Árvores B e B+

João Paulo Dias de Almeida
jp.dias.almeida@gmail.com

Universidade Federal de Sergipe

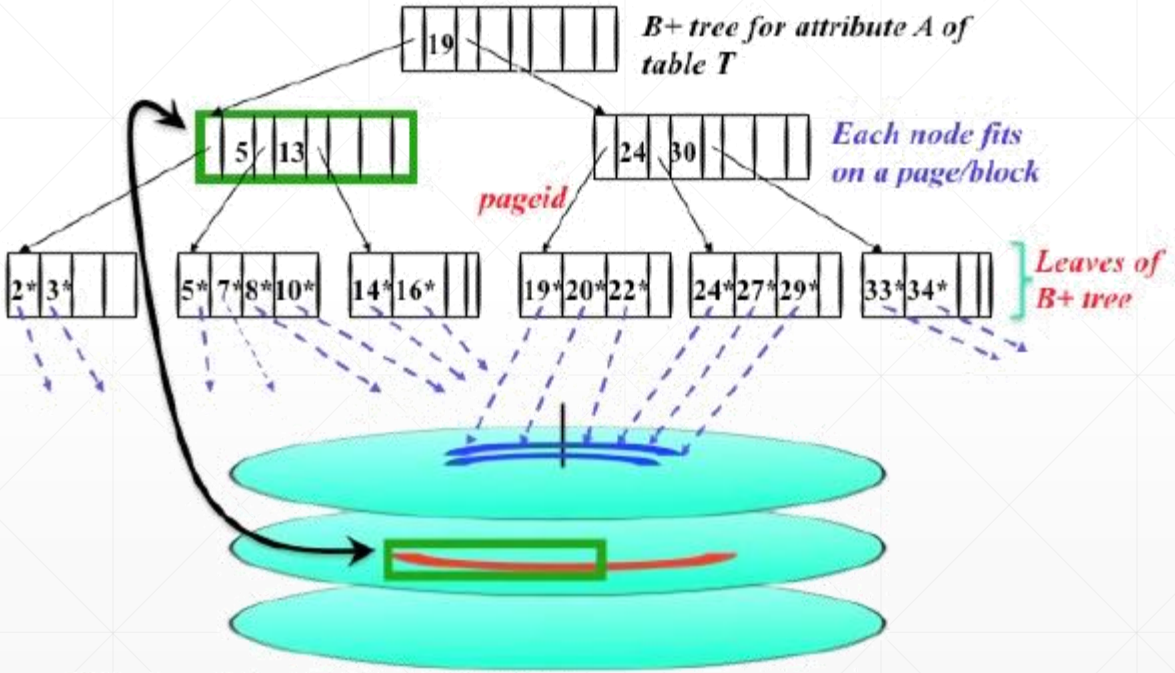
O que vamos aprender hoje?

- Entender como funciona uma árvore B e B+
- Identificar o contexto onde a árvore B é eficiente
- Descrever o custo da busca em uma árvore B

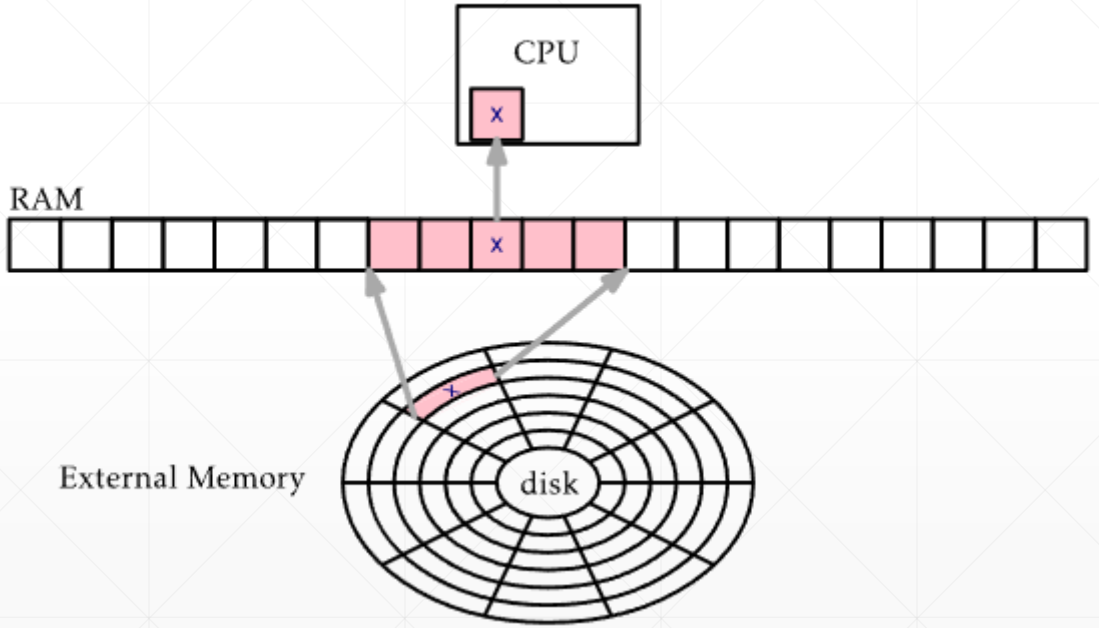
Introdução

- As árvores B são árvores de busca auto balanceáveis
  - Assim como AVL e rubro-negra
- Projetada especificamente para armazenamento de dados em discos magnéticos
  - A quantidade de acessos a disco é determinante para entender o custo desta estrutura
    - DISCO << RAM
  - Neste caso a complexidade é definida pela soma do custo da leitura no disco + o custo de acesso dos dados na árvore
    - A quantidade de acessos aumenta proporcionalmente à altura da árvore

