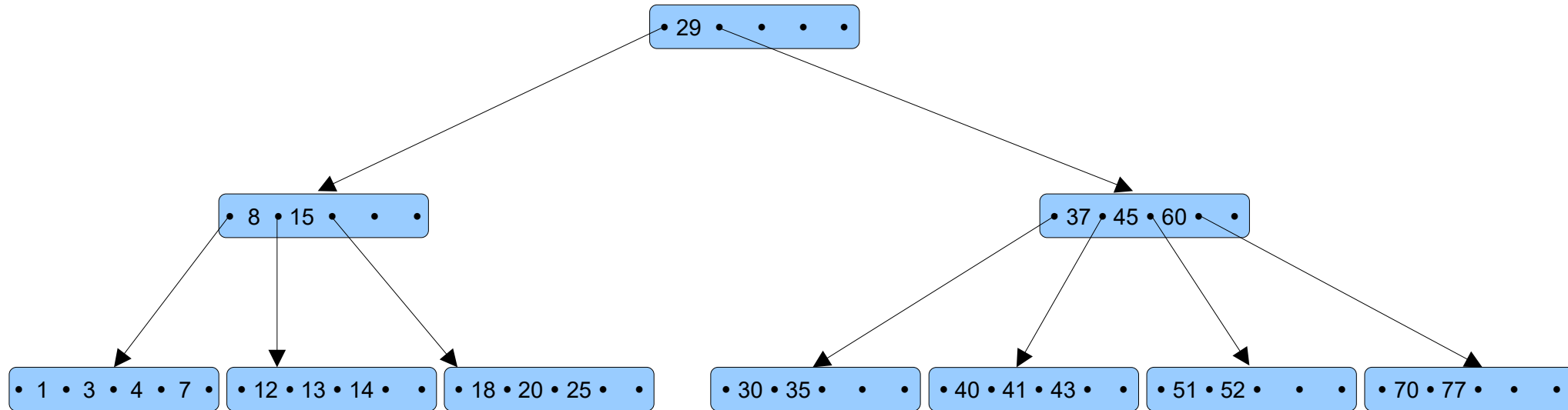
- **Caso 1 – Excluir a chave 10**



Árvore B

Exclusão

- **Caso 1 – Excluir a chave 10**



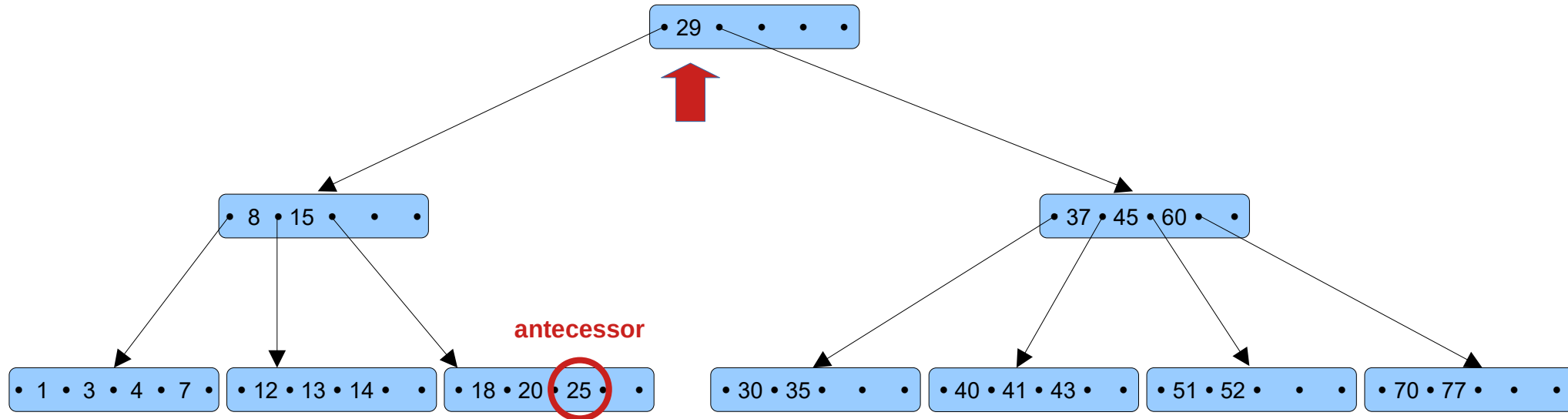
Operações em Árvores B

- **Exclusão: Caso 2**
 - Eliminação de uma chave que não está numa folha.
 - **Solução**: trocamos a chave com sua antecessora imediata e então eliminamos a mesma da folha.

