

Árvore B

Árvore B (ou B-Tree) é uma estrutura de apoio à organização e manutenção de grande quantidade de índices guardados em dispositivos de armazenamento secundários (ex. discos magnéticos), sendo um Método de Acesso de bom desempenho para pesquisas (sobre valores de chave primária) randômicas e ocasionalmente seqüenciais mesmo não necessitando de nenhuma reorganização.

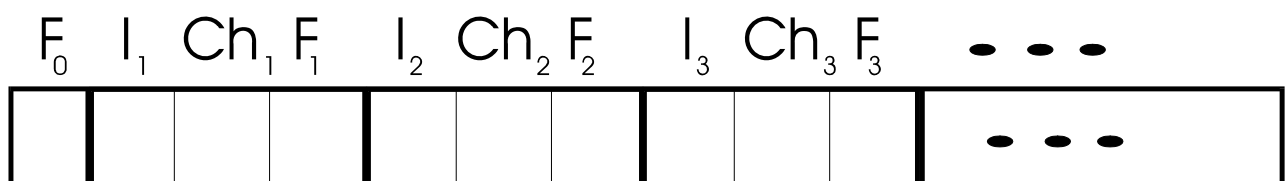
Esta estrutura foi inicialmente estudada em meados de 1970 por Bayer e McCreight tendo atualmente um grande aplicação em todos os SGBD Relacionais, com pequenas variações estruturais em cada fabricante.

VANTAGENS.

- O espaço de memória alocada para a Árvore B no dispositivo de armazenamento tem no mínimo 50 % de aproveitamento;
- Nenhuma degradação do desempenho da estrutura ocorre quando o aproveitamento da estrutura aumenta;
- O acesso randômico de um índice requer poucos passos (acessos à disco);
- Permite uma processamento seqüencial pois as chaves são mantidas organizadas/ordenadas lexograficamente;
- A Árvore se mantém balanceada para um eficiente acesso randômico.

CARACTERÍSTICAS.

Árvore B é um tipo especial de Árvore, cuja ordem K pode variar de 3 até um número bastante expressivo (ex. 30 ou 40). Os nós da Árvore B são homogêneos e se apresenta no seguinte formato:



onde

- F** é um ponteiro para um nó filho
I é o índice para o endereço de um registro no arquivo de dados
Ch é o valor de uma chave primária

A ordem de uma Árvore B define a quantidade de conjuntos (I, Ch, F) dos nós da árvore. Todo valor de chave primária (**Ch**) tem um ponteiro para um nó filho (**F**) correspondente e um endereço de registro (**I**) também correspondente. A exceção é **F₀** que não possui chave e índice correspondente.

Na organização lógica da Árvore B :

- os valores de chave contidos nos nós filhos são sempre maiores que o valor de chave correspondente no nó pai;
- os valores de chaves contidos no nó filho de **F₀** são sempre menores que **Ch₁**;
- o comprimento do caminho da Raiz da árvore até qualquer nó folha é sempre o mesmo e equivale ao tamanho da árvore (altura - h);
- todo nó, exceto a Raiz, tem no mínimo $k/2$ conjuntos utilizados;
- as chaves em um nó estão ordenadas de forma crescente (para permitir a pesquisa binária).

Desta forma, temos que a capacidade de indexação de uma Árvore B pode ser visualizada pela tabela abaixo:

Nível	Número de Nós	Número de Chaves
1	1	k
2	k+1	k * (k+1)
3	$(k+1)^2$	$k * (k+1)^2$
4	$(k+1)^3$	$k * (k+1)^3$
..		
..		
h	$(k+1)^{h-1}$	$k * (k+1)^{h-1}$

Exemplos:

(1)

Uma Árvore B com $K = 10$ terá em 4 níveis a capacidade de armazenar $10+(10*(10+1))+(10*(10+1)^2)+(10*(10+1)^3)$, ou seja, $10+110+1.210+13.310 = 14.640$ chaves correspondendo a 14.640 registros no arquivo de dados respectivo.

(2)

Uma Árvore B com $K = 30$ terá em 5 níveis a capacidade de armazenar $30+(30*(30+1))+(30*(30+1)^2)+(30*(30+1)^3)+(30*(30+1)^4)$, ou seja, $30+930+28.830+893.730+27.705.630 = 28.629.150$ chaves correspondendo aos registros no arquivo de dados respectivo.

