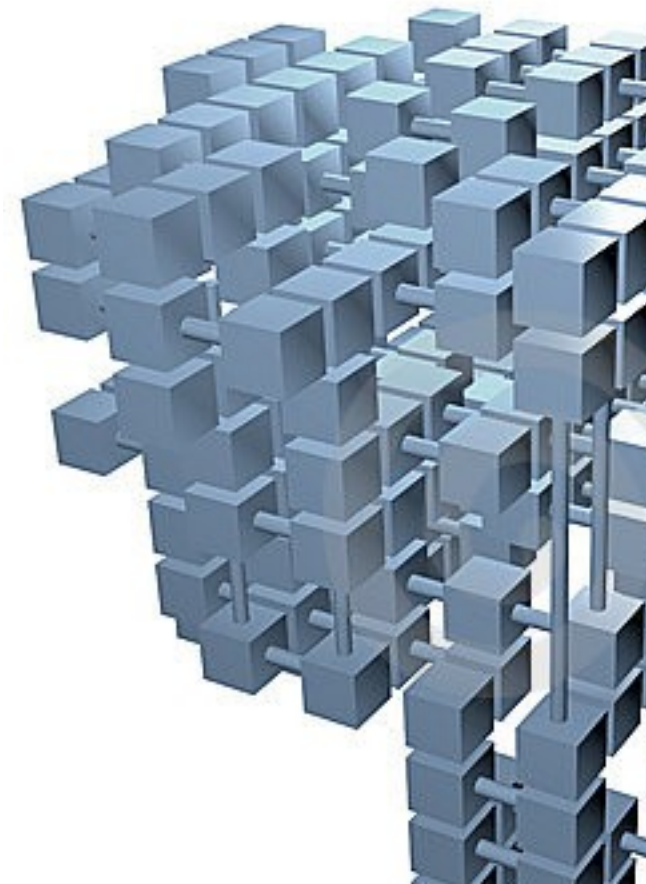


Sistemas de Informação

Estrutura de Dados II

Árvores multi-way

Prof. Ivan José dos Reis Filho
ivanfilhoreis@gmail.com

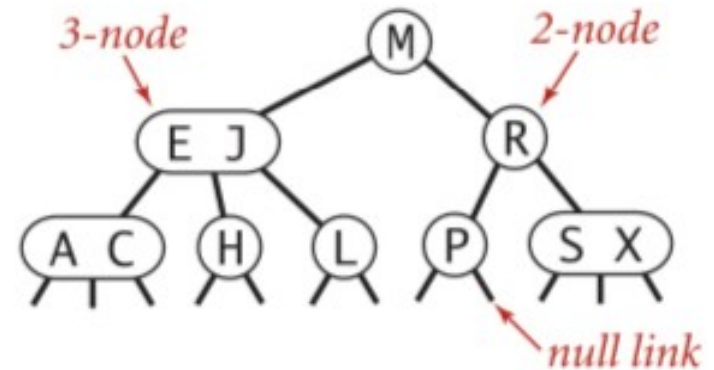


Aula Anterior

- Processamento de texto
- Trie Padrão
- Código de Huffman

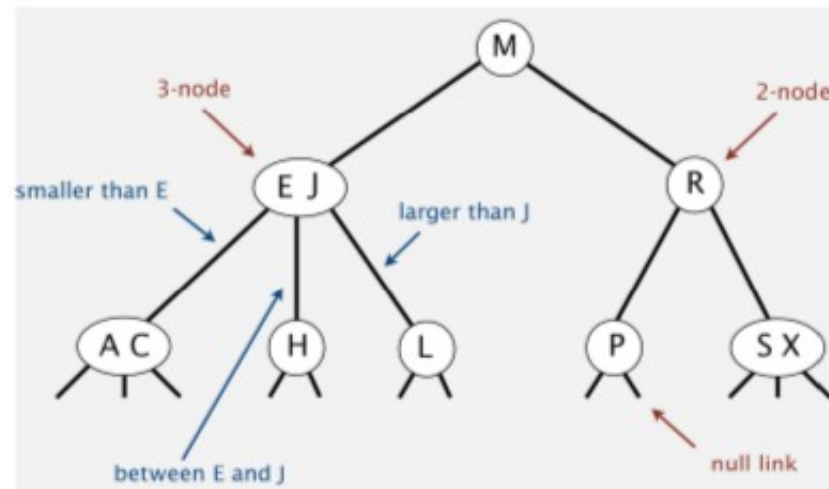
Árvores *multi-way*(*d*-node)

- Um ou mais elementos por nó:
 - $e_1, e_2, e_3, \dots, e_{d-1}$
 - $e_1 \leq e_2 \leq e_3 \leq \dots, e_{d-1}$
- Dois ou mais filhos por nó:
 - $f_1, f_2, f_3, \dots, f_d$
- Árvore de busca:
 - Sub-árvore f_i contém elementos maiores que e_{i-1} e maiores e_i .
- Balanceamento:
 - Crescimento de “baixo” para “cima”



2-3 trees

- Apenas nós com 1 ou 2 elementos
 - 2-node: um elemento e dois filhos
 - 3-node: dois elementos e três filhos
- Todas as folhas possuem a mesma profundidade (balanceamento perfeito)
- Crescimento de baixo para cima



2-3 trees

Tree height.