Árvore B

Exclusão

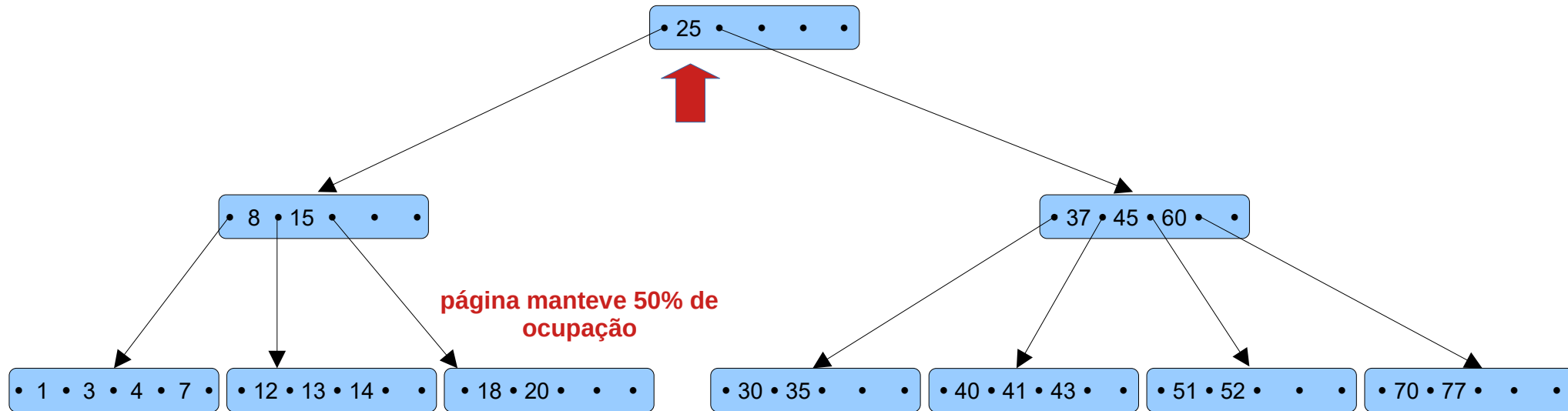
- Caso 2 – Excluir a chave **29**



Árvore B

Exclusão

- Caso 2 – Excluir a chave **29**



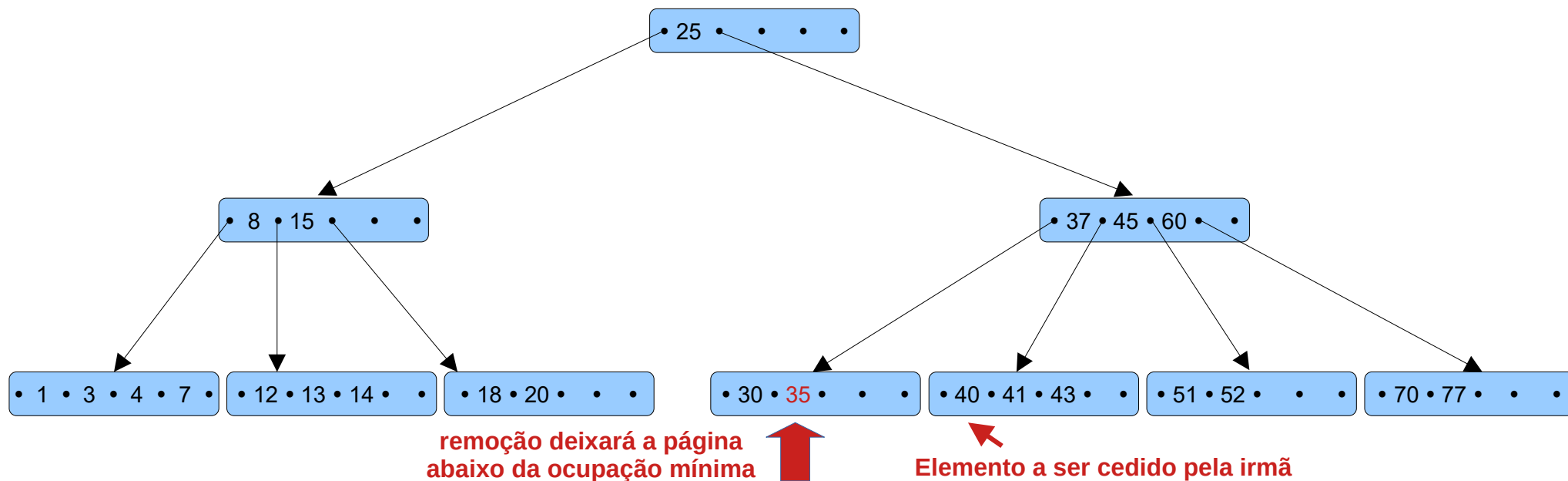
Operações em Árvores B

- **Exclusão: Caso 3**
 - Remoção causa *underflow* na folha, mas a página irmã pode ceder uma chave.
 - **Solução:** procura-se uma página irmã (mesmo pai) que contenha mais chaves do que o mínimo; se existir redistribui-se as chaves entre essas páginas.
 - A redistribuição pode provocar uma alteração na chave separadora que está no nó pai.