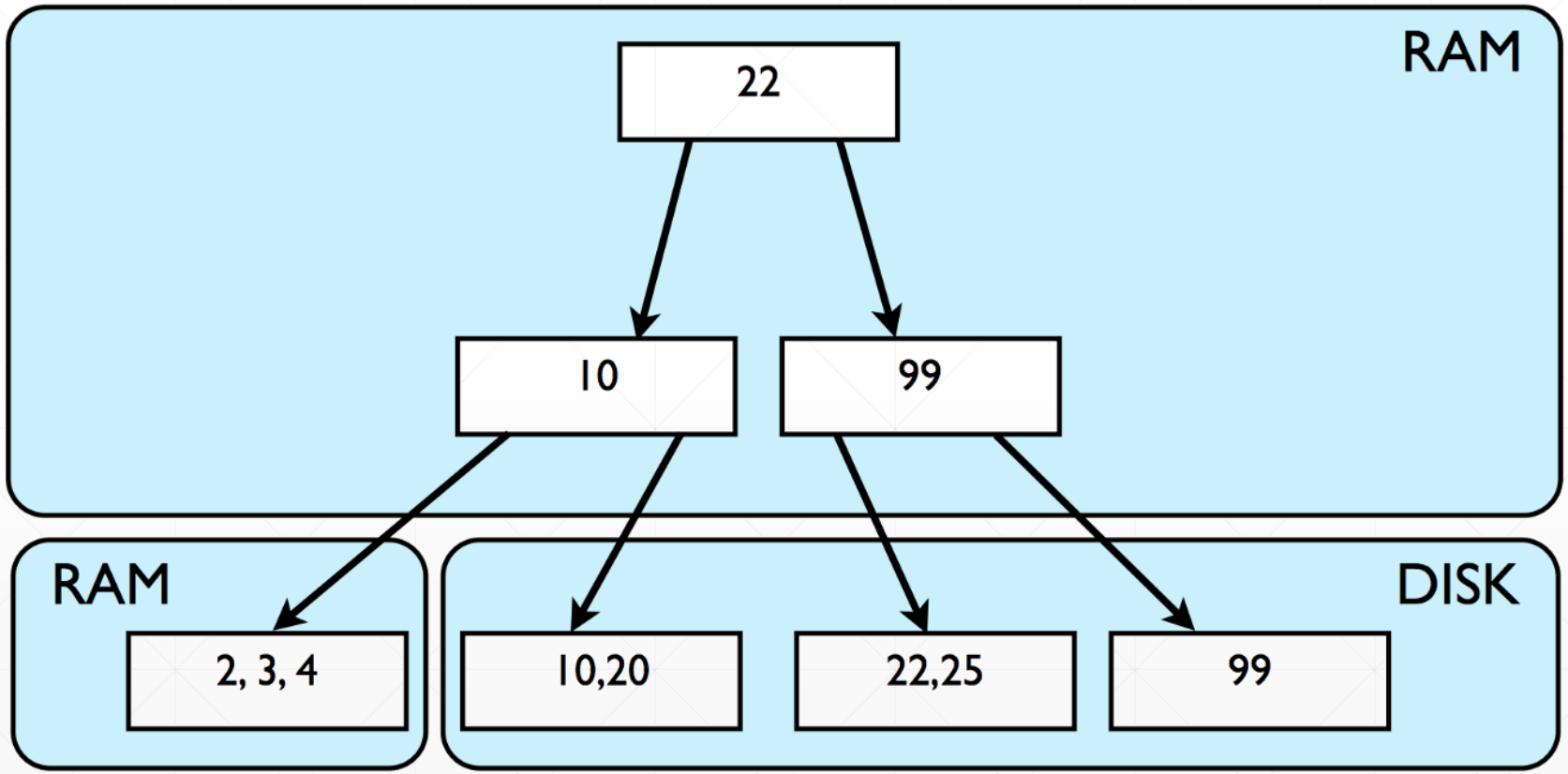
Armazenamento externo



Armazenamento externo



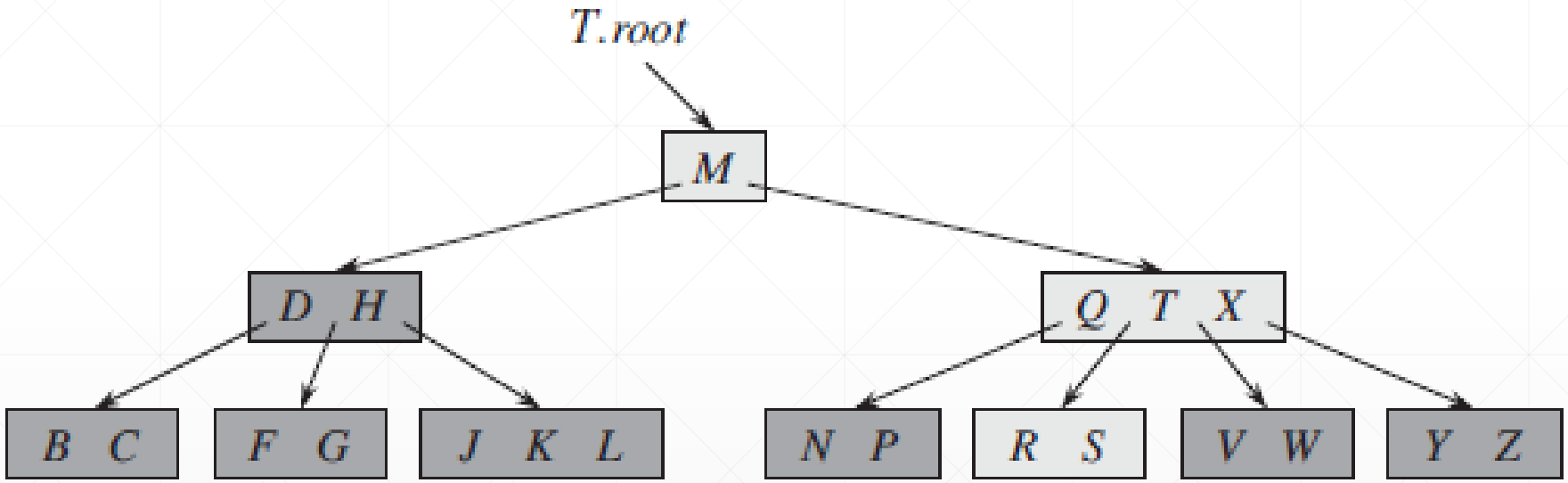
Armazenamento externo



Árvore-B

- As folhas possuem a mesma profundidade
- A árvore-B possui um grau mínimo t (t >= 2)
  - t costuma depender do tamanho do bloco de disco e da entrada
- Todo nó (exceto raiz) deve possuir no mínimo t-1 chaves e no máximo 2t – 1 chaves

