Inserção:

* Uma árvore B exige que todas as folhas tenham a mesma altura
* Processo de "rotação" implica em: dividir uma página cheia em duas e ceder um elemento para a página mãe (a chave do meio).
* Começamos o processo pela raiz da árvore, se ela estiver cheia (*x.n == 2t - 1*), procedemos com a divisão da raiz:
* criamos um novo nó raiz, que receberá a chave do centro da página a ser dividida,
* a antiga raiz se torna o filho da esquerda dessa chave
* em seguida, criamos uma nova página, para quem copiamos as chaves posteriores ao elemento do meio e seus descendentes, se houver.
* Podemos também observar que a altura da árvore B somente aumenta pela divisão da raiz,
* Depois de tratarmos a raiz (ou caso ela não esteja cheia), buscamos o filho *x.ci* (sendo *x* a raiz) em que a chave deverá ficar.
* Trazemos esse filho do disco para a memória, antes de iniciar a recursão, para verificar se ele está cheio.
* Se estiver cheio, fazemos a divisão desse filho, subindo uma chave e criando dois descendentes com *t - 1* chaves.
* Seguimos esse procedimento, descendo até chegarmos na folha que irá comportar a nova chave, onde faremos a inserção, mantendo a ordem crescente dentro da página.

#### Remoção:

* A remoção de chaves de uma árvore B pode ser feita em qualquer página, não somente nas folhas, de tal forma que teremos que reorganizar os filhos quando uma chave é removida.
* Vamos garantir que nenhuma página tenha menos que o mínimo (*t* - 1) chaves.
* Na inserção, dividíamos páginas cheias para garantir que as operações pudessem ocorrer sem precisar fazer *backtracking* na árvore, logo, teremos o inverso na remoção:
* vamos fundir páginas ou pegar chaves emprestadas de outras, para garantir que, no caminho da remoção, as páginas tenham, pelo menos, *t* chaves, ou seja, 1 chave a mais que o mínimo, para casos em que essas movimentações sejam necessárias e garantir que nenhuma fique abaixo do mínimo exigido.
* caso a raiz tenha apenas uma chave e seja necessário fazer a fusão dos seus dois filhos, esta chave ocupará a posição mediana do filho da esquerda, todas as chaves (e eventuais descendentes) do filho da direita serão copiados para depois da mediana e essa página se tornará a nova raiz.
* Diminuindo a altura da árvore diminui em 1 unidade quando fazemos a fusão da raiz com seus dois descendentes, mantendo a propriedade de que o crescimento ou retração da árvore ocorre sempre na raiz.
* Se a chave *k* está em uma página folha *x*, remova *k* de *x*.
* Se a chave *k* está em uma página interna *x*, faça:
* Se o filho *y* que precede *k* na página *x* tem, no mínimo, *t* chaves, então encontre o predecessor *k'* de *k* na sub-árvore enraizada em *y* (*k'* é o maior dos menores). Recursivamente, remova *k'* e troque *k* por *k'* em *x* (podemos encontrar e remover *k'* em uma única passagem pela árvore).
* Se *y* tem menos que *t* chaves, então examine, de forma simétrica, o filho *z* que segue *k* na página *x*. Se *z* tem pelo menos *t* chaves, então encontre o sucessor *k'* de *k* na sub-árvore enraizada em *z*(*k'* é o menor dos maiores). Recursivamente, remova *k'* e troque *k* por *k'* em *x* (podemos encontrar e remover *k'* em uma única passagem pela árvore).
* Caso contrário, se ambos *y* e *z* possuírem apenas *t* - 1 chaves, faça a fusão de *k* e todo o conteúdo de *z* em *y*, de tal forma que *x* perde a chave *k* e o ponteiro para *z*, e *y* agora passa a conter 2*t* - 1 chaves. Então desaloque *z* e recursivamente remova *k* de *y*.
* Se a chave *k* não está presente na página interna *x*, determine a raiz *x.ci* da sub-árvore apropriada que deve conter *k*, se *k* existir na árvore. Se *x.ci* tem apenas *t* - 1 chaves, execute o passo 3a ou 3b conforme necessário para garantir que vamos descer para uma página contendo, no mínimo, *t* chaves. Então finalize fazendo a recursão no filho apropriado de *x*.
* Se *x.ci* tem apenas *t* - 1 chaves mas tem um irmão imediato com, no mínimo, *t* chaves, dê a *x.ci* uma chave extra trocando uma chave de *x* para *x.ci*, e movendo a chave do irmão de *x.ci* imediatamente à esquerda ou à direita para *x*, e movendo o ponteiro apropriado dos descendentes desse irmão *x.ci* para *x.c*i. Inicie pelo irmão da esquerda, caso ele não tenha como emprestar essa chave, analise o irmão da direita.
* Se *x.ci* e ambos os irmãos imediatos possuem *t* - 1 chaves, faça a fusão de *x.ci* com um dos irmãos (preferencialmente o da esquerda), movendo uma chave de *x* para o nó em que as chaves serão agrupadas, para se tornar a mediana daquele nó, e todos os ponteiros dos descendentes, se houver.