- Worst case: $\lg N$. [all 2-nodes]
- Best case: $\log_3 N \approx .631 \lg N$. [all 3-nodes]
- Between 12 and 20 for a million nodes.
- Between 18 and 30 for a billion nodes.

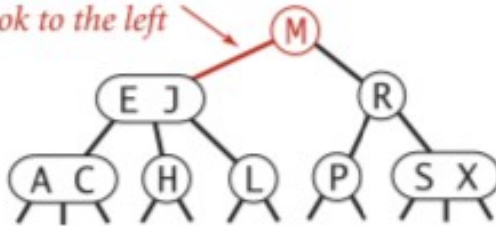
Guaranteed **logarithmic** performance for search and insert.

2-3 trees

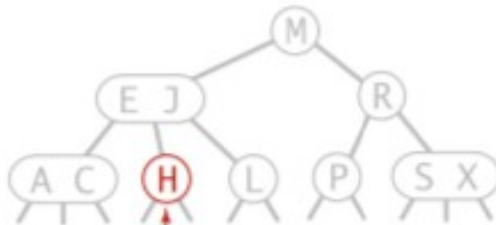
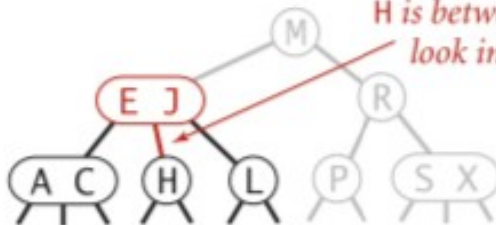
• Busca

successful search for H

H is less than M so
look to the left



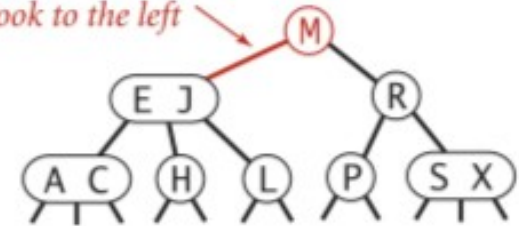
H is between E and J so
look in the middle



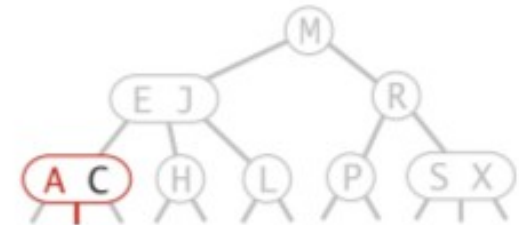
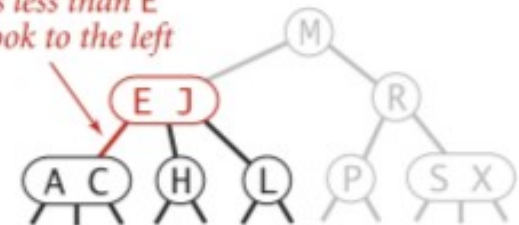
found H so return value (search hit)

unsuccessful search for B

B is less than M so
look to the left



B is less than E
so look to the left



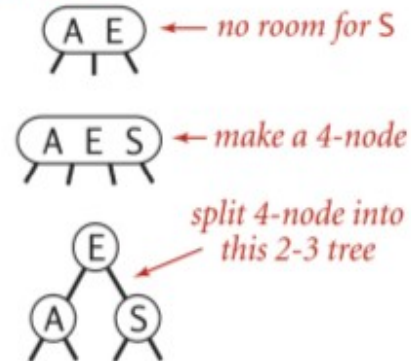
B is between A and C so look in the middle
link is null so B is not in the tree (search miss)

