UFRRJ
UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

Estrutura de Dados II

Aula 1 - Árvores Balanceadas - AVL e B

Raul Sena Ferreira

Introdução

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

- Árvores balanceadas são extremamente versáteis
- Estão presentes em várias linguagens de programação
  - C++: std::map / std::set
  - Java: TreeMap / TreeSet
  - def Tree(): return collections.defaultdict(Tree)
- São utilizadas em vários problemas reais como, acesso à disco, grafos dinâmicos e etc
- Conseguem suprir várias deficiências das árvores binárias de busca (Binary Search Trees, ou, simplesmente **BST**)

Árvores AVL

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Criadores soviéticos: Georgy **A**delson-**V**elsky and Evgenii **L**andis

Para cada Nó da Árvore, as alturas de suas Subárvores diferem de, no máximo, 1

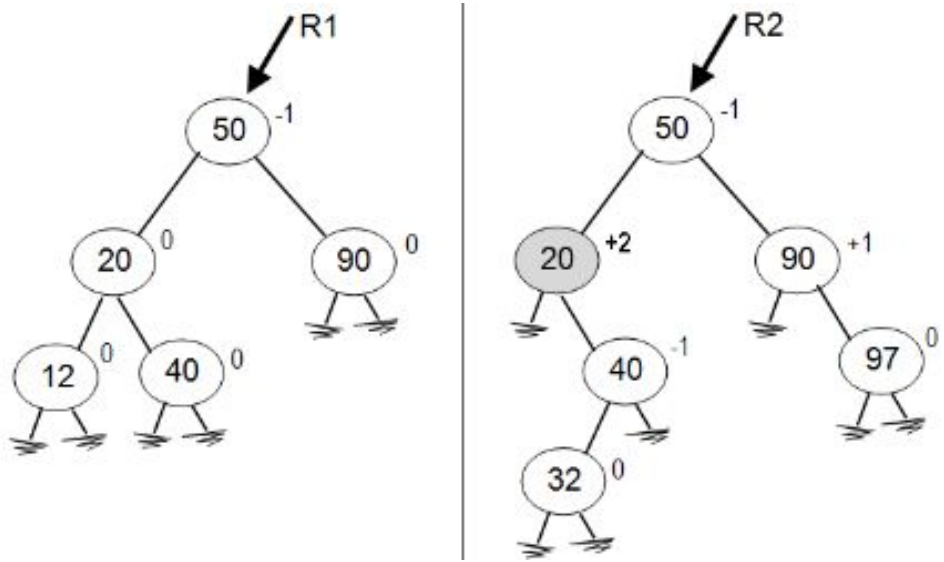
- Fator de Balanceamento (**FB**) = altura da subárvore direita - altura da subárvore esquerda

Estrutura de dados muito eficiente para busca, inserção e remoção em BST

- Busca em uma árvore desbalanceada de 10.000 nós pode resultar em 5.000 comparações. Em uma árvore AVL, a média baixa para 14 comparações

Árvores AVL

Como saber quando está balanceada / desbalanceada?



2017.1

Árvores AVL

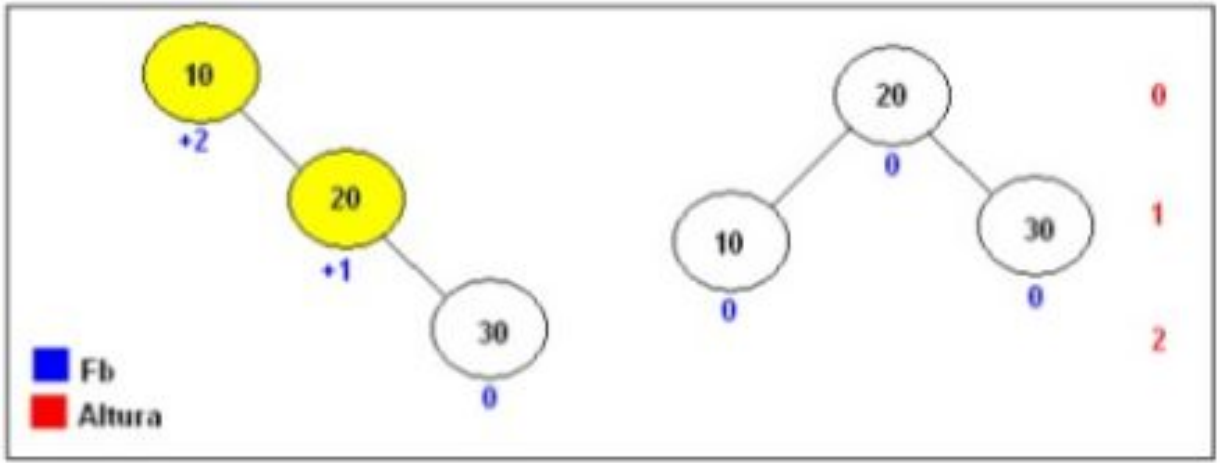
Como balancear?

