



**Ministério da Educação
Instituto Federal de Mato Grosso
Campus Cuiabá – Cel. Octayde Jorge da Silva
Departamento de Computação (DECOM)**

ÁRVORE B

**Discentes: Joicy Kelly e Mateus Goulart
Docente: João Paulo Preti**



Agenda

1. Contexto e Conceito
2. O que é uma árvore B
3. Nó ou Página
4. Operações Básicas
5. Código/Conclusão
6. Referência Bibliográfica



Contexto e Conceito

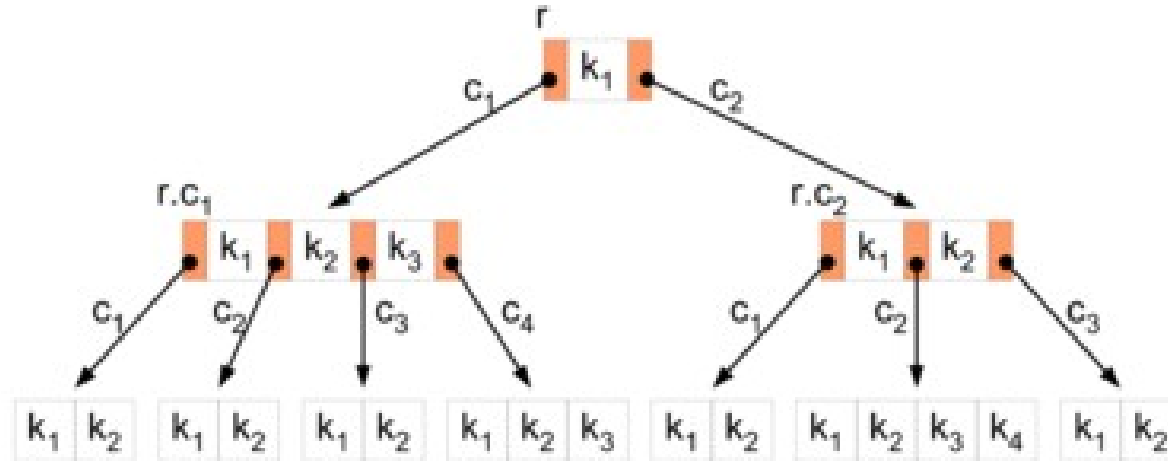


- Inventada por Rudolf Bayer e Edward Meyes McCreight em 1971
- Boeing Scientific Research Labs
- Origem do nome desconhecido
- Generalização das árvores binárias de busca
- É normalmente usada em bancos de dados e sistemas de arquivos
- foi projetada para funcionar especialmente em memória secundária como um disco magnético ou outros dispositivos de armazenamento secundário.
- Voltada para arquivos volumosos
- Proporciona rápido acesso ao dados.



O que é uma Árvore B ?

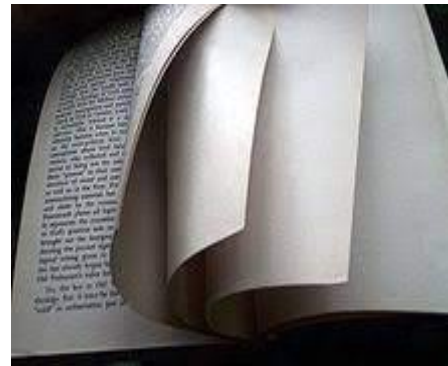
A árvore B é uma árvore balanceada em que o número de nós acessados em uma busca/inserção/remoção é muito pequeno se comparado às outras árvores estudadas. Outro ponto importante é que a árvore B garante que as folhas se encontrem todas em um mesmo nível, independente da ordem em que os dados serão inseridos.





Nó ou Página

Um nó ou página, geralmente é representado por um conjunto de elementos apontando para seus filhos. Alguns autores consideram a ordem de uma árvore B como sendo a quantidade de registros que a página pode suportar. Outros consideram a ordem como a quantidade de campos apontadores. Todo nó da árvore tem um mínimo de registros definido pela metade da ordem, arredondando-se para baixo, caso a árvore seja de ordem ímpar, exceto a raiz da árvore, que pode ter o mínimo de um registro.