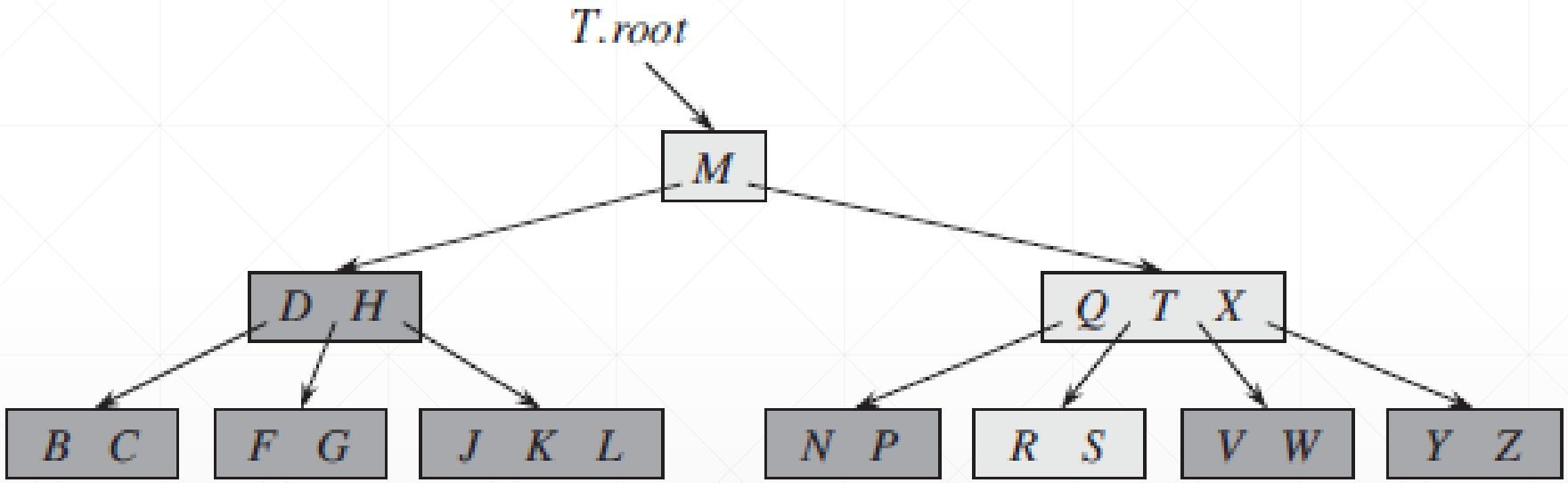
Exemplo de árvore B



Árvore-B

- As folhas possuem a mesma profundidade
- A árvore-B possui um grau mínimo t (t >= 2)
- Todo nó (exceto raiz) deve possuir no mínimo t-1 chaves e no máximo 2t – 1 chaves
- Se um nó interno x possui n chaves então x deve possuir n+1 filhos
  - As chaves nos nós intermediários servem como pontos de divisão que separam a faixa de chaves manipuladas por x em n+1 subfaixas
    - Cada subfaixa corresponde a um filho de x

Exemplo de árvore B



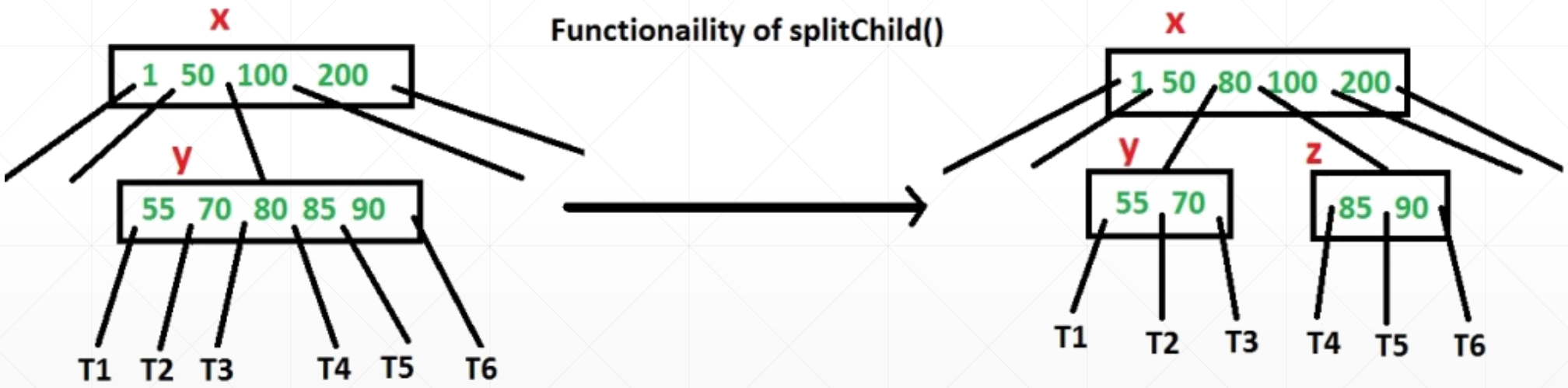
Busca: custo

- O algoritmo de busca na árvore-B é O(h)
  - h = altura da árvore
- A altura de toda árvore-B é O(lg n)
  - Portanto o custo da busca é O(t lg n) acessos ao disco

Inserção

- A inserção de valores sempre é feita nos nós folha
  - É importante verificar se o nó é capaz de armazenar o valor a ser inserido
    - Limite de 2t-1 chaves
  - Se o nó estiver cheio, é necessário dividi-lo