A capacidade de armazenamento de uma Árvore B (mesmo de pouca altura) é muito vantajosa porém deve-se ponderar entre valores de K “pequenos” ou “grandes” em relação ao arquivo de dados no qual pretende-se aplicar esta estrutura.

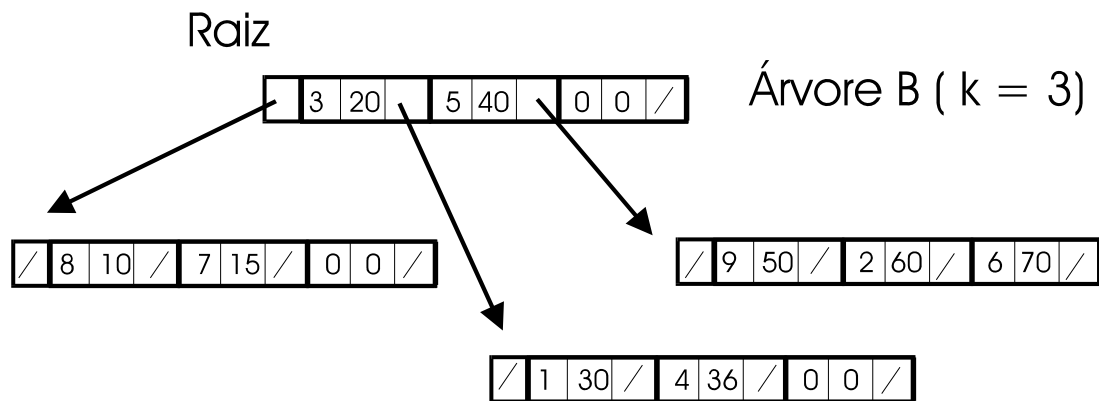
Exemplo de Árvore B:

Arquivo da B-Tree

1	0	1	30	0	4	36	0	0	0	0
2	0	9	50	0	2	60	0	6	70	0
3	0	8	10	0	7	15	0	0	0	0
4	3	3	20	1	5	40	2	0	0	0

Arquivo de Dados

	Cod	Nome	Empresa	Idade
1	30	a	b	31
2	60	c	d	23
3	20	e	f	18
4	36	g	h	25
5	40	i	j	30
6	70	k	l	28
7	15	m	n	21
8	10	o	p	23
9	50	q	r	31



ALGORITMO DE INSERÇÃO.

Caso 1 (Simple):

Inserção de uma chave (Ch) e seu respectivo índice (I) em um correspondente nó folha com disponibilidade para mais esta chave. A reorganização interna do nó que recebe esta nova chave pode ser necessária para que a sequência de chave do nó permaneça ordenada de forma crescente. Exemplo: Inserção de uma chave 38 cujo registro no arquivo de dados é 10.

Caso 2 ():