2-3 trees

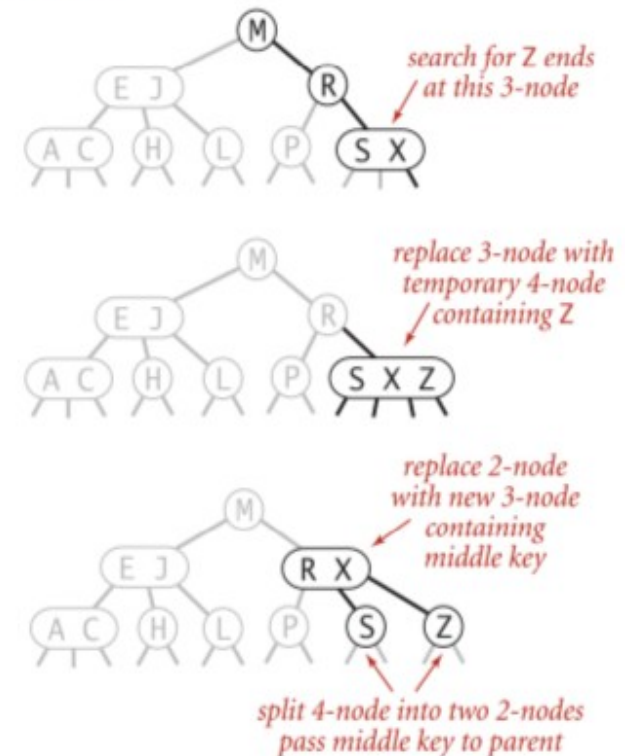
- Inserção

- Busca da posição correta
- Inserção no nó folha
 - 1-node: fácil
 - 2-node:
 - Temporariamente forma um 3-node
 - Operador *split* no 3-node