Divisão do nó folha



Inserção: algoritmo

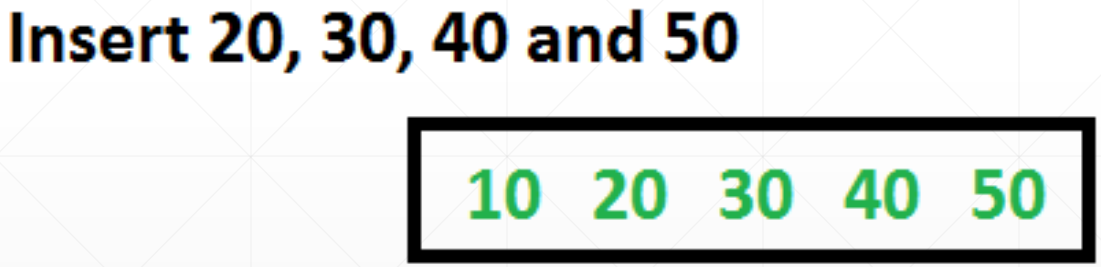
1. Inicialize um ponteiro x na raiz
2. Enquanto x não aponta para uma folha, faça:
    1. Encontre o filho de x que vai ser visitado na sequencia. Vamos chama-lo de y
    2. Se y não estiver cheio, faça x apontar para y
    3. Se y estiver cheio, divida-o e faça x apontar para uma das partes de y.
        1. Se o valor a ser inserido for menor do que a chave do meio de y, x aponta para primeira parte, caso contrário aponta para a segunda. Geralmente a chave do meio de y sobe para x.

Inserção: exemplo

- Vamos inserir as chaves 10, 20, 30, 40, 50, 60, 70, 80 e 90 em uma árvore-B de grau 3
  - 2*t – 1 = 5

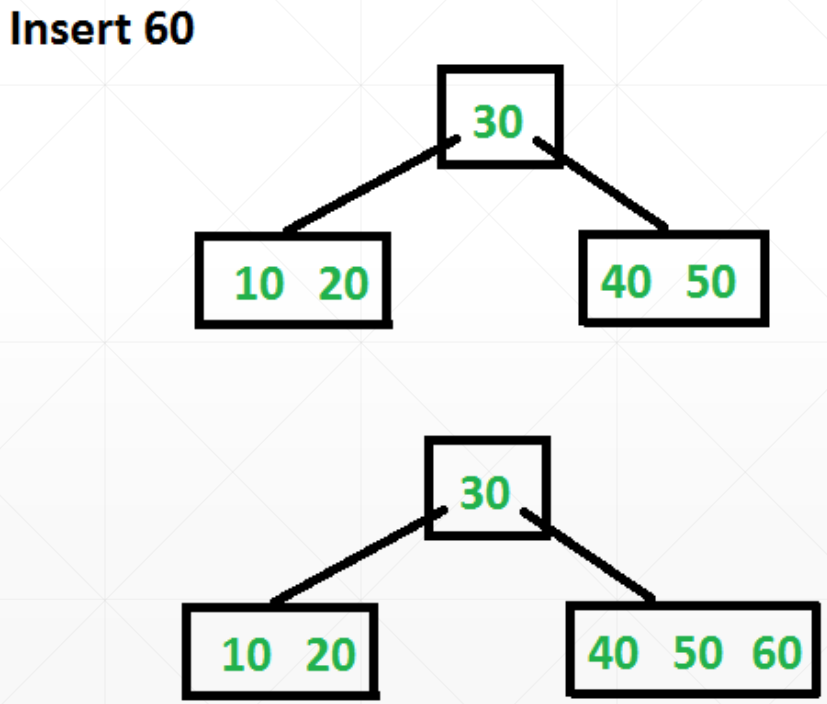
Inserção: exemplo

- 20, 30, 40 e 50 serão inseridos na raiz

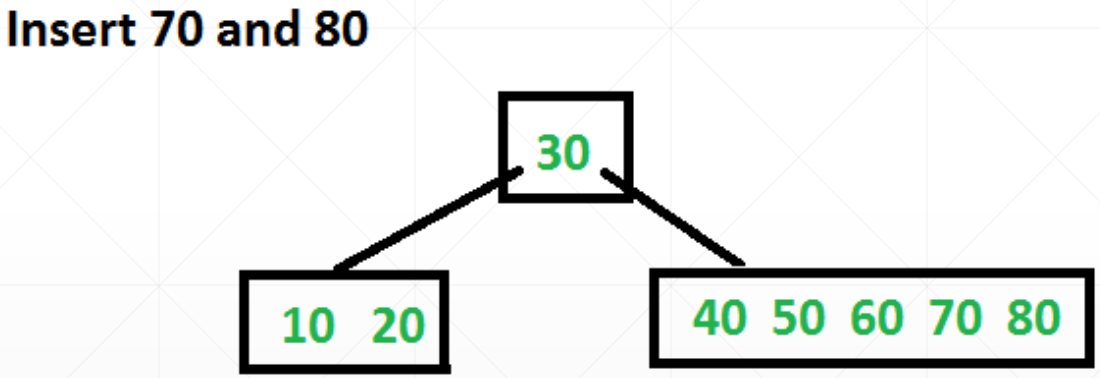


Inserção: exemplo

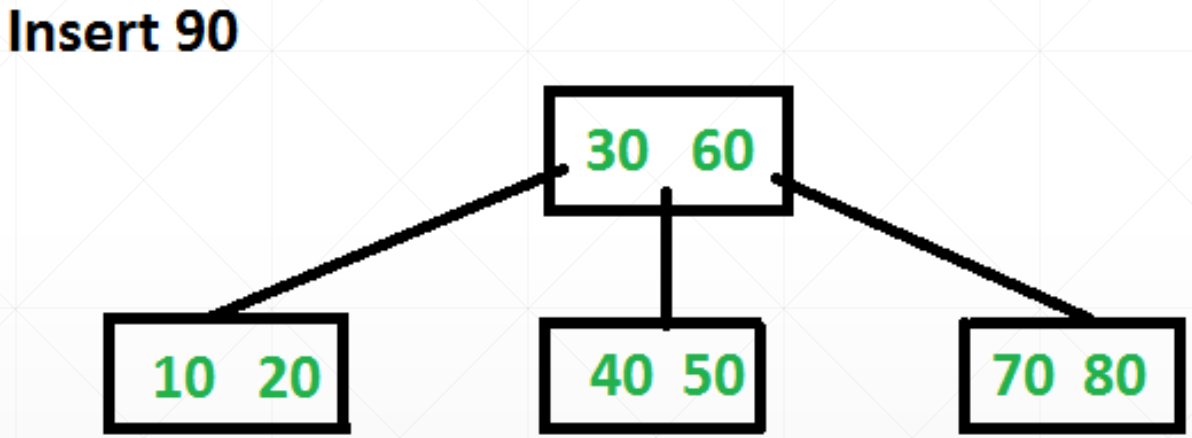
- Inserção do 60:



Inserção: exemplo



Inserção: exemplo

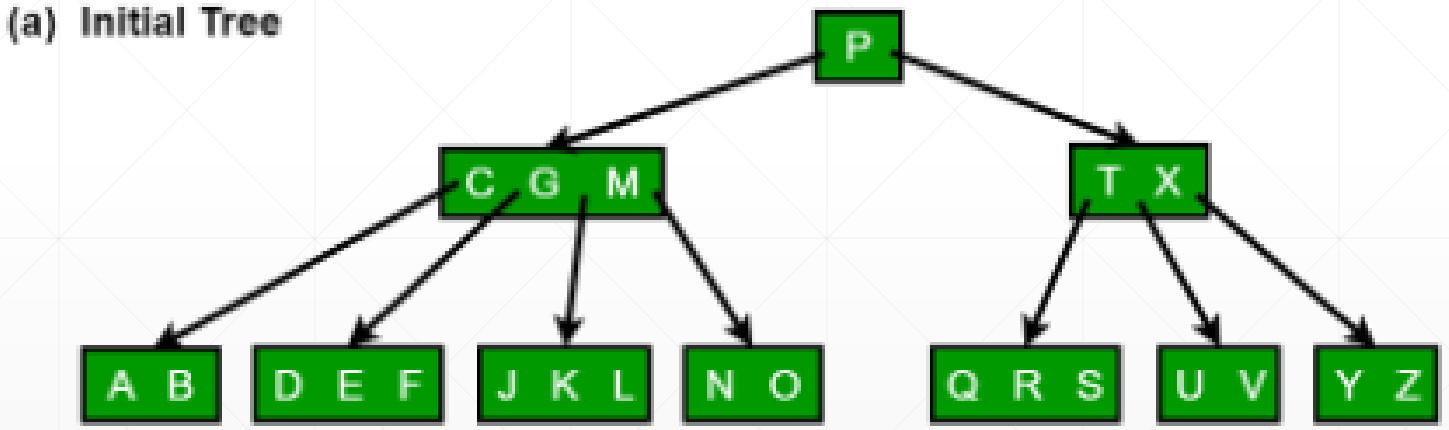


Remoção

- A remoção pode ocorrer em qualquer nó da árvore
  - Caso ocorra em um nó intermediário é necessário fazer o rearranjo de todos os nós inferiores
- Assim como nas outras árvores, é necessário garantir que as propriedades da árvore-B não sejam violadas
  - O nó possui tamanho mínimo e máximo
- Existem diversos casos de remoção, todos direcionados à manter as propriedades da árvore