- Rotação simples à esquerda
- Rotação simples à direita
- Rotação dupla à esquerda
- Rotação dupla à direita

UFRRJ
UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

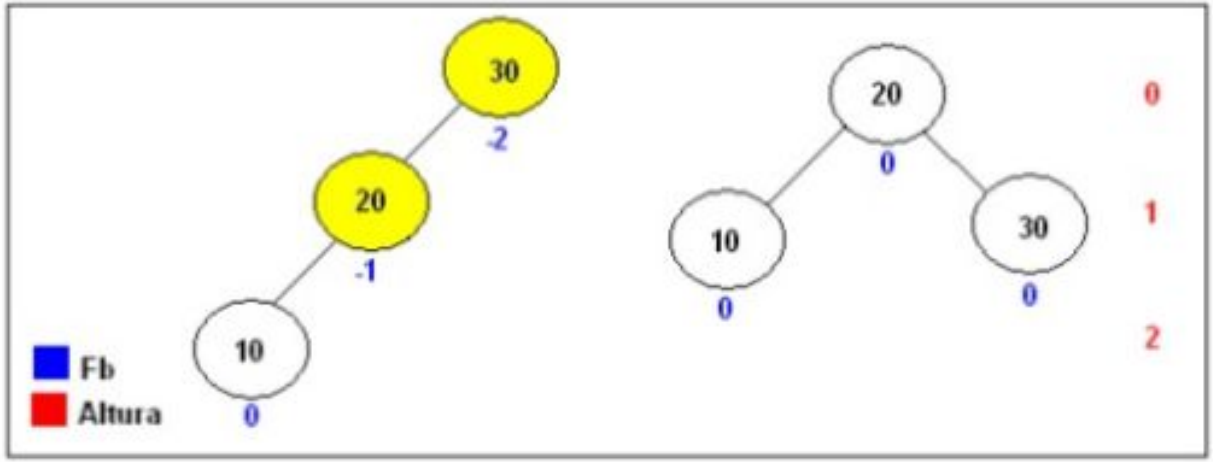
2017.1

Árvores AVL - Rotação à esquerda



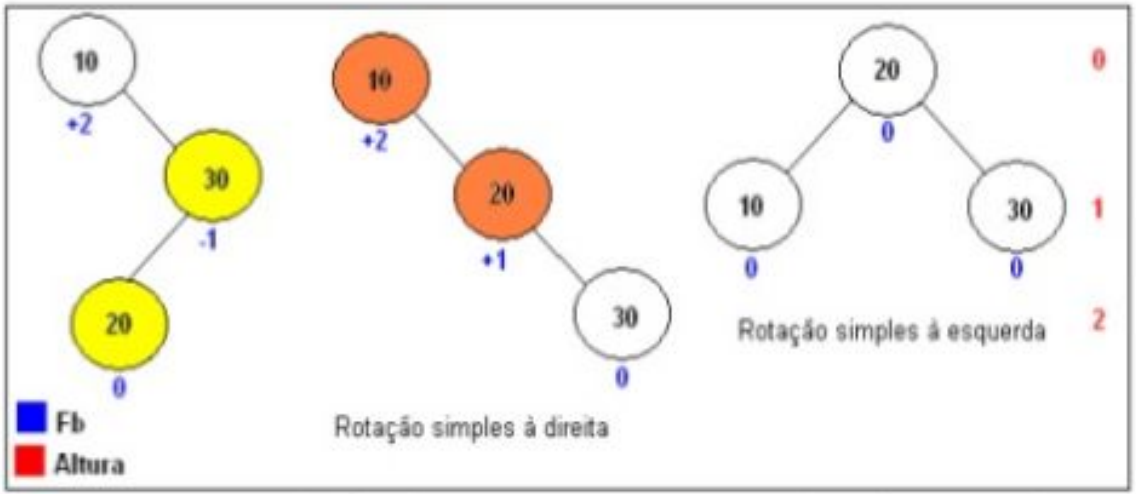
2017.1

Árvores AVL - Rotação à direita



2017.1

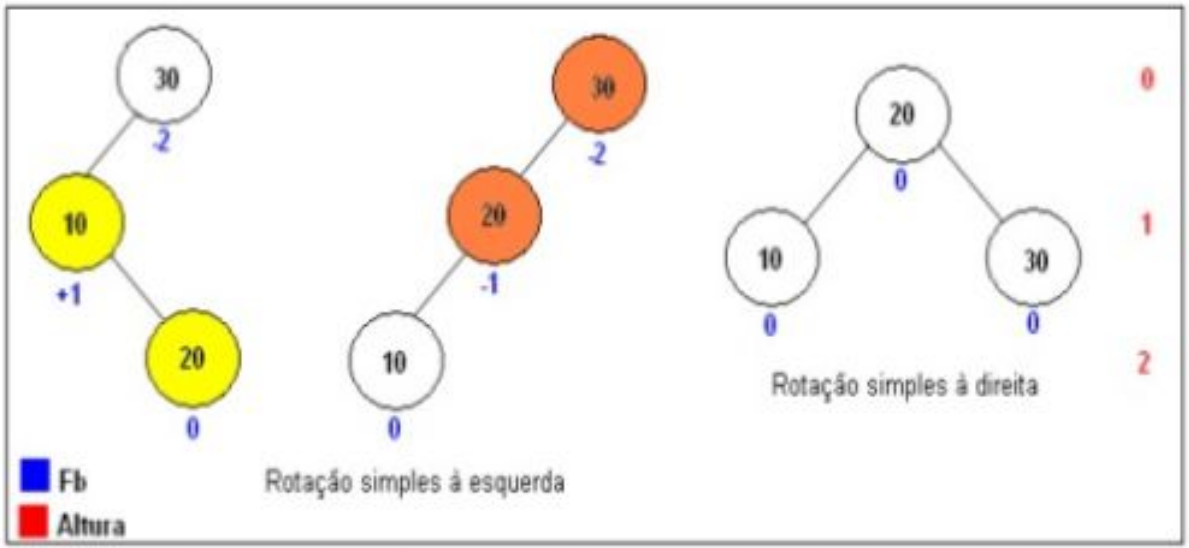
Árvores AVL - Rotação dupla à esquerda



(rotação simples à direita + rotação simples à esquerda)

2017.1

Árvores AVL - Rotação dupla à direita



(rotação simples à esquerda + rotação simples à direita)

2017.1

Árvores AVL

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Como saber quando fazer a rotação é simples ou dupla?

- Se o sinal do **FB** for **igual**, então é **rotação simples**
- Se o sinal do **FB** for **diferente**, então é **rotação dupla**

Como saber se a rotação deve ser à esquerda ou à direita?

- Se o **FB** for **positivo**, então a rotação é para a **esquerda**
- Se o **FB** for **negativo**, então a rotação é para a **direita**

Árvores B

Vantagens:

- Desenvolvido para guardar e acessar, de forma otimizada, blocos de informação
- É uma generalização da BST

Desvantagens:

- Nós podem ter múltiplas chaves. Mais tempo gasto dentro de cada nó
- Inserção e remoção podem ser custosos