Árvore B

Exclusão

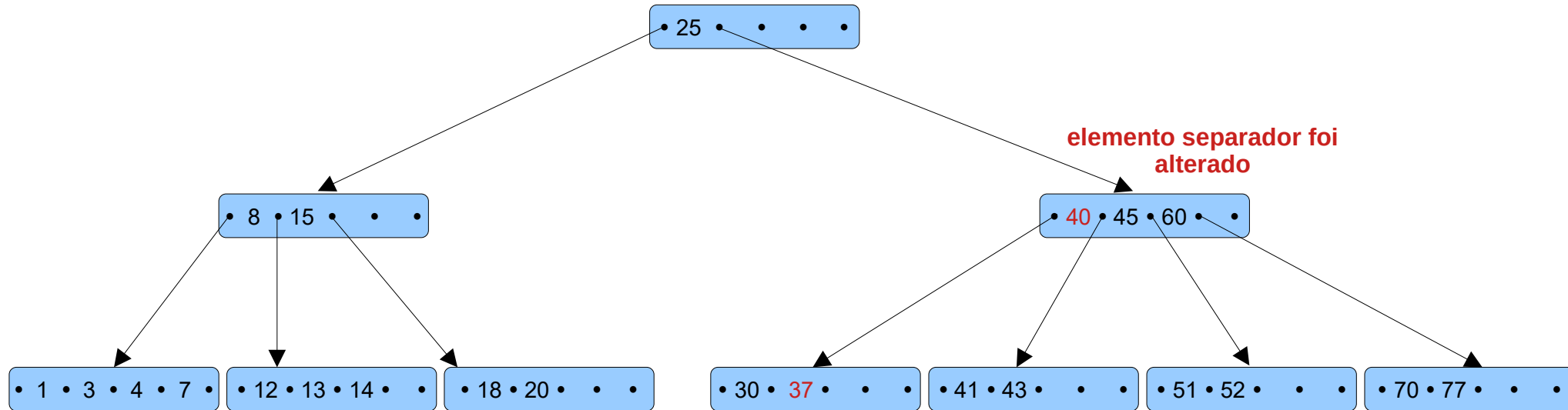
- **Caso 3 – Excluir a chave 35**



Árvore B

Exclusão

- **Caso 3 – Excluir a chave 35**



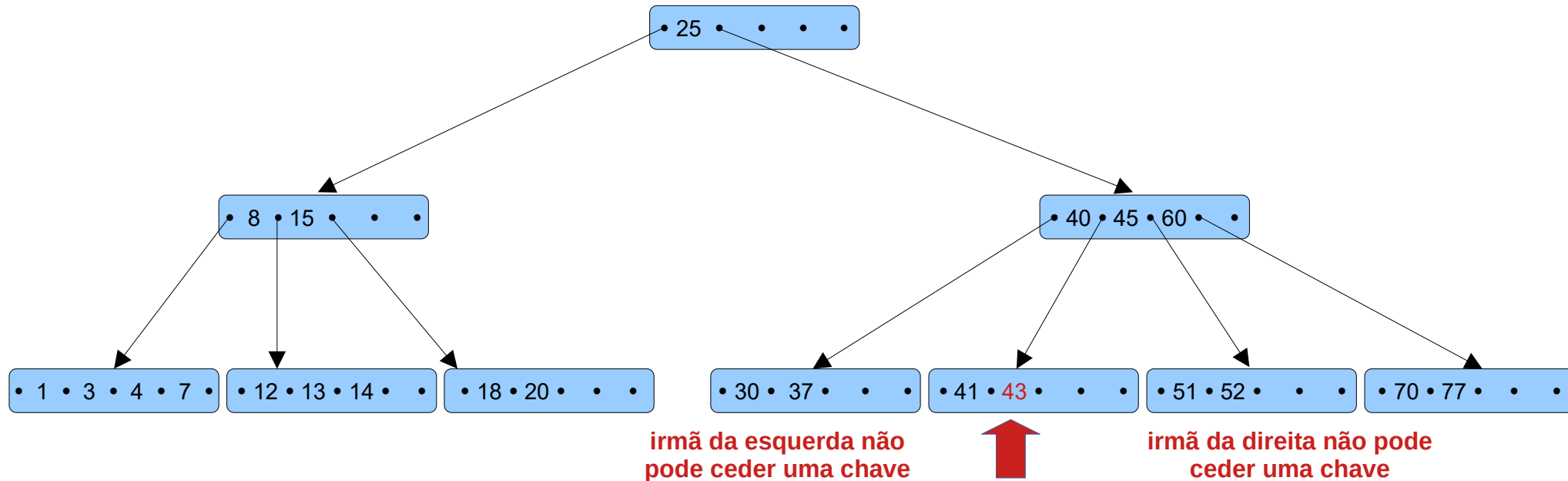
Operações em Árvores B

- **Exclusão: Caso 4**
 - Ocorre *underflow* e a redistribuição não pode ser aplicada, pois não existem chaves suficientes para dividir entre as duas páginas irmãs.
 - **Solução:** concatenar o conteúdo de duas páginas e a chave da página pai, formando uma única página.
 - Inverso do *split/promote*.
 - Pode provocar *underflow* da página pai.