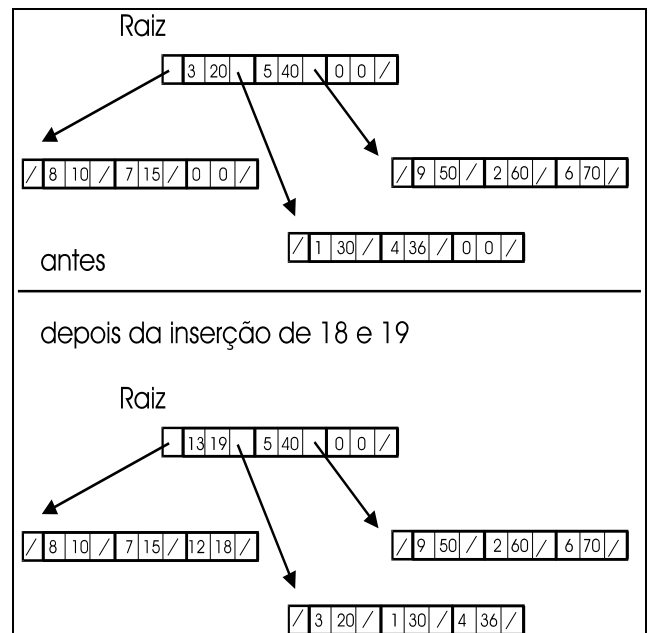
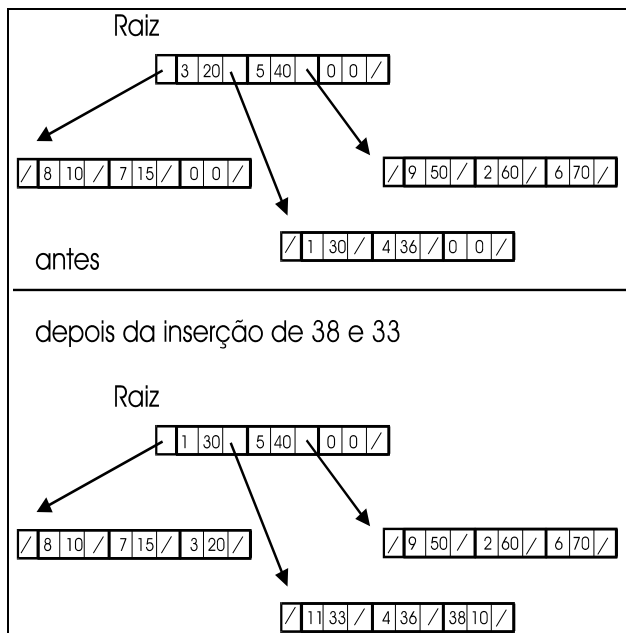
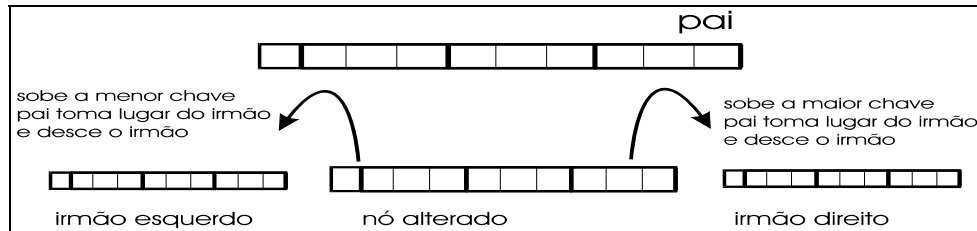
Inserção de uma chave (Ch) e seu respectivo índice (I) em um correspondente nó folha “cheio”, ou seja sem disponibilidade para mais esta chave.

Opção 1 (Shift Left):

Caso o irmão esquerdo do nó correspondente tenha ao menos um conjunto livre, então desloca-se a menor chave do nó para o correspondente pai e a chave do pai desce para o filho de seu irmão esquerdo, de modo a manter a ordenação interna dos nós envolvidos.