Usaremos o termo página para falar dos nós de uma árvore B

Árvores B

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

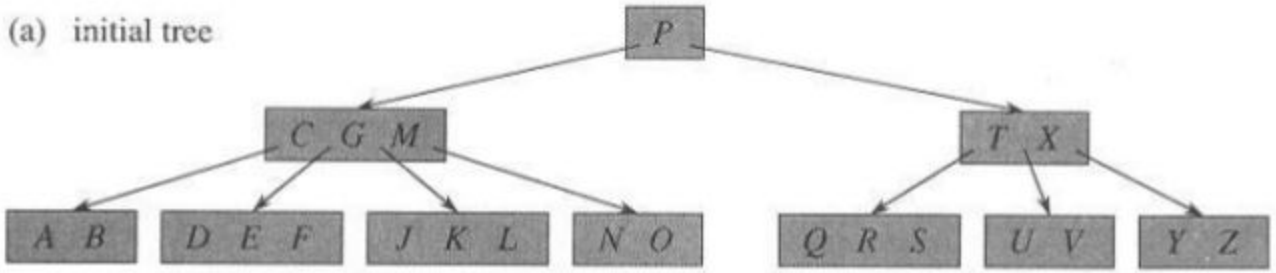
As folhas da árvore são páginas externas e os demais nós são páginas internas.

Uma árvore B de ordem m possui as seguintes propriedades:

- Todos as folhas possuem a mesma profundidade
- Todos os nós, com exceção da raiz, têm entre m-1 e (m/2)-1 chaves
- A raiz tem no máximo m-1 chaves

Árvores B

- Ordem m = 3
- Máximo de registros por página = m-1 = 4
- Mínimo de registros por página = (m-1)/2 = 1
- Altura máxima da árvore contendo n páginas = $\log_m ((n+1)/2)$ ou $O(\log_m n)$



2017.1

Árvores B - Busca

Busque na raiz. Se nao encontrar, busque nas páginas filhas

Complexidade da busca:

- $O(\log m * \log_m n) = O(\log m * (\log n / \log m)) = O(\log n)$

Árvores B - Inserção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Inserção em uma página da árvore B de ordem m:

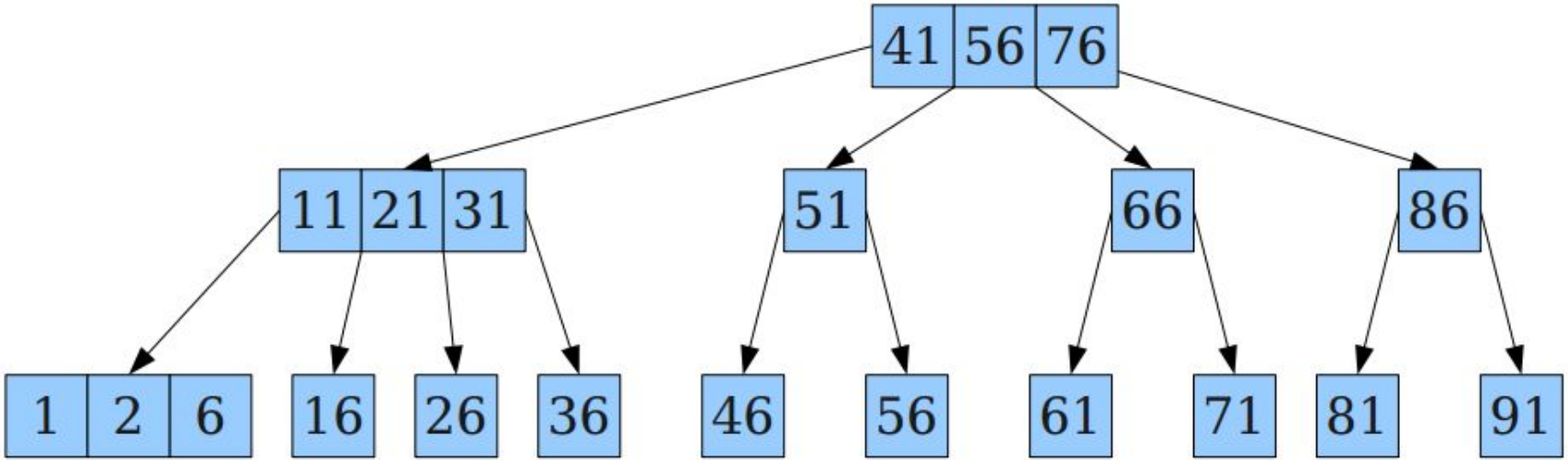
- Procure a chave e a insira no último nó (página) visitado
- Se a página estiver cheia
  - Divida (split) a página em duas páginas de tamanho m
  - Remova a maior chave do primeiro bloco e faça dessa chave o pai dos dois blocos
  - Recursivamente adicione a página ao pai, fazendo o mesmo processo de “split” pra cima
- Complexidade total
  - $O(m \log_m n)$