Árvore B

Exclusão

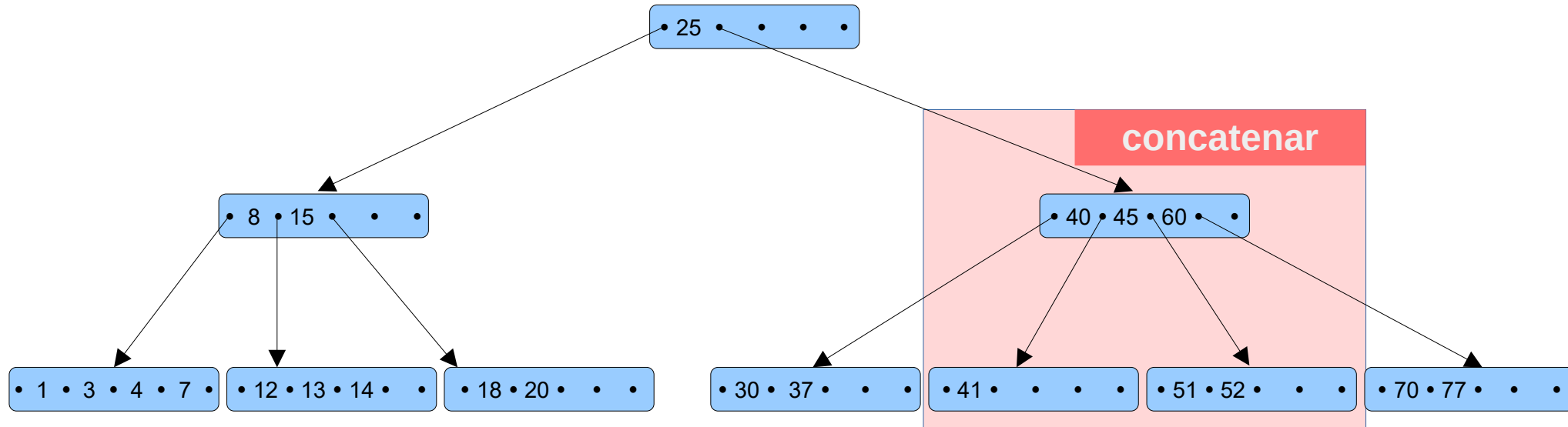
- Caso 4 – Excluir a chave **43**



Árvore B

Exclusão

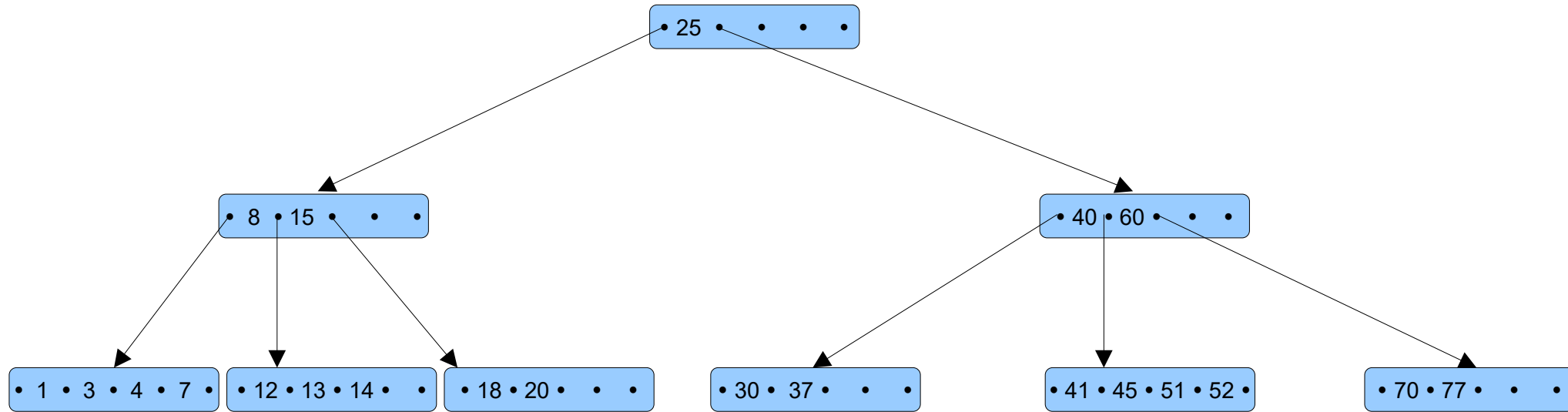
- **Caso 4 – Excluir a chave 43**



Árvore B

Exclusão

- **Caso 4 – Excluir a chave 43**



uniu 2 páginas + separador