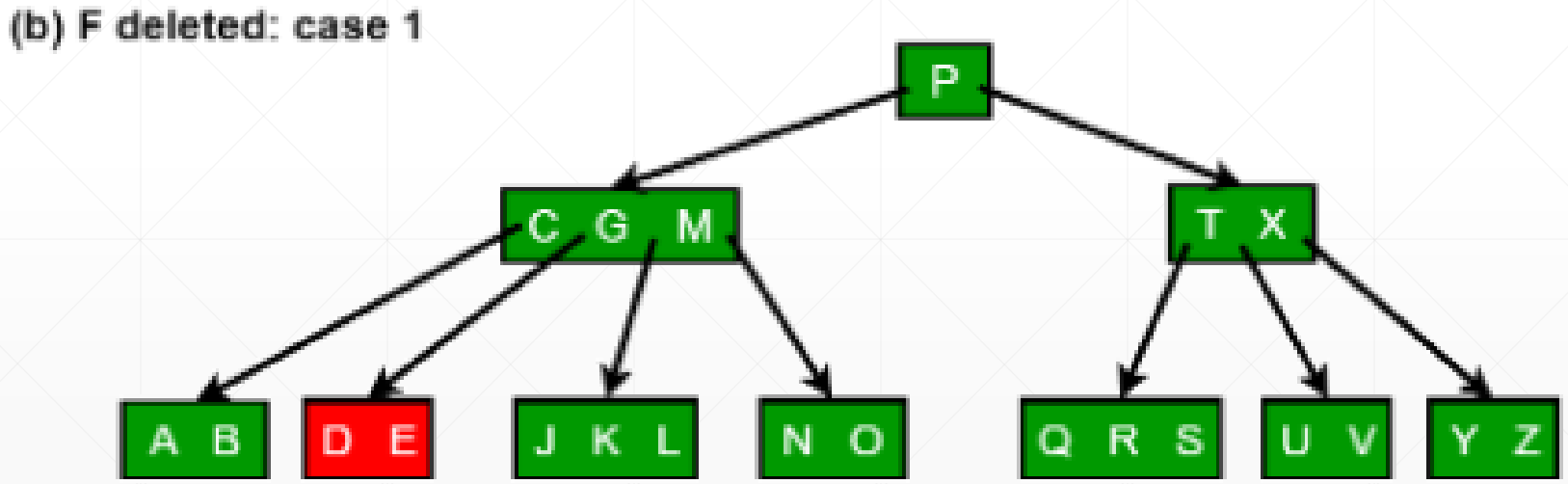
Remoção

- Considere a árvore abaixo de grau 3



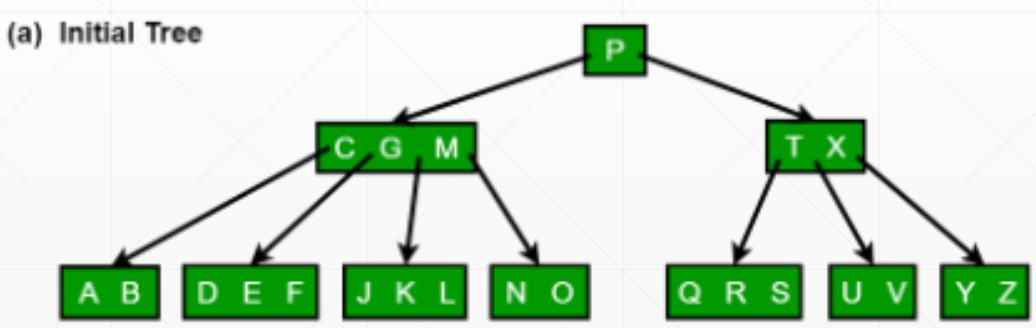
Caso 1: Remova F

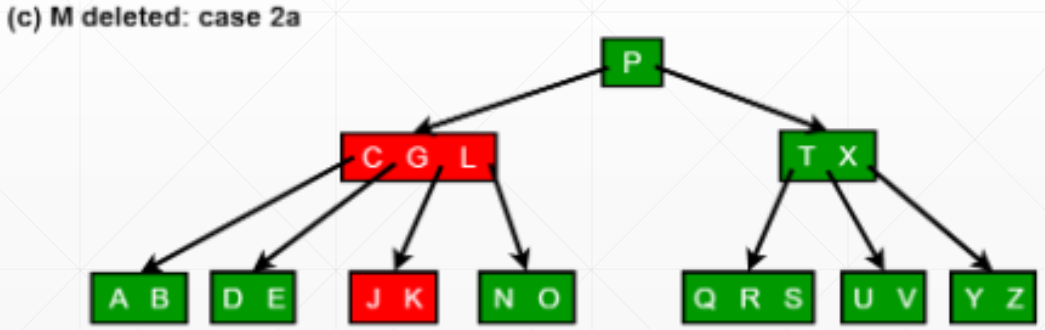
- **A chave a ser removida esta na folha**



Caso 2a: Remova M

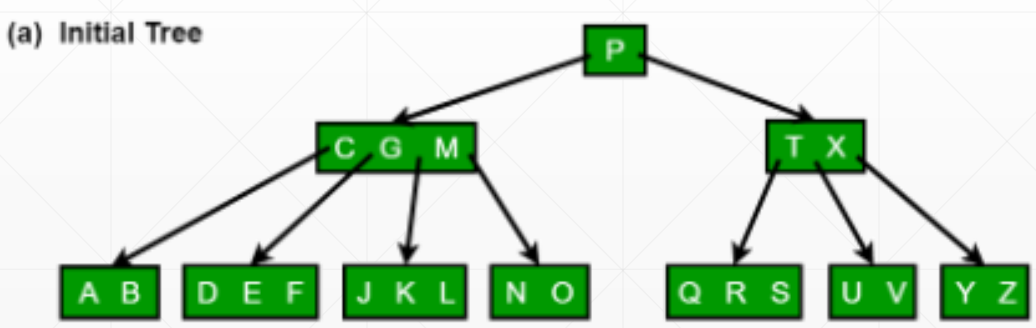
- **A chave a ser removida está em um nó intermediário**
  - Se o filho y que precede a chave tem pelo menos t chaves, encontre o predecessor da chave, remova-o e substitua-o no nó intermediário

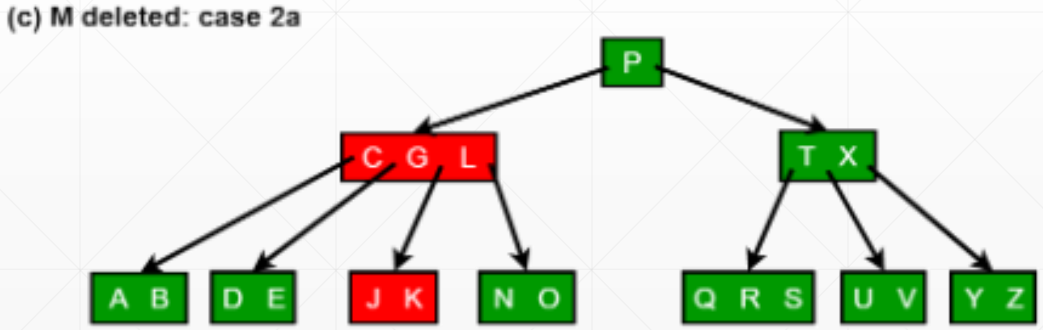




Caso 2a: Remova M

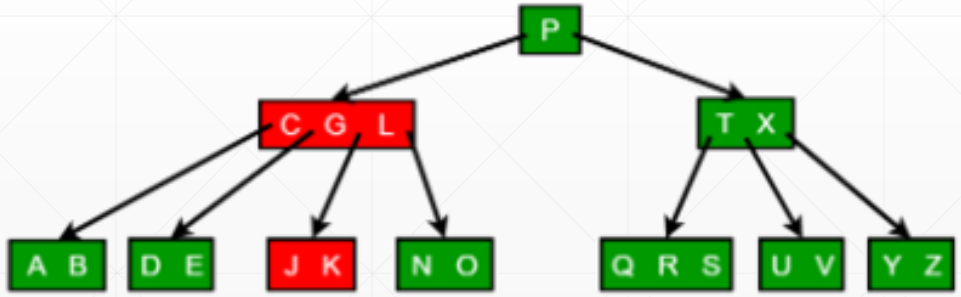
- **A chave a ser removida está em um nó intermediário**
  - Caso não funcione com o antecessor, tente com o sucessor

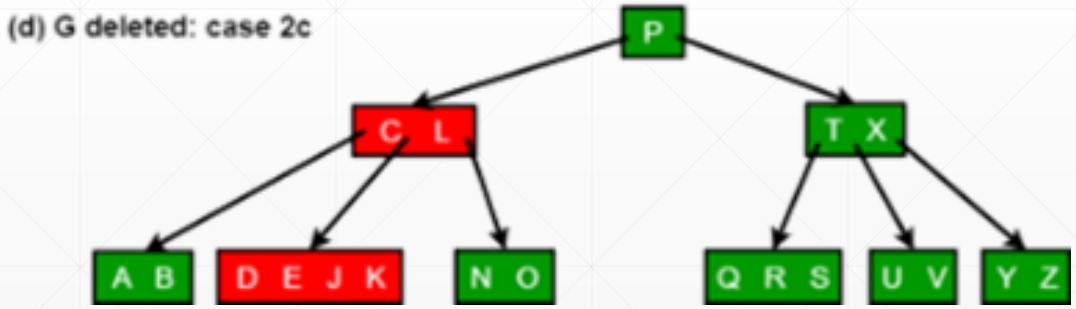




Caso 2c: Remova G

- **A chave a ser removida está em um nó intermediário**
  - Caso antecessor e sucessor possuam t-1 chaves: junte antecessor e sucessor. Nó intermediário perde a chave e a referência para o sucessor da chave.