2017.1

Árvores B - Inserção

Inserção em uma página cheia:

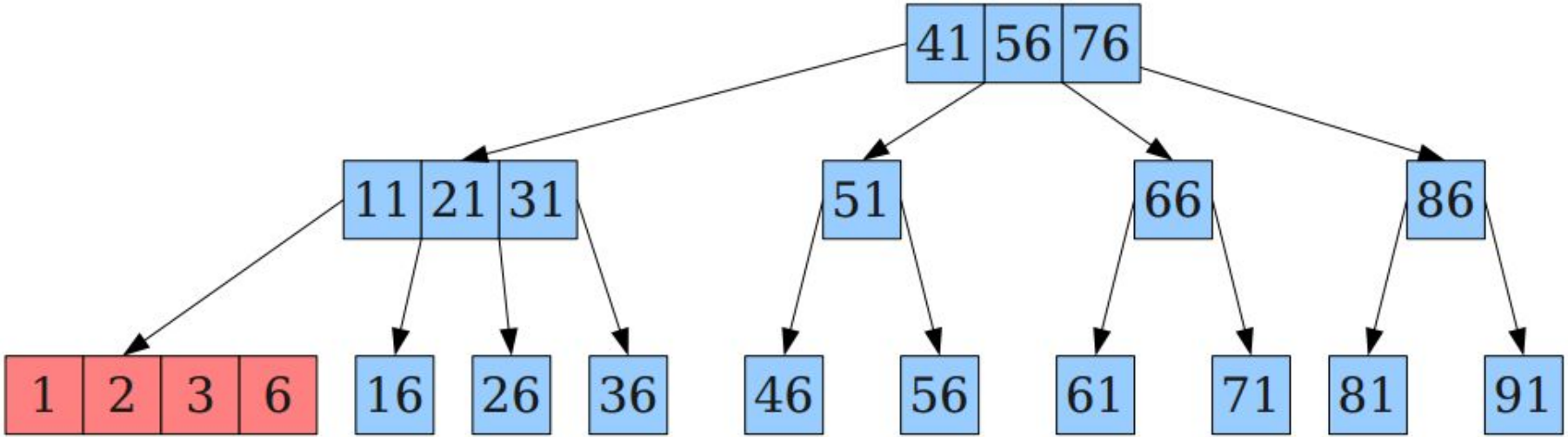
- Dividir a página em dois e propagar pra cima



Árvores B - Inserção

Inserção em uma página cheia:

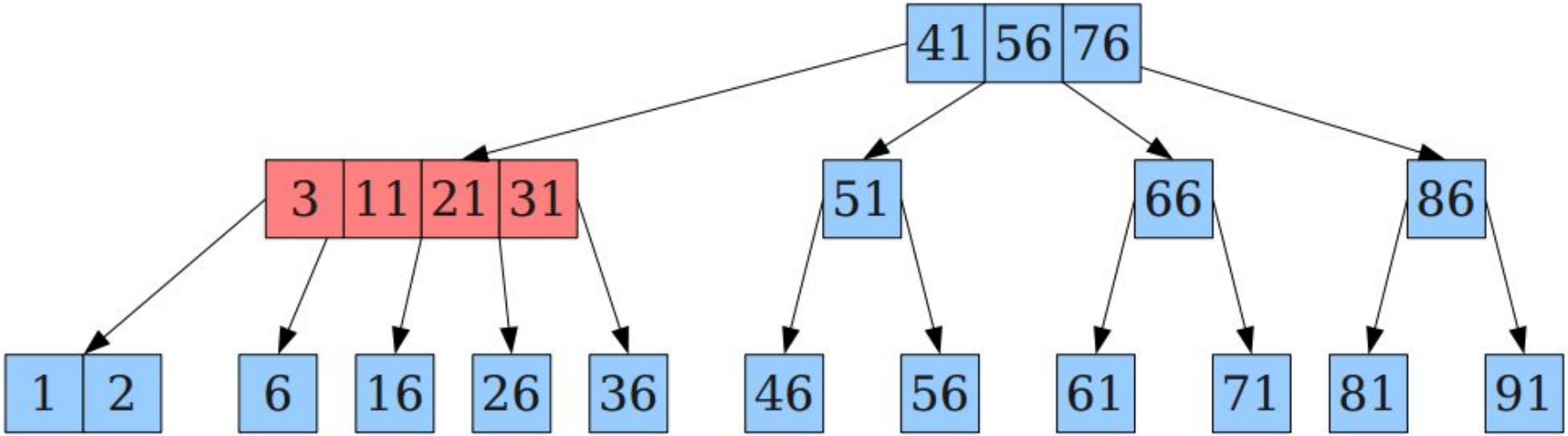
- Dividir a página em dois e propagar pra cima