inserting S

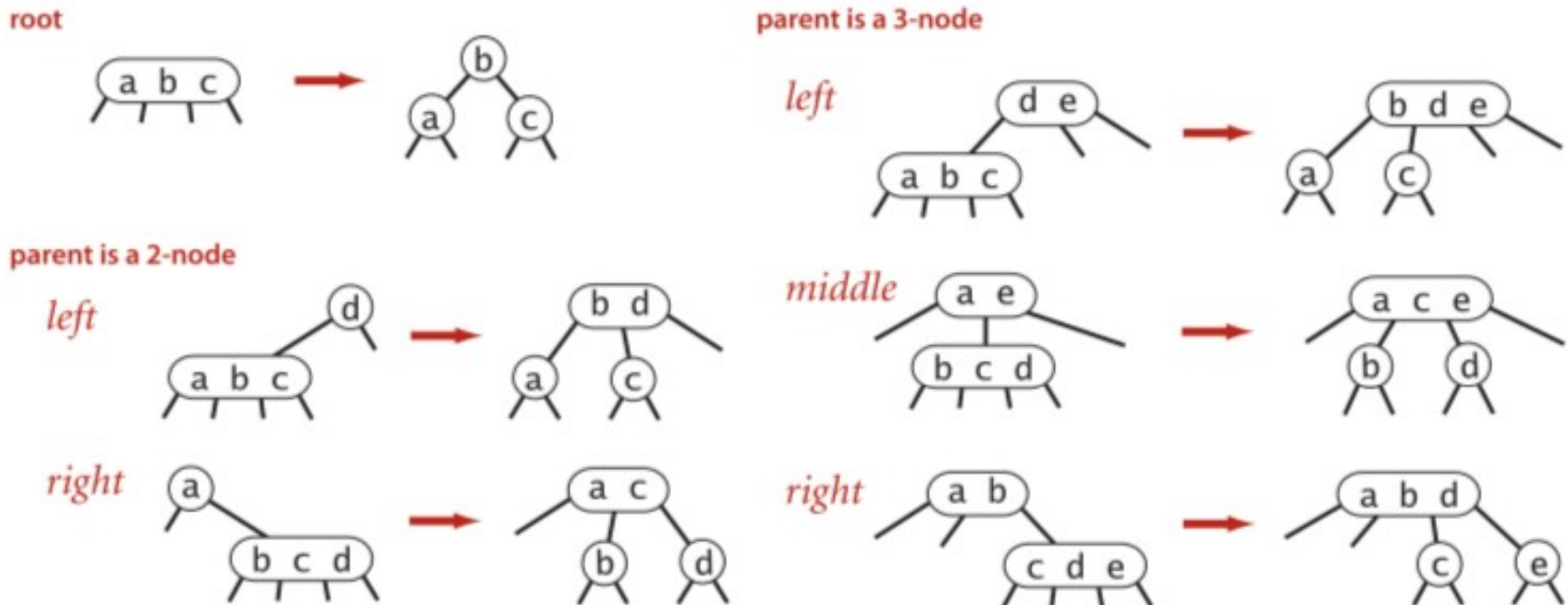


inserting Z



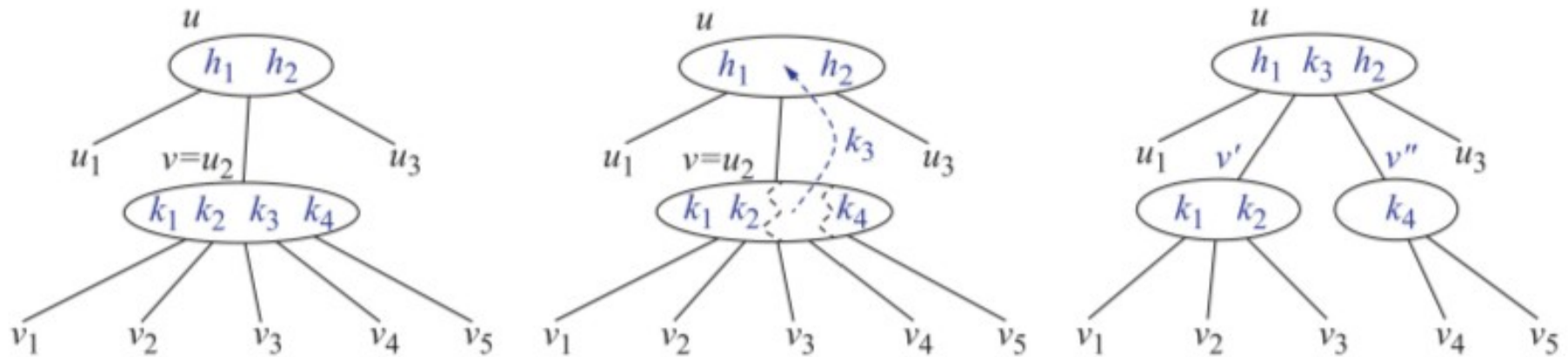
2-3 trees

- Inserção:



2-4 trees

- Formada por nós: 2-node, 3-node e 4-node
 - No máximo 4 filhos
- Busca: fácil
- Inserção: generalização da 2-3 tree

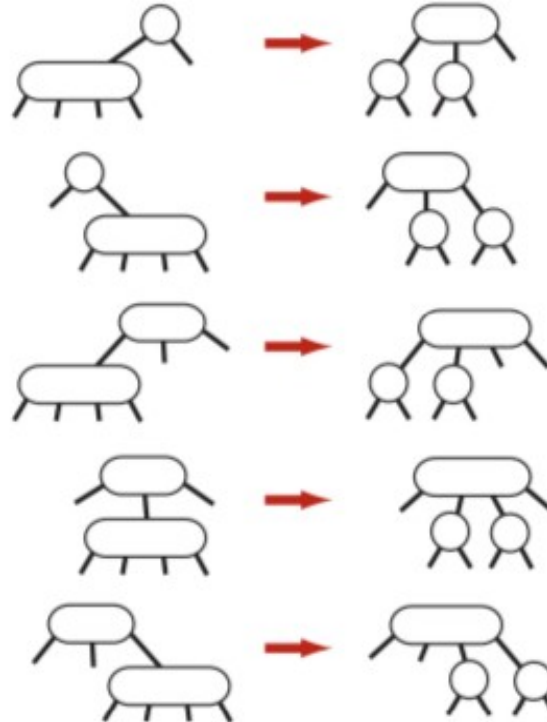


2-4 trees

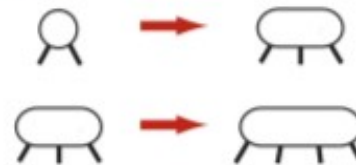
at the root



on the way down



at the bottom



Transformations for insert
in top-down 2-3-4 trees

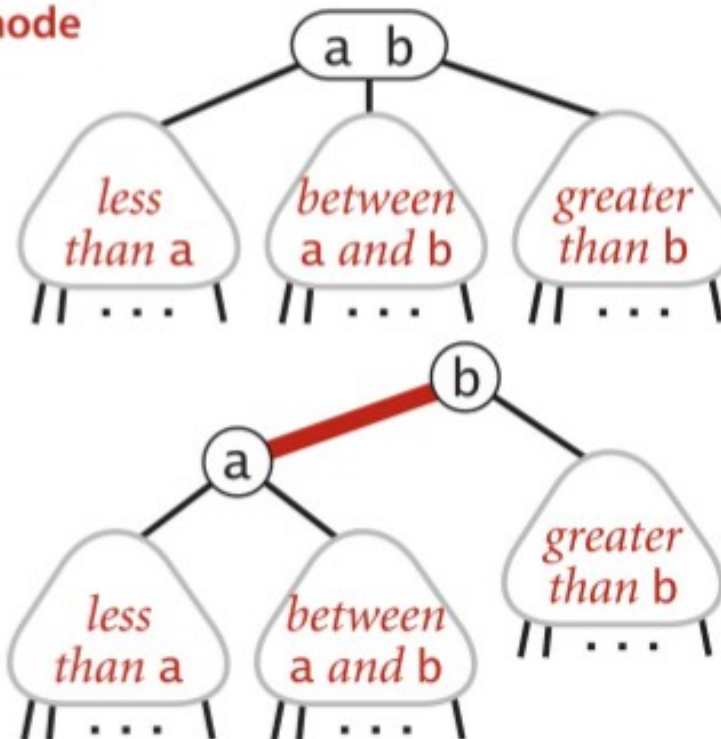
Árvores *Red-Black*

- Árvore binária de busca
- Correspondência direta com a 2-4 tree
- *Left-learning* RB-tree (Sedgewick)
 - Correspondência direta com a 2-3 tree
- Armazenamento de uma flag binária (red/black)
- Associação dos nós vermelhos e pretos com cada d-node das árvores 2-3 ou 2-4
- Balanceamento perfeito considerado apenas ascendentes pretos (perfect black balance)