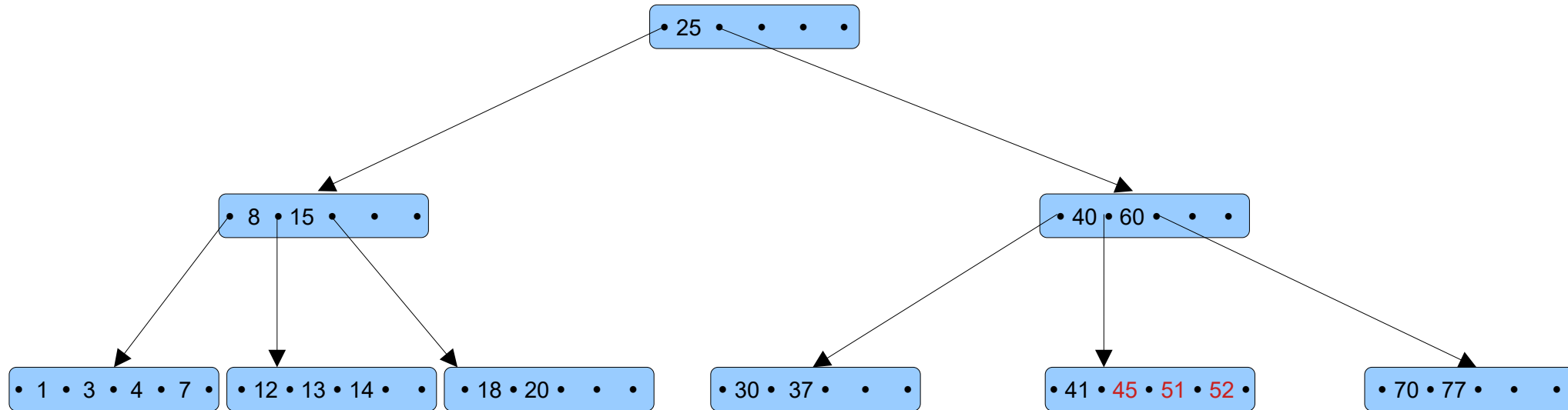
Operações em Árvores B

- **Exclusão: Caso 5**
 - *Underflow* da página pai sem possibilidade de redistribuição.
 - **Solução**: deve-se utilizar concatenação novamente. Pode levar à diminuição da altura da árvore.