Árvores B - Inserção

Inserção em uma página cheia:

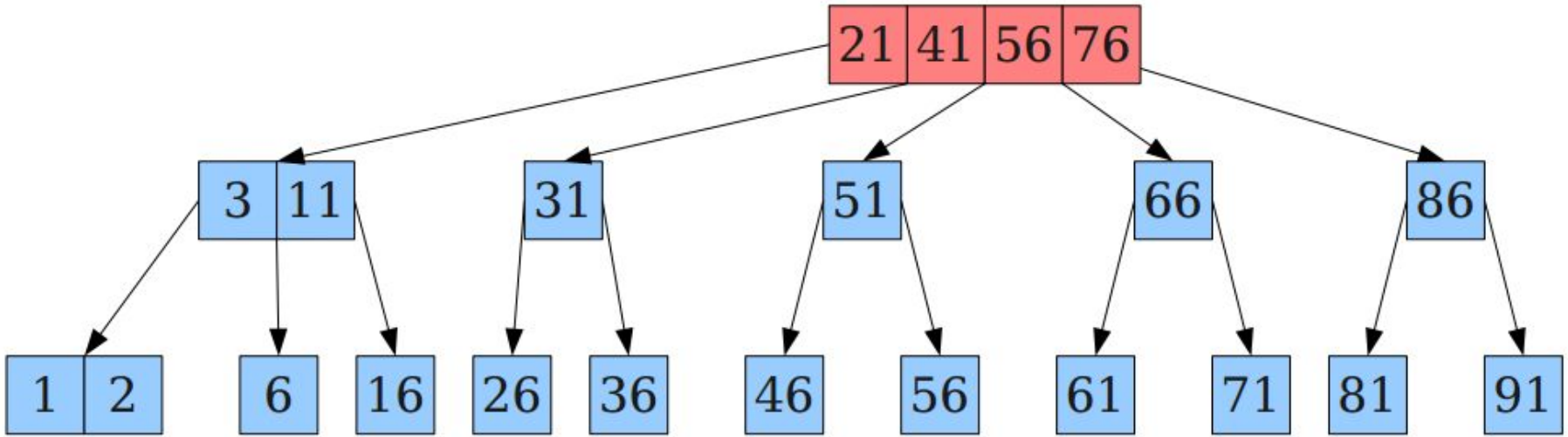
- Dividir a página em dois e propagar pra cima