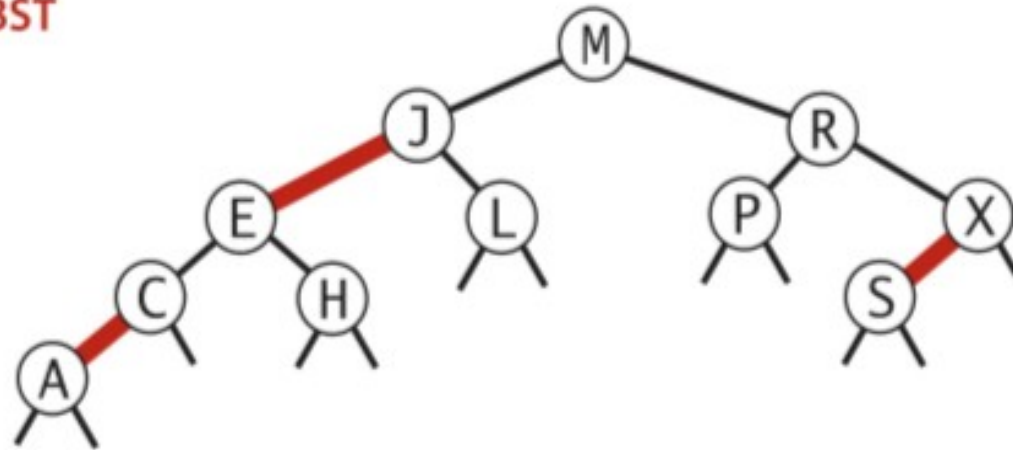
Árvores *Red-Black*

- Nós vermelhos sempre a esquerda de um nó preto;
- Filho de nós vermelhos são sempre pretos