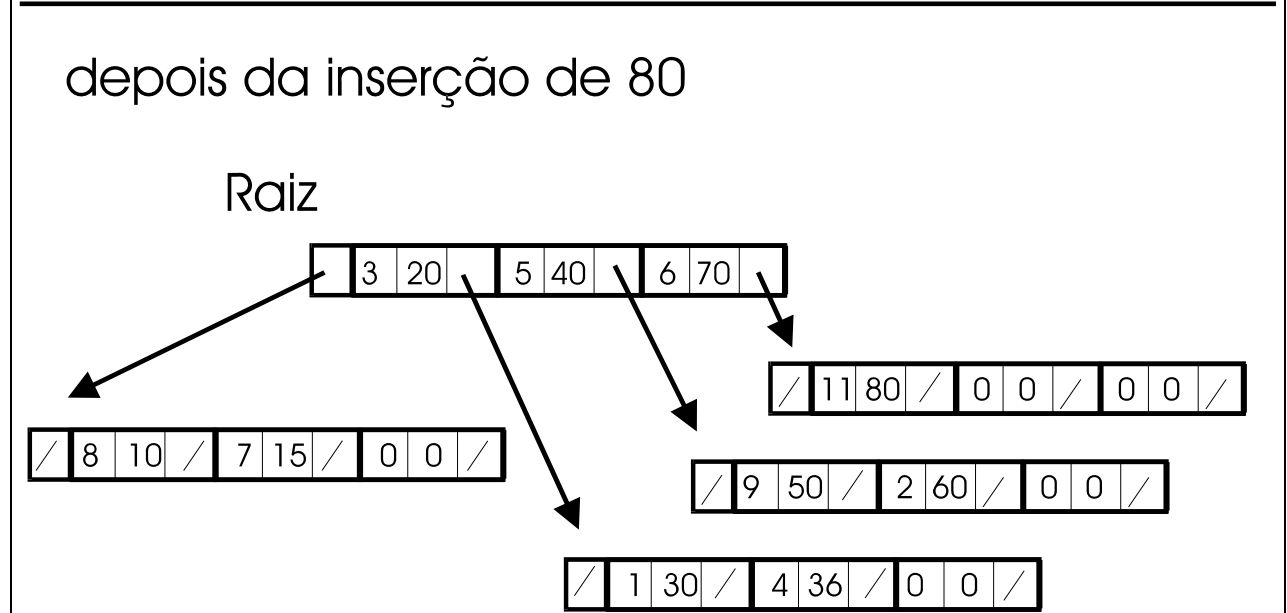
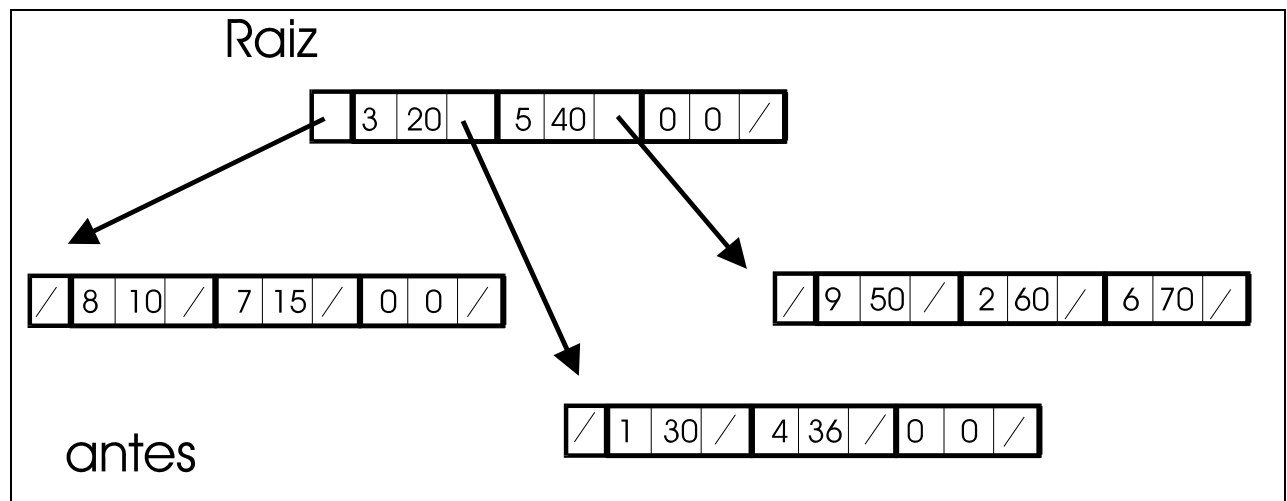
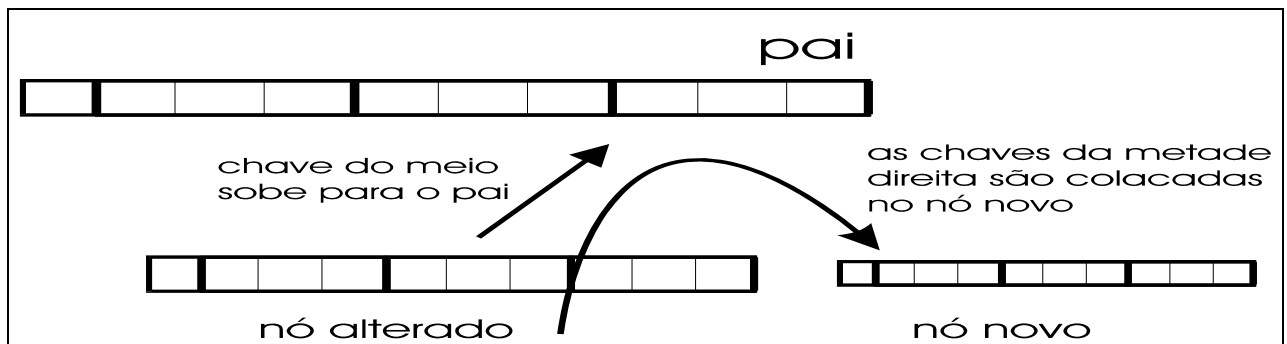
Opção 2 (Shift Right):

Caso o irmão direito do nó correspondente tenha ao menos um conjunto livre, então desloca-se a maior chave do nó para o irmão direito do pai e a chave do irmão direito do pai desce para o filho correspondente, de modo a manter a ordenação interna dos nós envolvidos.