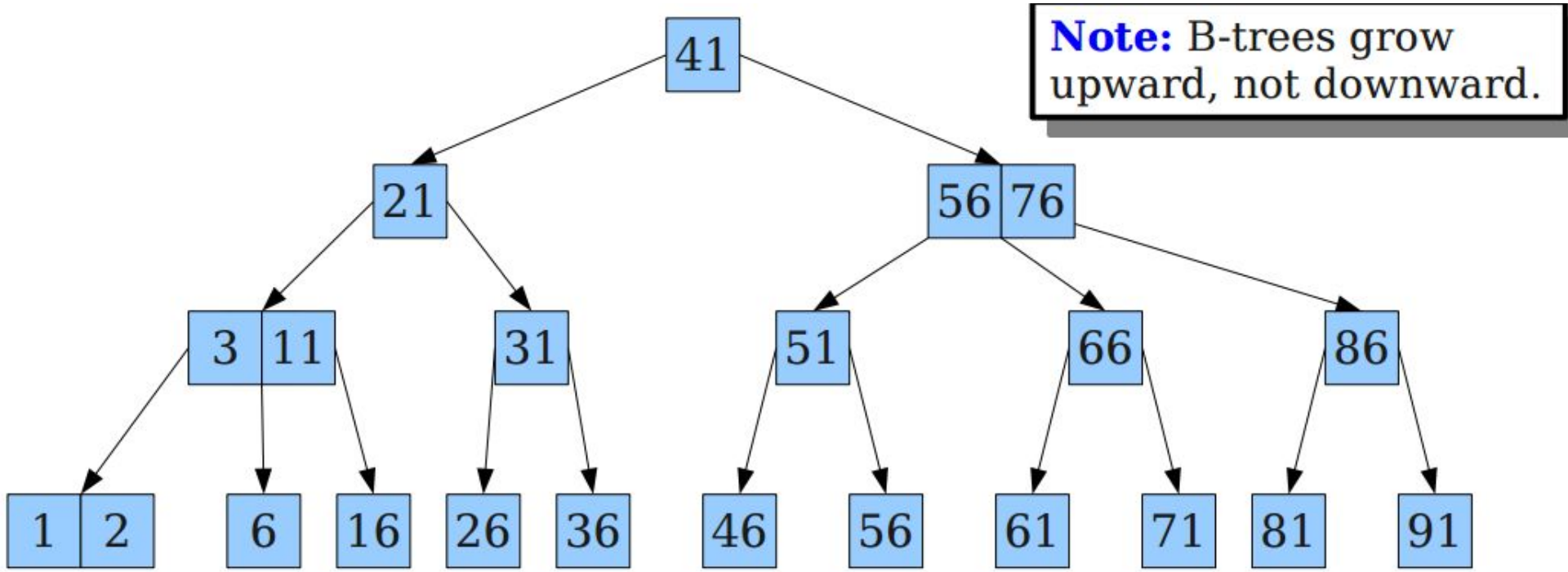
Árvores B - Inserção

Inserção em uma página cheia:

- Dividir a página em dois e propagar pra cima



Árvores B - Inserção



2017.1

Árvores B - Remoção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

A remoção de um elemento de uma árvore B pode ser dividida em dois casos:

- O elemento que será removido está em uma página externa (folha)
- O elemento que será removido está em uma página interna

Duas operações podem acontecer:

- Redistribuição
- Concatenação

2017.1

Árvores B - Remoção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

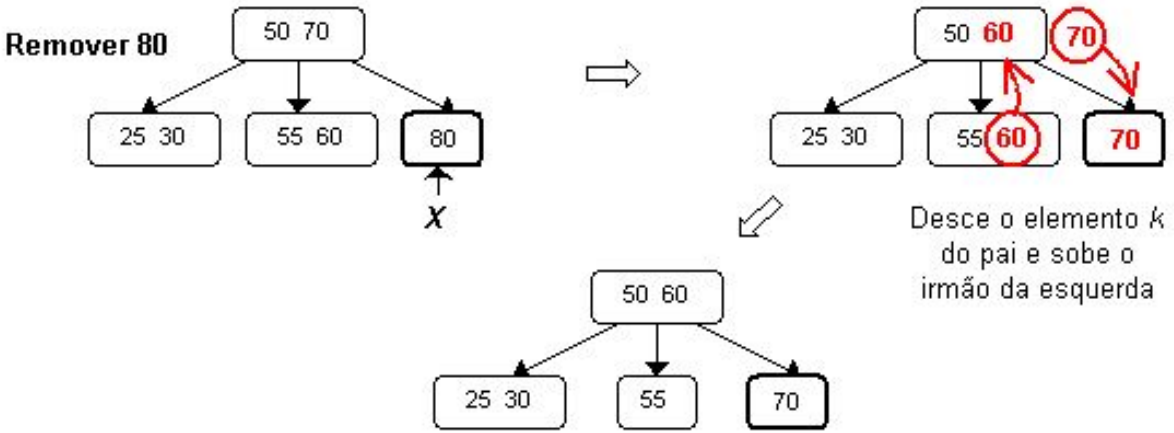
Se a entrada a ser removida não está em uma folha, troque-a com seu sucessor (ou predecessor) na ordem natural das chaves: Remova a entrada da folha

Se a folha contém mais que o número mínimo de entradas então não há mais nenhuma ação.