Página Folha

O termo página folha também é inconsistente, pois é referenciado diferentemente por vários autores. Bayer e McCreight referem-se a estas como as páginas mais distantes da raiz, ou aquelas que contém chaves no nível mais baixo da árvore. Já Knuth define o termo como as páginas que estão abaixo do ultimo nível da árvore, ou seja, páginas que não contém nenhuma chave.



De acordo com a definição de Knuth de ordem e página folha de Bayer e McCreight, uma árvore B de ordem d (número máximo de páginas filhas para uma página pai) deve satisfazer as seguintes propriedades:

-
- Cada página contém no máximo d páginas filhas
- Cada página, exceto a raiz e as folhas, tem pelo menos $\lceil d/2 \rceil$ páginas filhas
- A página raiz tem ao menos duas páginas filhas (ao menos que ela seja uma folha)
- Toda página folha possui a mesma profundidade, na qual é equivalente à altura da árvore
- Uma página não folha com k páginas filha contém $k-1$ chaves
- Uma página folha contém pelo menos $\lceil d/2 \rceil - 1$ chaves e no máximo $d-1$ chaves



Página raiz

A página raiz das árvores B possuem o limite superior de $d-1$ chaves armazenadas, mas não apresentam um número mínimo de chaves, ou seja, elas podem ter um número inferior a $\lceil d/2 \rceil - 1$ de chaves. Na figura acima, essa página é representada pelo nó que possui o registro 7 e 16.

Páginas internas

As páginas internas são as páginas em que não são folhas e nem raiz, estas devem conter o número mínimo $(\lceil d/2 \rceil - 1)$ e máximo $(d-1)$ de chaves.

Páginas folha

Estes são os nós que possuem a mesma restrição de máximo e mínimo de chaves das páginas internas, mas estes não possuem apontadores para páginas filhas. Na figura acima são todos os demais nós exceto a raiz.



Operações Básicas

Altura de uma árvore B

O número de acessos ao disco exigidos para a maioria das operações em uma árvore B é proporcional a altura da árvore B.

Como criar uma árvore vazia

Para construir uma árvore B, primeiro criamos um nó raiz vazio, e depois inserimos novas chaves. Esses procedimentos alocam uma página de disco para ser usada como um novo nó no tempo $O(1)$

As operações básicas sobre uma árvore B são a busca, inserção e remoção de chaves.

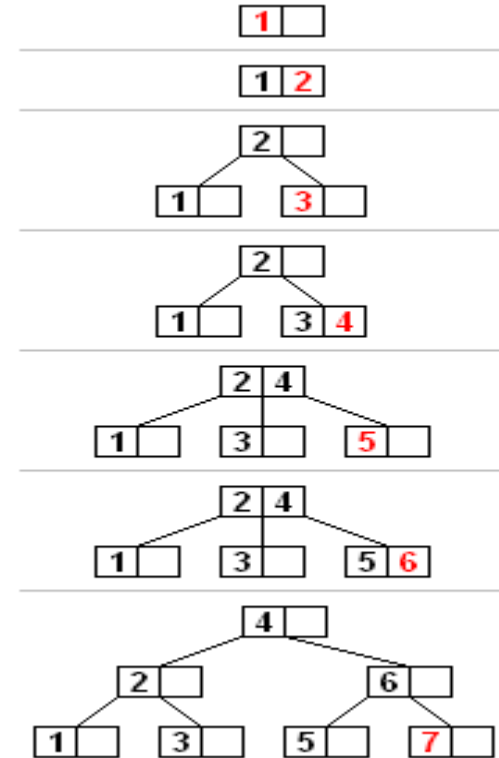


Busca

A busca de uma chave k em uma árvore B é muito parecido com uma busca em árvore binária, exceto pelo fato de que, em vez de tomar uma decisão de ramificação binária ou de “duas vias” em cada nó, tomamos uma decisão de ramificação de várias vias, de acordo com o número de filhos do nó. Em cada nó interno x , tomamos uma decisão de ramificação de $(n[x] + 1)$ vias.

Inserção

A operação de inserção, inicialmente com a árvore vazia, deve garantir que o nó raiz será criado. Criado o nó raiz, a inserção das próximas chaves seguem o mesmo procedimento: busca-se a posição correta da chave em um nó folha e insere a chave garantindo a ordenação destas.