Opção 3 (Split):

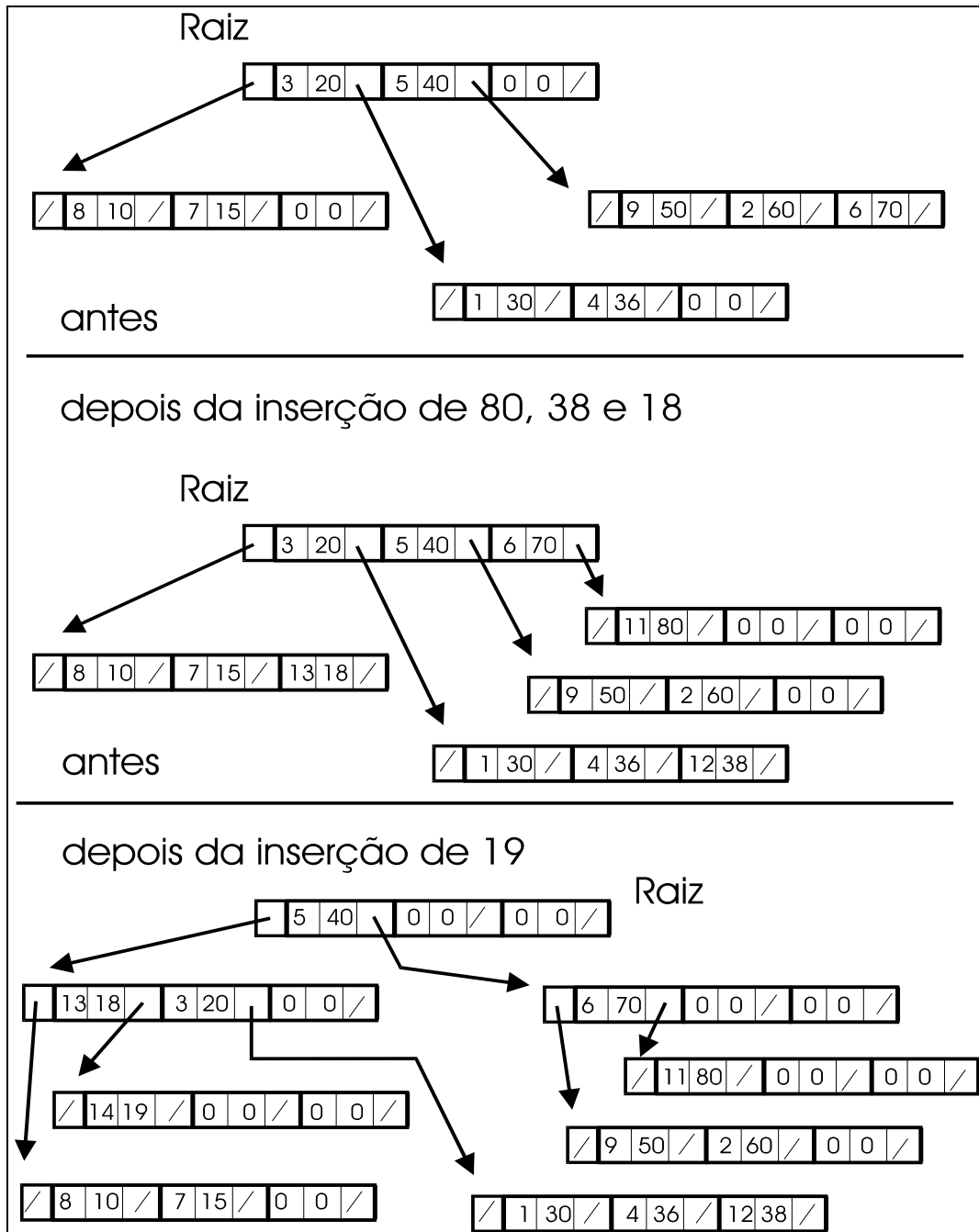
Caso nenhum dos irmãos tenha conjuntos vazio (que permita o “shift”), então cria-se um novo nó que terá a metade direita de chaves do nó “cheio”. A Chave posicionada no meio da seqüência de chaves do nó repartido é elevada para o nó pai.



Opção 4 (Split with New Root):

Caso nenhum dos irmãos tenha conjuntos vazios (que permita o “shift”), então cria-se um novo nó que terá a metade direita de chaves do nó “cheio”. A Chave posicionada no meio da sequência de chaves do nó repartido é elevada para o nó pai. Se o nó pai não

existe ou também não comporta uma chave nova e seu irmãos direito e esquerdo não têm conjunto disponíveis, então cria-se uma nova raiz para a árvore (ou sub-árvore).