Caso contrário, se o nó contém o número mínimo de entradas, considere os dois irmãos imediatos do nó:

- Se um dos irmãos têm mais que o número mínimo de entradas, então redistribua uma entrada deste irmão para o nó pai e uma entrada do nó pai para o filho “deficiente”

Árvores B - Remoção



2017.1

Árvores B - Remoção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Exemplode redistribuição:

Seja uma árvore B de ordem 5 onde queremos remover o elemento F

EH

ABCD FG IJ

2017.1

Árvores B - Remoção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Exemplode redistribuição:

Seja uma árvore B de ordem 5 onde queremos remover o elemento F

EH

ABCD G IJ

Quantidade de chaves menor do que o número mínimo de elementos (5/2)-1=2: Redistribuição é necessária!

Árvores B - Remoção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Exemplode redistribuição:

Seja uma árvore B de ordem 5 onde queremos remover o elemento F

H

ABCDEG IJ

Irmão à direita tem o mínimo mas irmão à esquerda tem mais do que o mínimo.

Redistribuição: Junte o nó pai aos dois nós irmãos separados.

Árvores B - Remoção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Exemplode redistribuição:

Seja uma árvore B de ordem 5 onde queremos remover o elemento F

CH

AB DEG IJ

Overflow! Uma divisão deve ser feita para completar a redistribuição. O elemento do meio sobe e o nó então se transforma em dois

Árvores B - Remoção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Exemplo de redistribuição:

Seja uma árvore B de ordem 5 onde queremos remover o elemento F

CH

AB DEG IJ

Overflow! Uma divisão deve ser feita para completar a redistribuição. O elemento do meio sobe e o nó então se transforma em dois. Todas as propriedades satisfeitas. Redistribuição concluída!

2017.1

Árvores B - Remoção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Exemplo de concatenação:

Seja uma árvore B de ordem 5 onde queremos remover o elemento B

CF

AB DE GHIJ

2017.1

Árvores B - Remoção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Exemplo de concatenação:

Seja uma árvore B de ordem 5 onde queremos remover o elemento B

CF

A DE GHIJ

Número mínimo de chaves não respeitado. Olha-se para a página irmã. Ela possui o mínimo de chaves e não possui máximo de chaves. Então aplica-se concatenação!

Árvores B - Remoção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Exemplo de concatenação:

Seja uma árvore B de ordem 5 onde queremos remover o elemento B

F

ACDE GHIJ

Elemento pai e seu irmão se junta ao nó

Percebe-se que em uma concatenação, o elemento pai sempre diminui de tamanho, ao contrário da redistribuição, onde o elemento pai continua com a mesma cardinalidade

Árvores B - Remoção

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Exemplo de concatenação:

Seja uma árvore B de ordem 5 onde queremos remover o elemento B

FG

ACDE HIJ

O processo é recursivo. Nesse caso o pai ficou com sua cardinalidade menor que o mínimo.

Suba então o filho mais à direita. Número mínimo da folha está sendo satisfeita? Caso sim, acaba o processo. Se não, aplica-se os métodos anteriores

Segundo trabalho!

UFRRJ
UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

Implementar uma árvore AVL em python

Deve permitir inserção, busca e remoção, dado uma chave

Árvore deve sempre se manter balanceada

Entrega até às 23:59 do dia 01/05 através do moodle do curso!

Fazer o mais completo possível ainda que o trabalho incompleto valha nota

Os trabalhos são em dupla e... Trabalhos copiados levarão 0 !

2017.1

Terceiro trabalho!

UFRRJ
UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

Implementar uma árvore B em python

Deve permitir inserção, busca e remoção, dado uma chave

Entrega até às 23:59 do dia 08/05 através do moodle do curso!

Fazer o mais completo possível ainda que o trabalho incompleto valha nota

Os trabalhos são em dupla e... Trabalhos copiados levarão 0 !

2017.1

"...if you aren't, at any given time, scandalized by code you wrote five or even three years ago, you're not learning anywhere near enough."

Nick Black