Split

A função do split é dividir o nó em duas partes e "subir" o valor central do nó para um nó acima ou, caso o nó que sofreu o split seja a raiz, criar uma nova raiz com um novo nó. O que ocorre quando é feito um split:

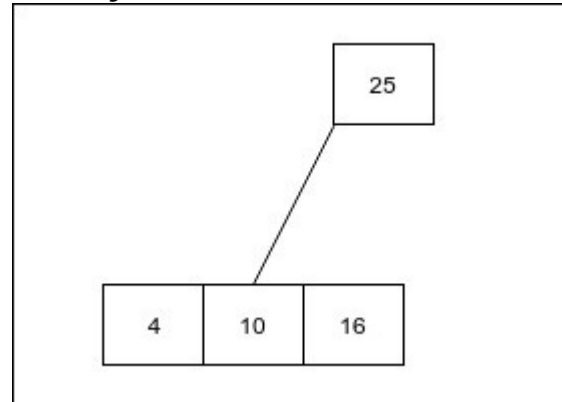
Primeiramente calcula-se qual a mediana dos valores do nó, no caso o valor central do nó. Sendo $\text{tamanho} = \text{quantidade de elementos no nó}$, $\text{mediana} = \text{tamanho}/2$ e usamos a mediana para acessar o elemento que se encontra no centro do nó, no caso $\text{valor_central} = \text{valores}[\text{mediana}]$;

É testado se o nó que sofreu split tem pai, caso não, cria-se um novo nó apenas com o valor valor_central e o define como a nova raiz. São criados mais dois nós, cada um irá conter os valores do nó que estavam antes da mediana e depois da mediana. Um nó terá os valores menores que o valor_central e ficará na primeira posição dos filhos da nova raiz, e o outro nó terá os valores maiores que o valor_central e ficará na segunda posição dos filhos da nova raiz;

Caso o nó tenha pai, adicionamos o valor_central ao nó pai.



Caso o nó pai já esteja cheio, este também vai sofrer split após a inserção do valor nele. E da mesma forma que criamos dois nós para o caso do nó não ter pai, criaremos dois nós que conterão os valores menores e maiores que o `valor_central`. O nó com os menores valores ficará posicionado como filho do lado esquerdo do `valor_central` e o nó com os maiores valores ficará posicionado como filho do lado direito do `valor_central`. Por exemplo: Caso o `valor_central` seja inserido na posição 0 do array de valores do nó pai, o nó filho com os menores valores ficará na posição 0 do array de filhos, e o nó com os maiores valores ficará na posição 1 do array de filhos.

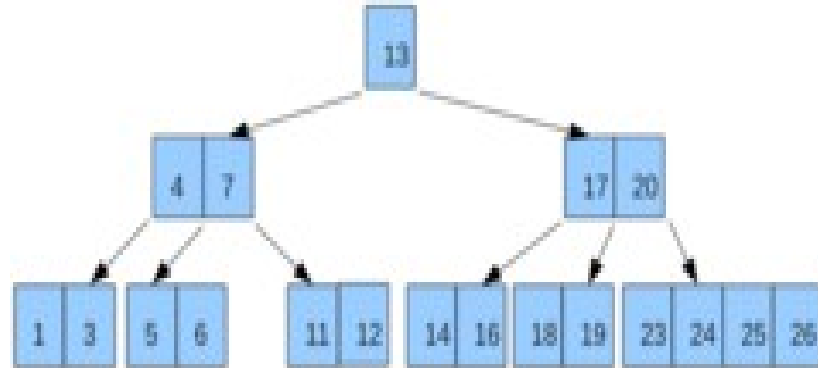
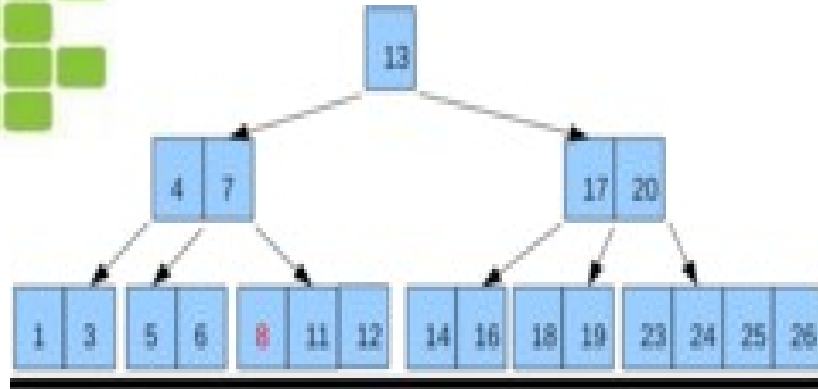