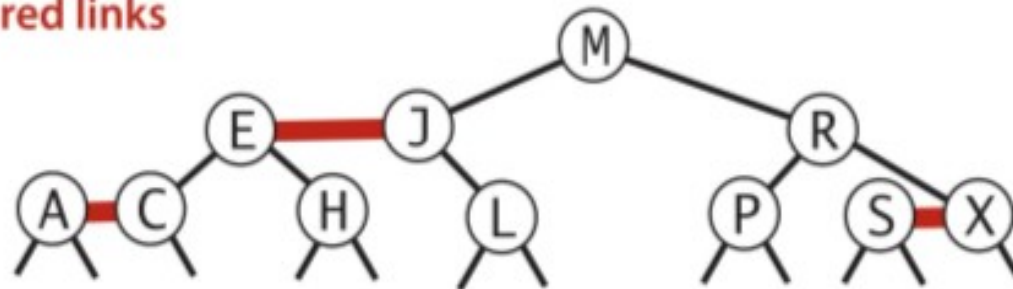
3-node



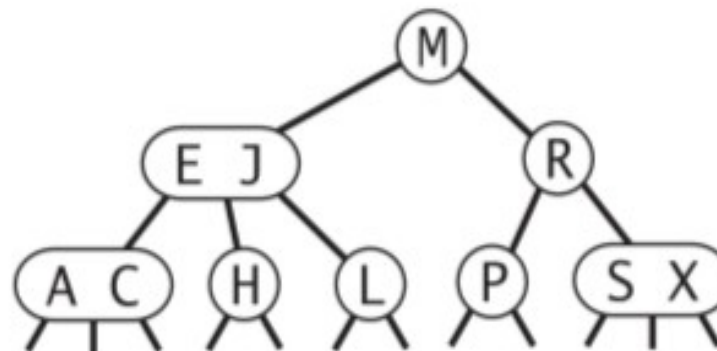
red-black BST



horizontal red links

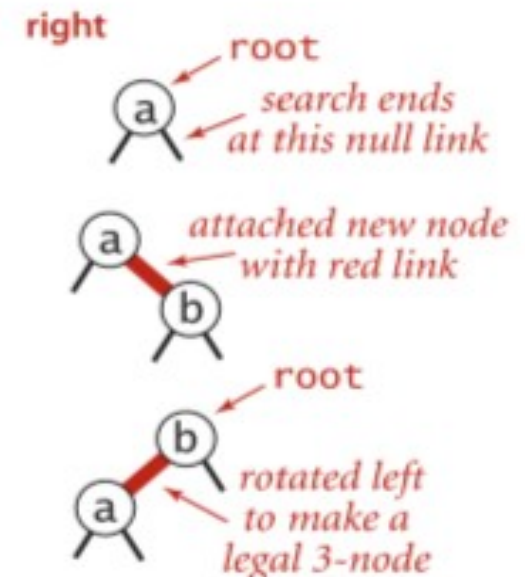
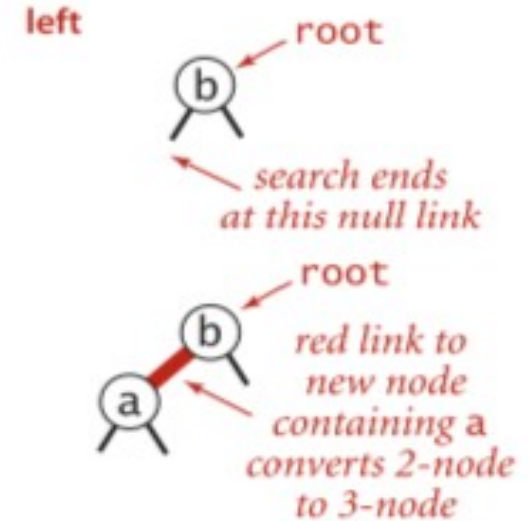