Árvore B

Exclusão

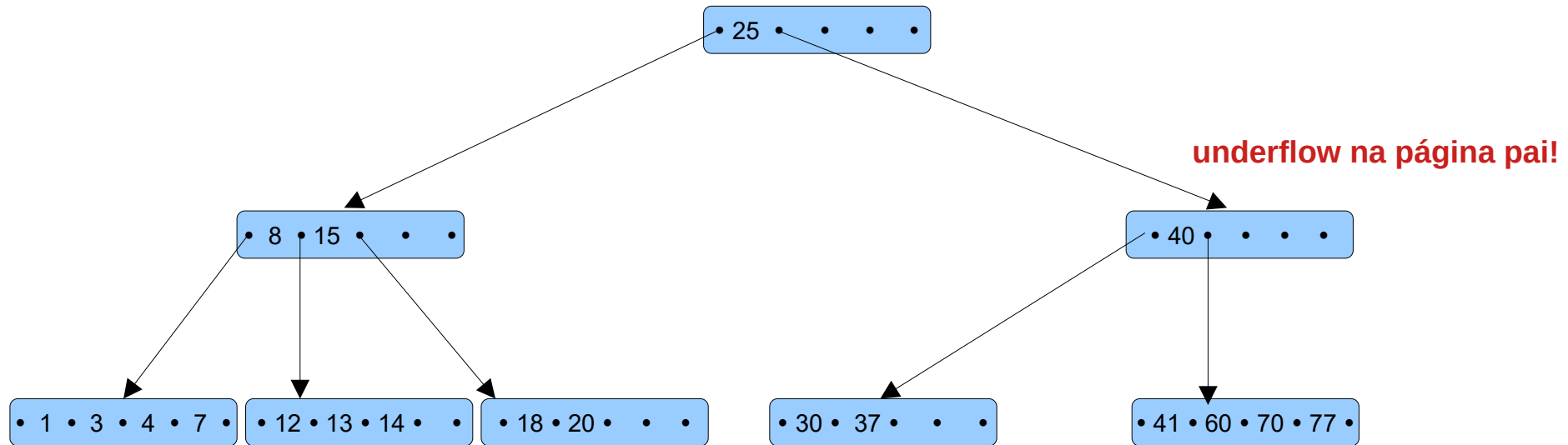
- Caso 5 – Excluir as chaves **52** (ok), **51** (ok), **45** (caso 4)



Árvore B

Exclusão

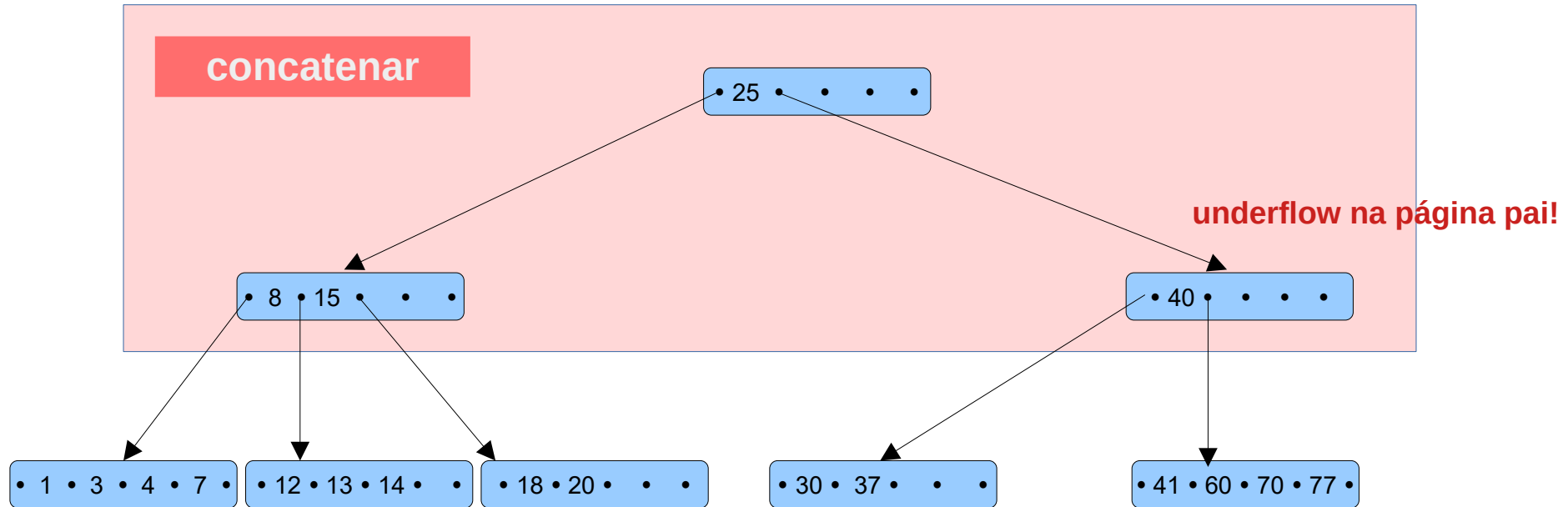
- Caso 5 – Excluir as chaves **52** (ok), **51** (ok), **45** (caso 4)