ALGORITMO DE REMOÇÃO.

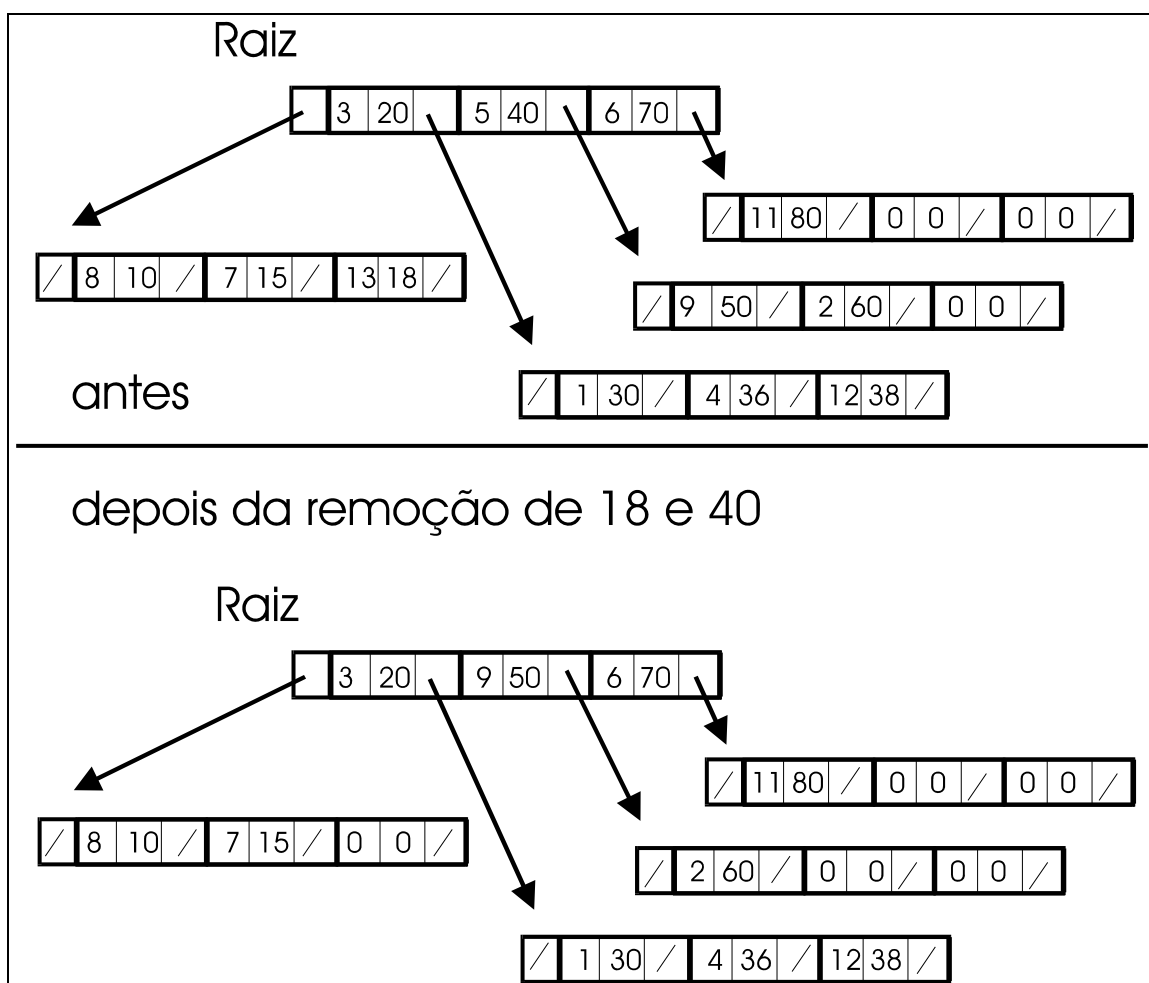
Caso 1 (Simple):

Remoção de uma chave (Ch) e seu respectivo índice (I) em um correspondente nó folha com mais de $K/2$ chaves (Leaf node no underflow). A reorganização interna

do nó que recebe esta nova chave pode ser necessária para que a sequência de chave do nó permaneça ordenada de forma crescente. Exemplo: Inserção de uma chave 38 cujo registro no arquivo de dados é 12.