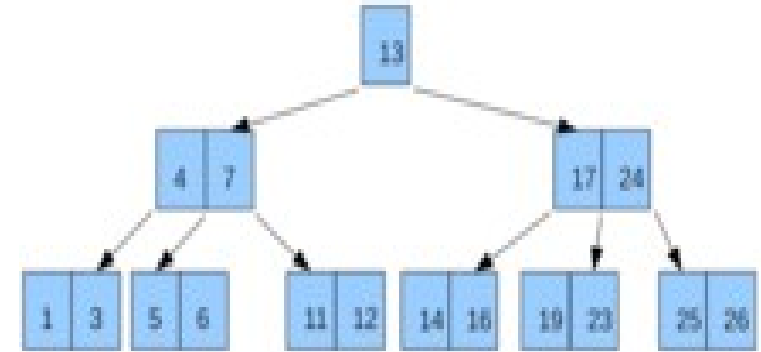
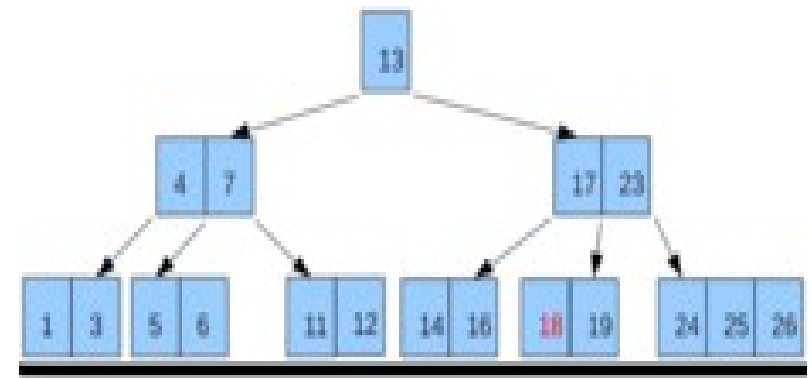
Remoção

A remoção é análoga a inserção, sendo que o algoritmo de remoção de uma árvore B deve garantir que as propriedades da árvore sejam mantidas, pois uma chave pode ser eliminada de qualquer página e não apenas de páginas folha. A remoção de um nó interno exige que os filhos do nó sejam reorganizados. Como na inserção, devemos nos resguardar contra a possibilidade da eliminação produzir uma árvore cuja estrutura viole as propriedades de árvores B. Lembrando que devemos assegurar que um nó não ficará pequeno demais durante a eliminação (a não ser pelo fato da raiz pode ter essa pequena quantidade de filhos).

O método para remover a chave k da subárvore com raiz em x deve estar estruturado para garantir que quando ele for chamado recursivamente em um nó x , o número de chaves em x seja pelo menos o grau mínimo t . Essa condição exige uma chave além do mínimo exigido pelas condições normais da árvore B, de forma que, quando necessário, uma chave seja movida para dentro do nó filho.



Neste caso a remoção da chave 8 não causa o underflow na página folha em que ela está, portanto ela é simplesmente apagada e as outras chaves são reorganizadas mantendo sua ordenação.



é apresentado a técnica de redistribuição de chaves. Na remoção da chave 18, a página que contém essa chave possui uma página irmã à direita com um número superior ao mínimo de chaves (página com chaves 24, 25 e 26) e, portanto, estas podem ser redistribuídas entre elas de maneira que no final nenhuma delas tenha um número inferior ao mínimo permitido.



Código/Conclusão

Acesso ao código: <https://github.com/jppreti/ED220221>, na branch Mateus_Joicy_LG





Referência Bibliográfica

ÁRVORE B. Wikipédia, a enciclopédia livre. Disponível em:

https://pt.wikipedia.org/wiki/%C3%81rvore_B#:~:text=A%20%C3%A1rvore%20B%20%C3%A9%20uma,dados%20relativamente%20grandes%2C%20como%20discos

.Acesso em: 24 maio 2022.





INSTITUTO FEDERAL
MATO GROSSO
Campus Cuiabá

Obrigada!

Joicy Kelly¹ e Mateus Goulart²

Email: lara.j@estudante.ifmt.edu.br¹
mateus.goulart@estudante.ifmt.edu.br²