2-3 tree



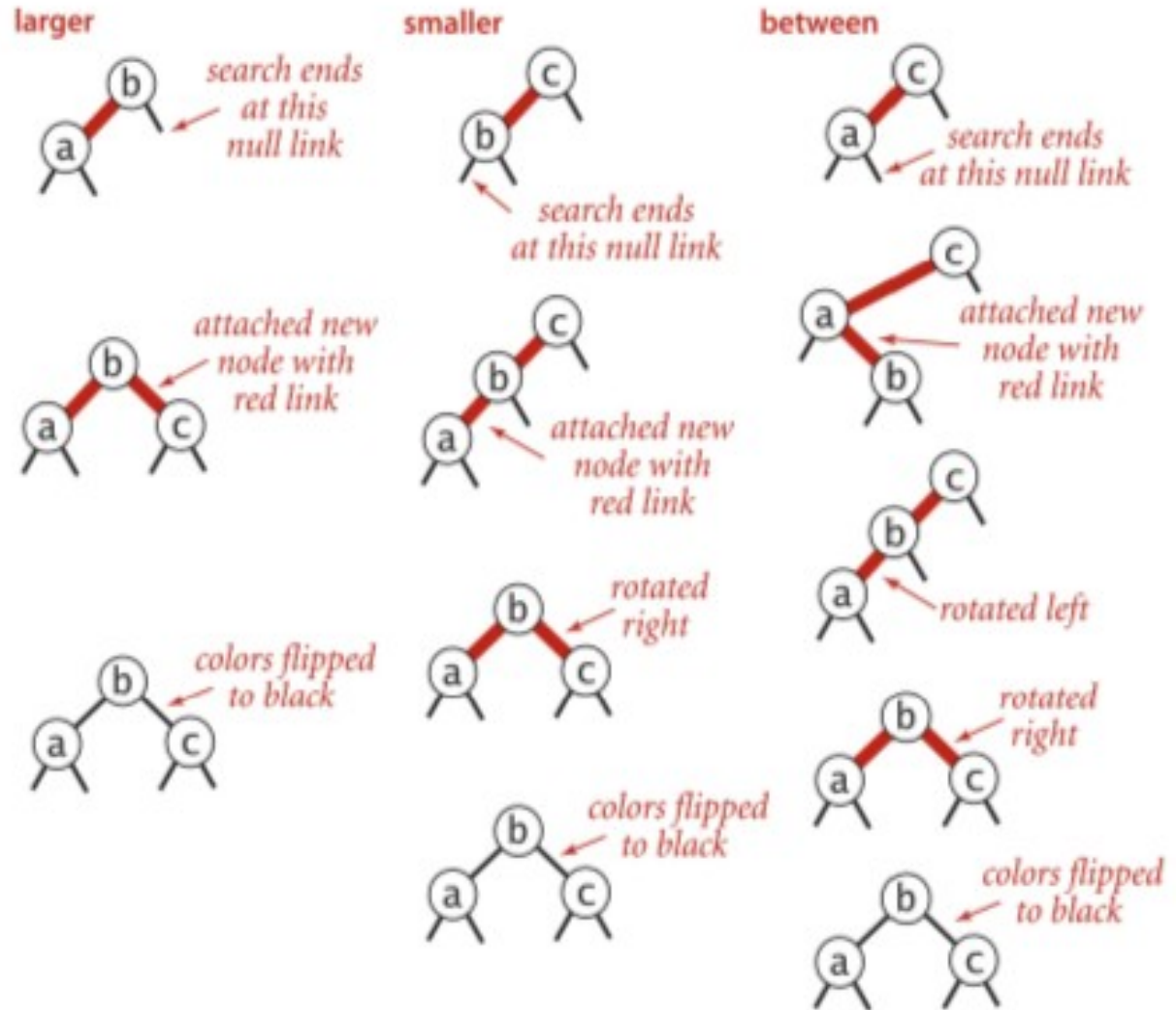
Árvores *Red-Black*

- Inserção
 - 2-node
 - 2 casos (era direto na 2-3)
 - 3-node
 - 3 casos



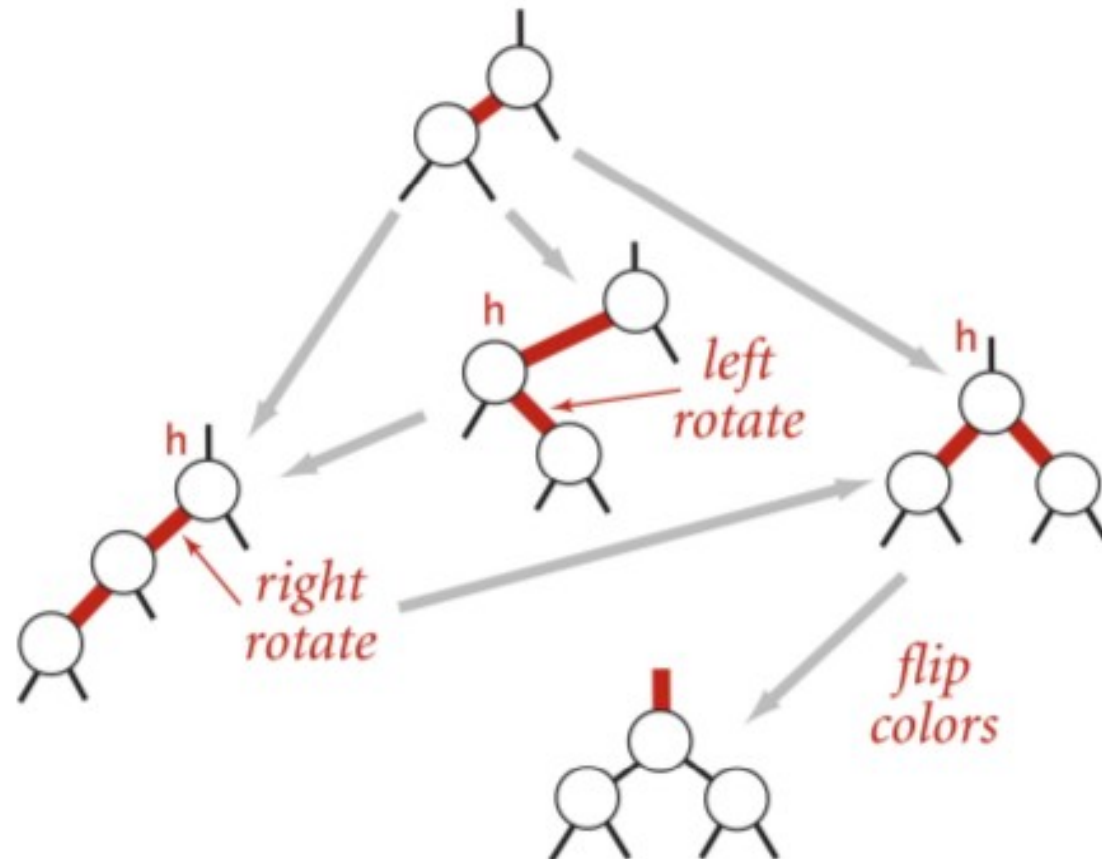
Árvores *Red-Black*

- Inserção
 - 2-node
 - 2 casos
 - 3-node
 - 3 casos



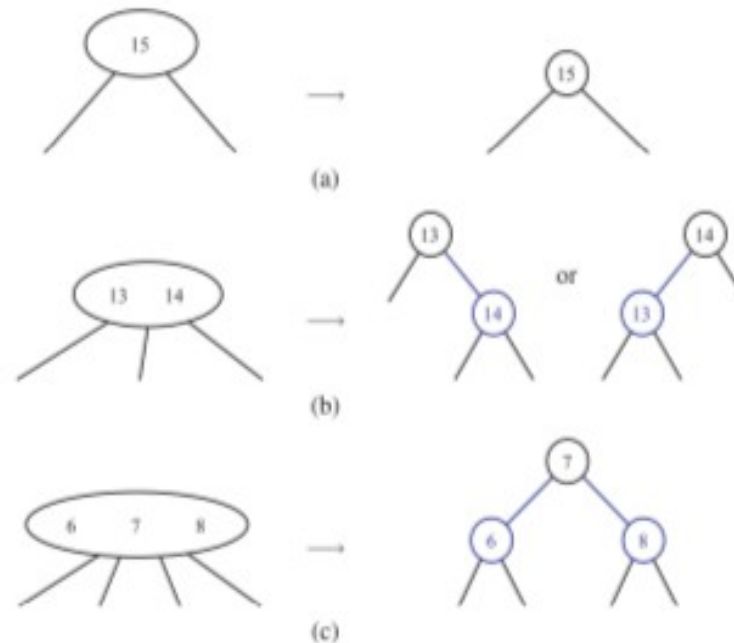
Árvores *Red-Black* (*2-3 trees*)

- Raiz sempre preta
- Resumo da inserção



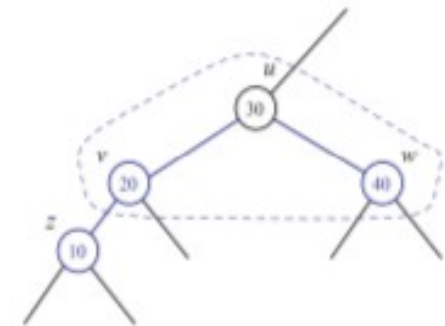