Caso 2 (Replacement):

Remoção de uma chave (Ch) e seu respectivo índice (I) em um correspondente nó não folha com mais de $K/2$ chaves (Branch node no underflow). No lugar da chave (e índice) removido coloca-se a menor chave e seu índice da subárvore direita, ou então a maior chave e seu índice da subárvore esquerda. Conseqüentemente, a chave copiada (que pertence a um nó folha) deve ser removida, o que leva a resolução de um dos outros casos.



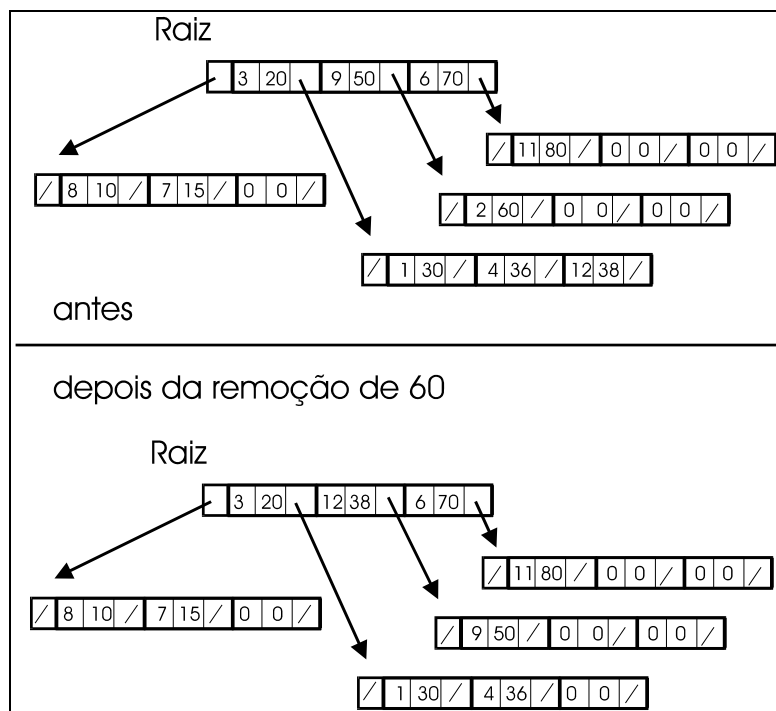
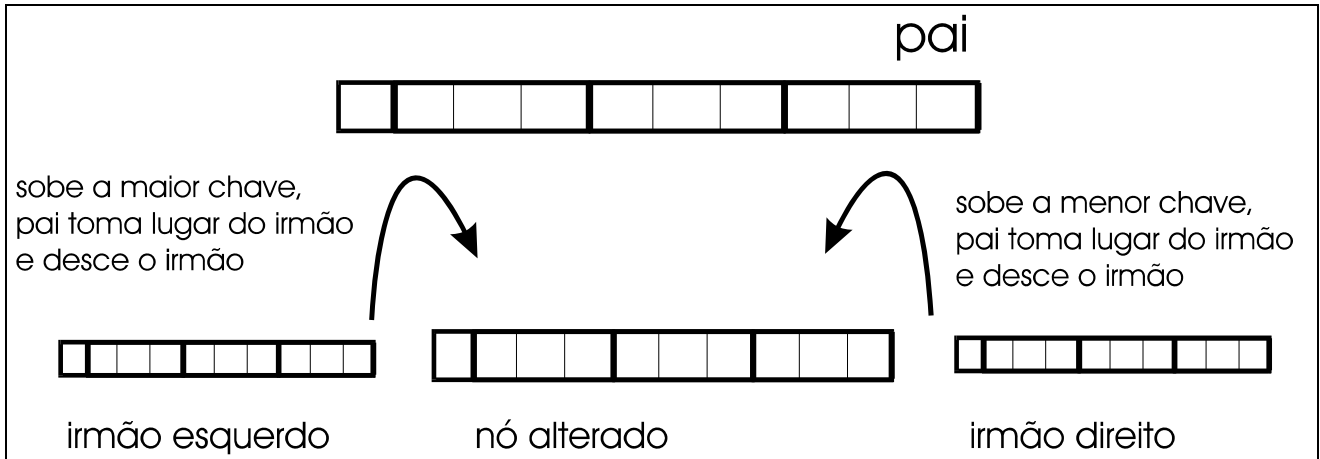
Caso 3 :

Opção 1 (Left Borrow):

Remoção de uma chave (Ch) e seu respectivo índice (I) em um correspondente nó folha com de $K/2$ chaves (Leaf node with underflow), então desloca-se a maior chave do irmão esquerdo (caso este irmão possa emprestar) para o correspondente pai e a chave do pai desce para o nó correspondente, de modo a manter a ordenação interna dos nós envolvidos.

