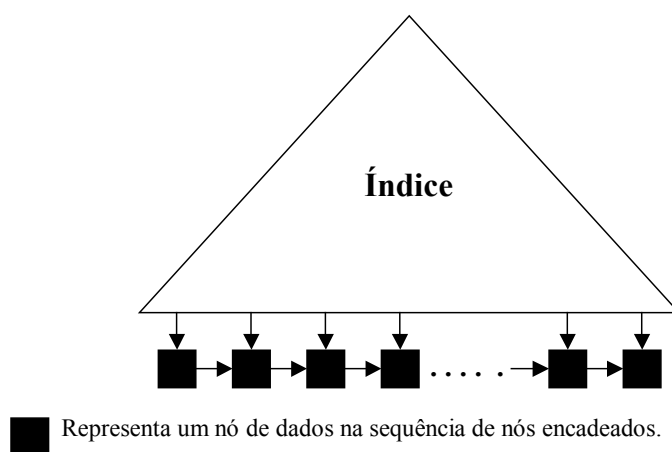
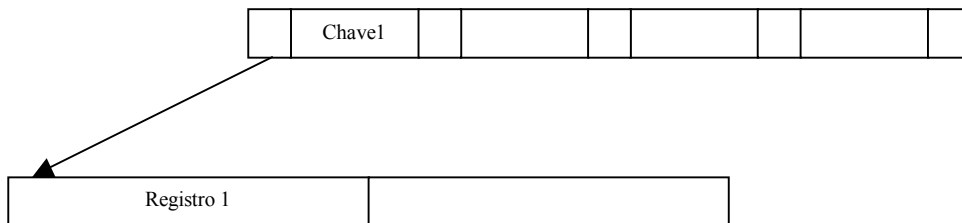


Árvore B+

- ◆ Variante da árvore B.
- ◆ Os nós não folha armazenam as chaves dos registros e não os registros inteiros.
 - Em nó interno de uma árvore B+ cabe mais registro do que o de uma árvore B.
 - Consequentemente a altura da árvore B+ é menor do que a da árvore B.
- ◆ Os registros são armazenados em nós folha.
- ◆ Nó não folha age como se fossem uma estrutura de índice para nós que contém os dados.
- ◆ Diferentemente da árvore B, a árvore B+ tem a estrutura de índices separada da estrutura de dados.
- ◆ Para acessar qualquer registro é necessário o mesmo número de acessos, já que todos ficam no mesmo nível.
- ◆ Vantagens:
 - Na maioria das vezes a árvore B+ tem altura menor do que a árvore B para os mesmos registros, o que significa que menos operações de E/S são necessárias durante a busca.
 - Não é necessário subir e descer na árvore para acessar sequencialmente os registros.
 - Tanto o acesso direto como o acesso sequencial são melhores que na árvore B.
- ◆ Definição: Uma árvore B+ é uma árvore B na qual os registros dados são armazenados em nós folha e os nós não folha armazenam apenas os valores das chaves, formando um índice para os nós de dados.
- ◆ Melhoria no acesso sequencial é obtida interligando os nós de dados. Como resultado não é necessário mover através da árvore mas apenas processar uma lista encadeada formada pelos nós não folha.
- ◆ Pela própria definição da árvore B, os registros dos nós folha estão automaticamente ordenados na lista encadeada.



- ◆ Dado que os registros não mais armazenados nos nós não folha, o procedimento de busca é modificado a fim de que, numa comparação igual, a busca prossiga à esquerda ou à direita para localizar o nó folha apropriado. Arbitrariamente, escolheremos ir à esquerda – o que é importante ser definido a priori.
- ◆ As principais diferenças entre a árvore B e a árvore B+ está **na inserção em um nó folha e no que acontece quando ocorre overflow.**
- ◆ Quando ocorre overflow, realiza-se a divisão do nó, mas apenas a chave do registro do meio é promovida para o nível mais alto.
- ◆ Como nós folha e nós de índice têm funções diferentes em uma árvore B+, a inserção do primeiro registro de uma árvore B+ é um caso especial.
 - O registro é armazenado em um nó folha e sua chave é armazenada em um nó de índice, com um ponteiro para o nó folha, por exemplo:



Algoritmo de Inserção em árvore B+

1. Navegar nos nós não folha da árvore B+ com um índice a fim de localizar o nó folha apropriado para inserir um novo registro. Se \leq , seguir à esquerda, se $>$ e existe outro registro à direita, repetir a comparação anterior (de \leq), senão seguir a ramificação da direita.
/*Ao encontrar o nó folha*/
2. SE existir espaço disponível ENTÃO
 - 2.1. Inserir o registro (de forma ordenada).
3. SENÃO
 - 3.1. Dividir o nó folha onde ocorreu overflow.
 - 3.1.1. Subir a chave do registro do meio para a estrutura de índice. O registro do meio é escolhido entre os registros do nó que sofreu overflow e o registro sendo inserido, considerando-os de forma ordenada.
 - 3.1.2. Dividir os registros do nó que sofreu overflow, incluindo o registro sendo inserido, entre dois nós tal que todos os registros com chave menores do que a chave do registro sendo inserido e o registro sendo inserido sejam armazenados no nó da esquerda. Os demais são armazenados no nó da direita.
 - 3.2. Processar a chave que foi recentemente elevada para a estrutura de índice como uma inserção em uma árvore B.

Exercício 1

Em árvore B+ com ordem de capacidade do índice igual a 2 e ordem dos nós dos dados (folhas) igual a 1 inserir as seguintes chaves:

80, 50, 100, 90, 60, 65, 70, 75, 55, 64, 51, 76, 77, 78, 200, 300 e 150

Exercício 2

Qual o caminho percorrido para buscar o 65?

- Comparar com 65 → igual → seguir à esquerda
- Comparar com 51 → maior → continuar à direita
- Comparar com 55 → maior → continuar à direita
- Comparar com 60 → maior e último registro da direita → seguir ramificação da direita
- Comparar com 64 → maior → continuar à direita
- Comparar com 65 → igual e é nó folha → achou o registro!!!!

Árvore B+

- ♦ Se o índice é pequeno o suficiente para caber inteiramente na memória principal, será necessário apenas um acesso à memória secundária para buscar um registro.
- ♦ Se o índice não couber inteiramente em memória principal, já armazenando alguns níveis inteiros será obtido ganho de performance, pois estes são acessados em toda busca.
- ♦ Os nós folhas podem ser duplamente encadeados permitindo busca ordenadas de forma ascendente e descendente.

Remoção em árvore B+

- ♦ Em uma árvore B+ o índice é usado com propósito navegacional e as folhas (também chamado de conjunto sequencial) são usadas para armazenar os dados.
- ♦ Assim, não é necessário remover uma chave do índice sempre que um registro é removido da folha. Mesmo que a chave do registro removido exista no índice.
- ♦ Uma chave só é removida do índice quando nós das folhas são combinadas (merge) devido à violação da restrição de capacidade mínima.