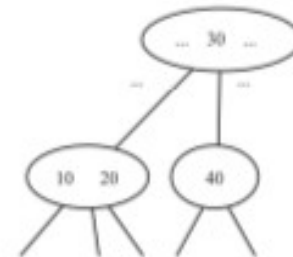
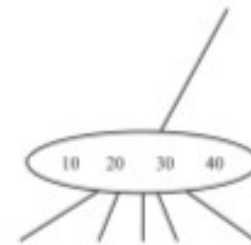
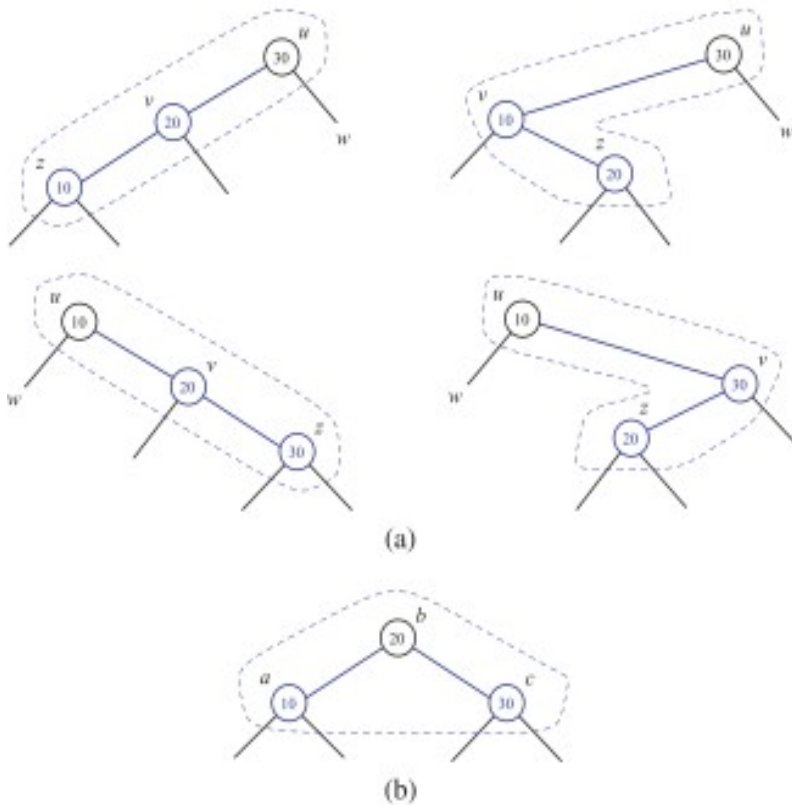
Árvores *Red-Black* (2-4 trees)

- Correspondência com árvores 2-4
- Balanceamento perfeito considerando ascendentes pretos
- Filhos de nós vermelhos são pretos
- Novos casos:

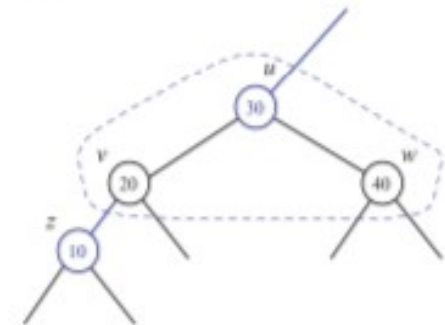


Árvores *Red-Black*

- Inserção



(a)



(b)

DÚVIDAS?