Opção 2 (Right Borrow):

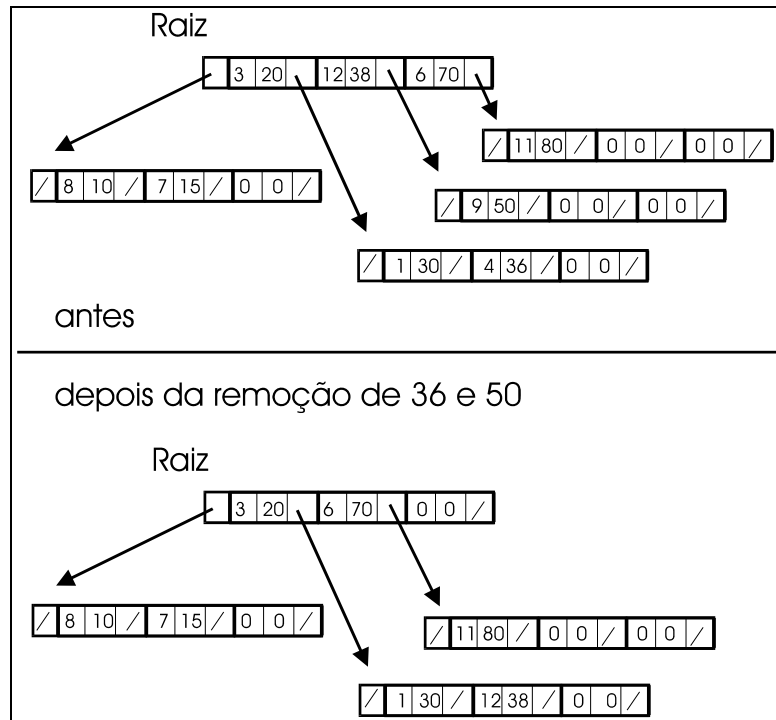
Remoção de uma chave (Ch) e seu respectivo índice (I) em um correspondente nó folha com de $K/2$ chaves (Leaf node with underflow), então desloca-se a menor chave do irmão direito (caso este irmão possa emprestar) para o correspondente pai e a chave do pai desce para o nó correspondente, de modo a manter a ordenação interna dos nós envolvidos.



Caso 4 (Concatenate):

Quando o irmão esquerdo e direito não podem emprestar então realiza-se a concatenação das chaves do nó esquerdo ou direito (quando existir) de modo que as chaves fiquem organizadas em apenas um nó. A chave (no nó pai) cujo filho foi concatenado também é incluído na concatenação e isto pode causar o “underflow” do nó pai que deve ser resolvido também com uma

concatenação (isto pode ocorrer recursivamente até a raiz da árvore quando então a concatenação termina).



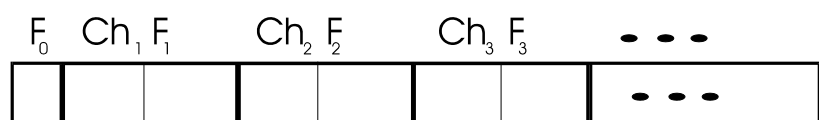
ÁRVORE

B^* (OU B^+).

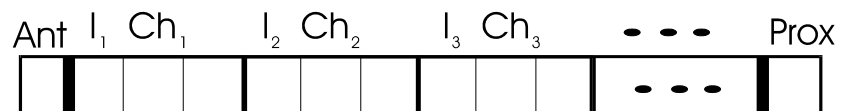
Uma das mais conhecidas variações da Árvore B que se caracteriza pela existência de todas as chaves e seus índices apenas nos nós folhas, enquanto os nós internos possuem apenas algumas chaves para reconhecimento de desvio e os respectivos ponteiro para seus filhos. Desta forma algumas chave aparecem repetidas na estrutura.

Existem dois tipos de nós:

Nó Desvio



Nó Informação:



Os nós de Desvio armazenam apenas as chaves indicativas do menor valor contido em sua respectiva subárvore. Os nós de informação estabelecem uma seqüência (lista encadeada - dupla ou simples) para apoiar as pesquisa seqüencial. Os dois conjuntos de nós podem ser armazenados fisicamente em um mesmo arquivo ou em arquivos distintos.