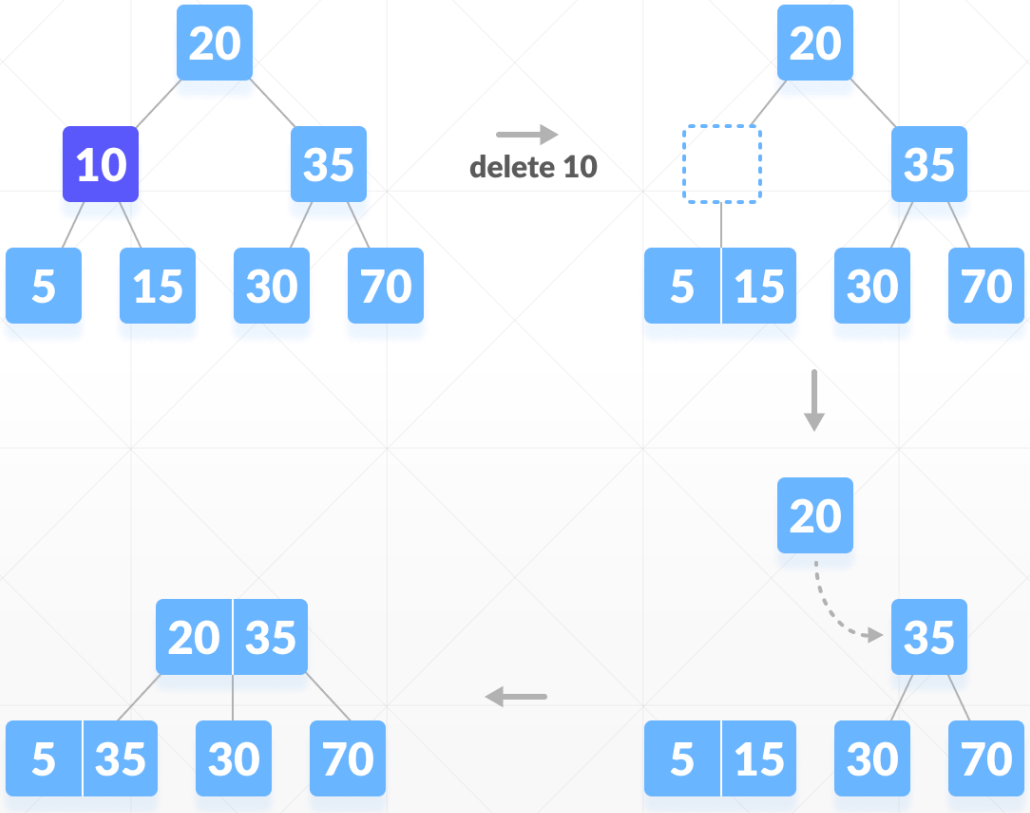




Caso 3:

- **A altura da árvore precisa diminuir**
  - A chave a ser removida está em um nó intermediário e a remoção deixa o nó com menos chave do que o permitido e não tem como pegar emprestado do filho (2a)
    - Filhos já possuem menor quantidade possível
  - Junte o filho com o irmão e atualize o pai da chave removida.

Caso 3:


20
10
35
5
15
30
70
delete 10
20
35
5
15
30
70
20
35
5
15
30
70
20
35
5
35
30
70

Árvore B+

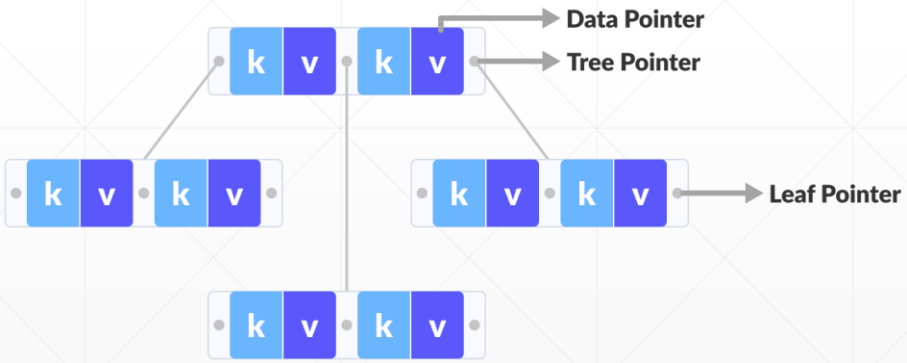
- A árvore-B+ não possui ponteiros para dados nos nós intermediários
  - Isso permite que os nós intermediários possuam mais ponteiros para nós da árvore
  - A altura fica menor do que uma árvore-B

Árvore-B+: propriedades

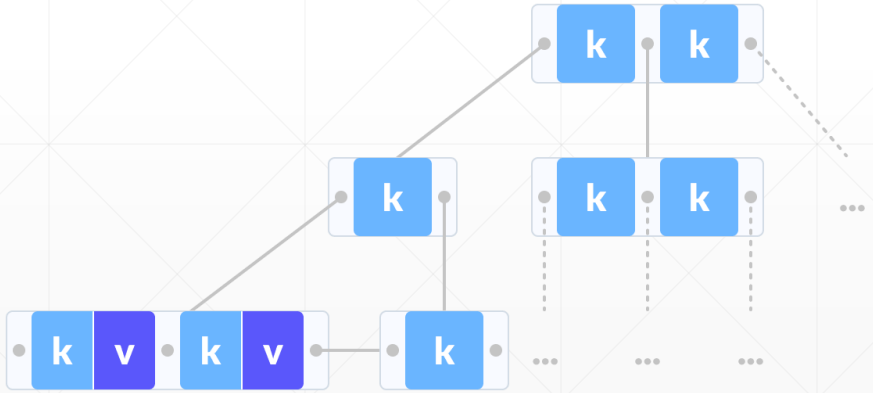
- Todas as folhas devem estar no mesmo nível
- A raiz deve possuir pelo menos dois filhos
- Cada nó (exceto a raiz) deve possuir um máximo de m filhos e mínimo de m/2 filhos
  - m → número máximo de ponteiros em um nó
- Cada nó pode possuir um máximo de m-1 chaves e um mínimo de (m/2)-1 chaves