Árvore B

Exclusão

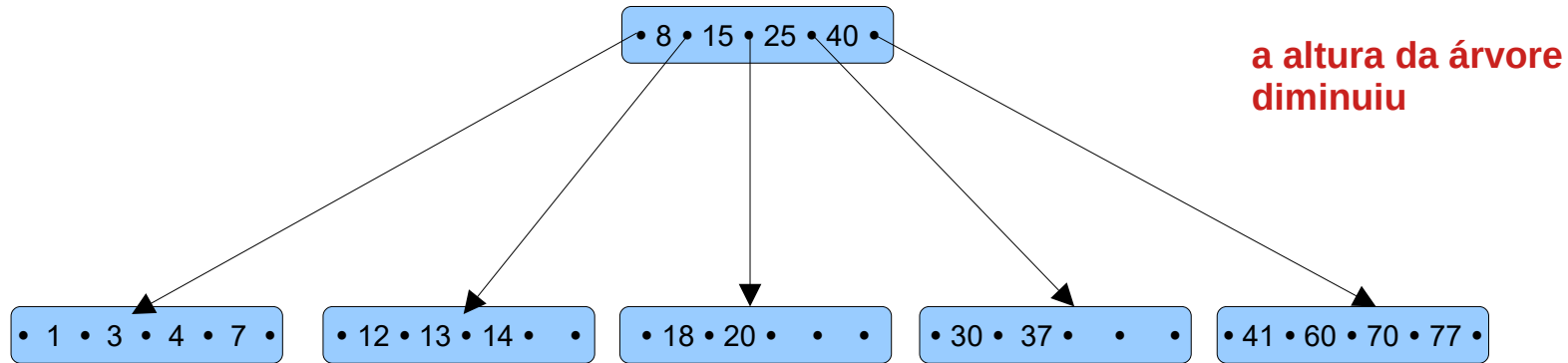
- Caso 5 – Excluir as chaves **52** (ok), **51** (ok), **45** (caso 4)



Árvore B

Exclusão

- Caso 5 – Excluir as chaves **52** (ok), **51** (ok), **45** (caso 4)



Altura de uma Árvore B

- Qual a altura máxima de uma árvore B com m chaves?
- Esta questão é importante pois a altura da árvore dará o limite máximo de acesso ao disco.

Altura de uma Árvore B

- Sendo N o maior número de chaves na árvore, N' o menor e m o número máximo de chaves em um nó

- Assim, a menor altura será
$$h = \log_m \left(\frac{N' + 1}{2} \right)$$

- A maior será
$$h = \log_{\frac{m}{2}} \left(\frac{N + 1}{2} \right)$$

Altura de uma Árvore B

- Exemplificando:
 - Para $N = 2.000.000$ (dois milhões) e $m = 200$, a altura será de apenas **3**
 - em uma árvore binária, seria mais de **20**

Visualização

- <https://www.cs.usfca.edu/~galles/visualization/BTree.html>