Comparação

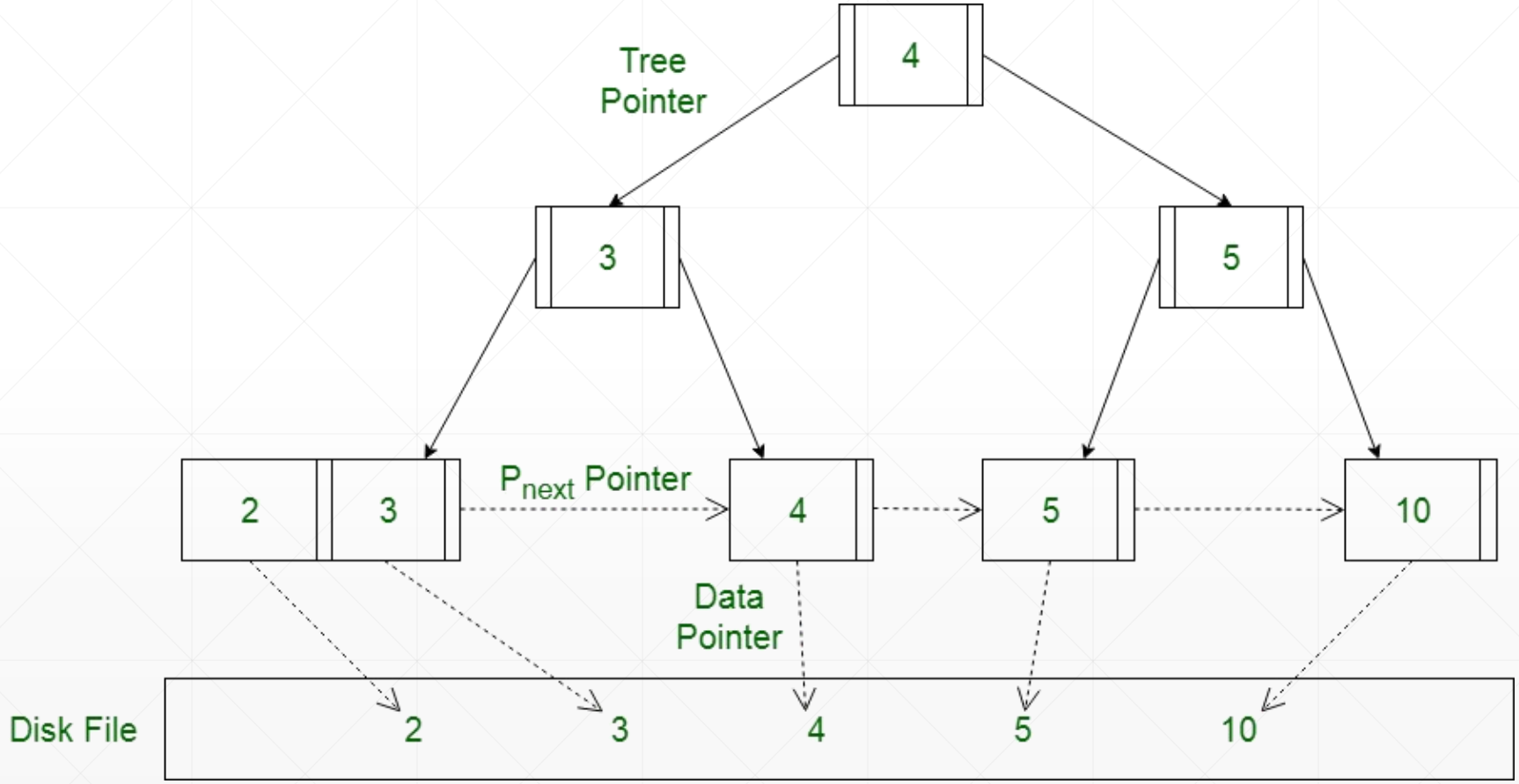
Árvore-B



Árvore-B+



Árvore-B+


Tree
Pointer
4
3
5
2
3
P_{next} Pointer
4
5
10
Data
Pointer
Disk File
2
3
4
5
10

Árvore-B+: vantagens

- Armazena mais entradas nos nós intermediários na mesma altura h de uma árvore-b
- Os dados podem ser acessados diretamente e sequencialmente
- A quantidade de acessos ao disco para acessar qualquer dado é igual

Aplicações

- São utilizadas em grandes bases de dados para acessar dados armazenados em discos
  - Os nós costumam ter milhares de filhos
- Busca por dados no disco de forma eficiente
- Servidores costumam utilizar árvores-B

Atividade

- Implementar a operação de inserção na árvore-B descrita no Classroom

Referências

- CORMEN, T. H. et al. **Algoritmos: teoria e prática**. Elsevier, 2012.
- ZIVIANI, N. **Projeto De Algoritmos com Implementações em Pascal e C**. Cengage Learning, 3ª Ed, 2010.
- SKIENA, Steven S. **The Algorithm Design Manual**. Springer, 2020.
- GeeksforGeeks. Insert Operation in B-tree. Disponível em: https://www.geeksforgeeks.org/insert-operation-in-b-tree/

Referências

- GeeksforGeeks. **Introduction of B+-tree**. Disponível em: https://www.geeksforgeeks.org/introduction-of-b-tree/
- Programiz. **B+-tree**. Disponível em: https://www.programiz.com/dsa/b-plus-tree#:~:text=Properties%20of%20a%20B%2B%20Tree&text=Each%20node%20can%20contain%20a,%2F2%E2%8C%89%20%2D%201%20keys

Dúvidas?