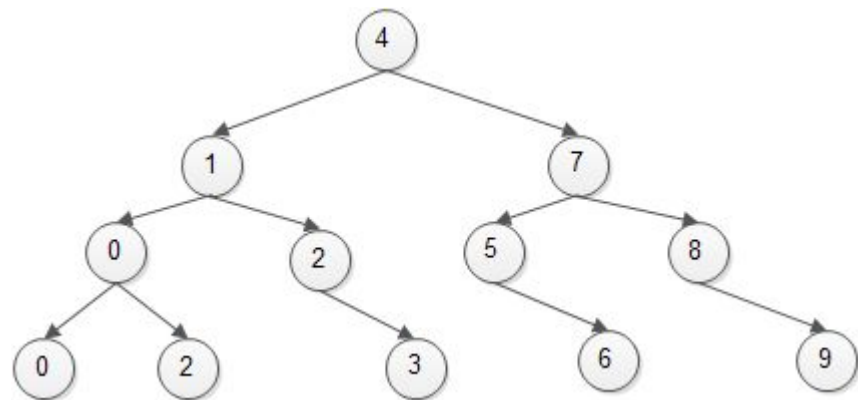


ÁRVORES AVL

CONCEITO

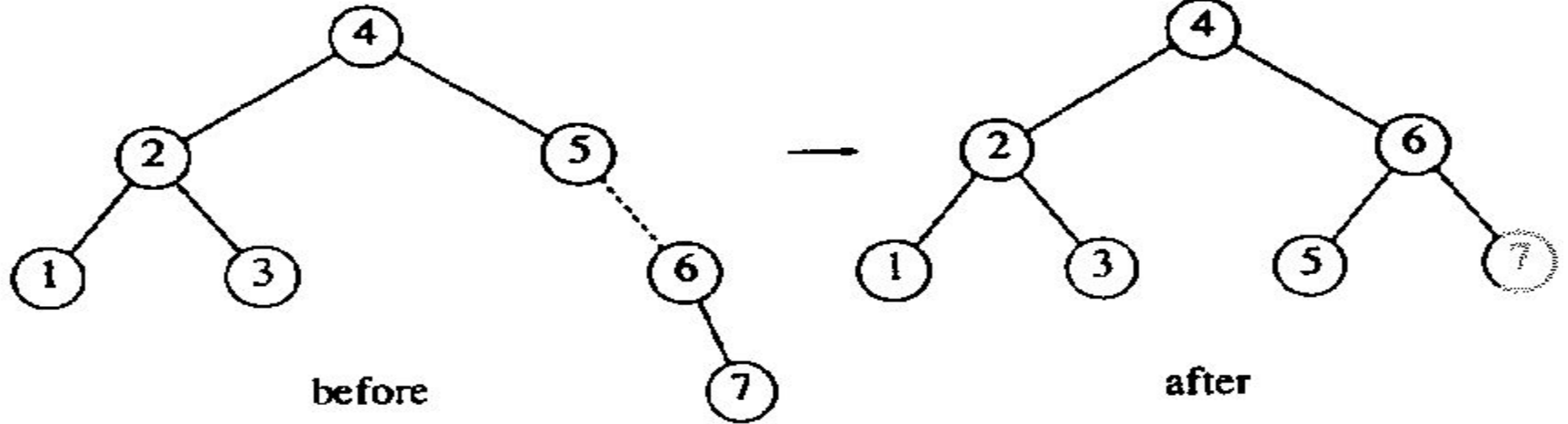
- Uma árvore binária balanceada com base na diferença da altura entre seus nós à direita e à esquerda.



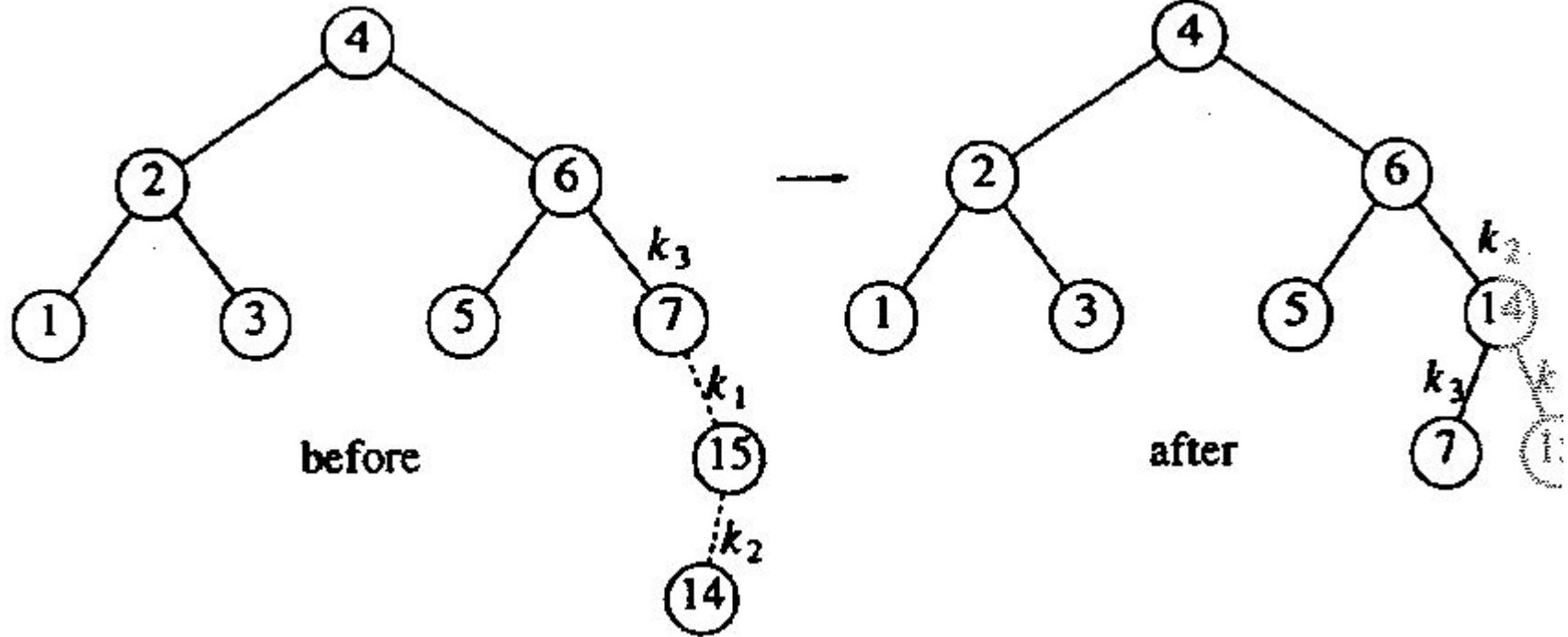
BALANCEAMENTO DA ÁRVORE AVL

- Se as operações de inserção ou remoção causarem desbalanceamento, a árvore deve ser balanceada.
- A árvore está desbalanceada quando a diferença de altura entre seu nó direito e esquerdo é maior do que 1 ou menor que -1.
- O balanceamento é realizado através de rotações.

ROTAÇÃO SIMPLES(ESQUERDA)



ROTAÇÃO DUPLA(DIREITA-ESQUERDA)



INSERÇÃO

- A primeira etapa é igual a inserção em árvores binárias não balanceadas.
- Após a inserção, a altura dos seus nós ascendentes é alterada, alterando também o fator de balanceamento.
- Se houver desbalanceamento é realizada a rotação no primeiro nó encontrado desbalanceado.

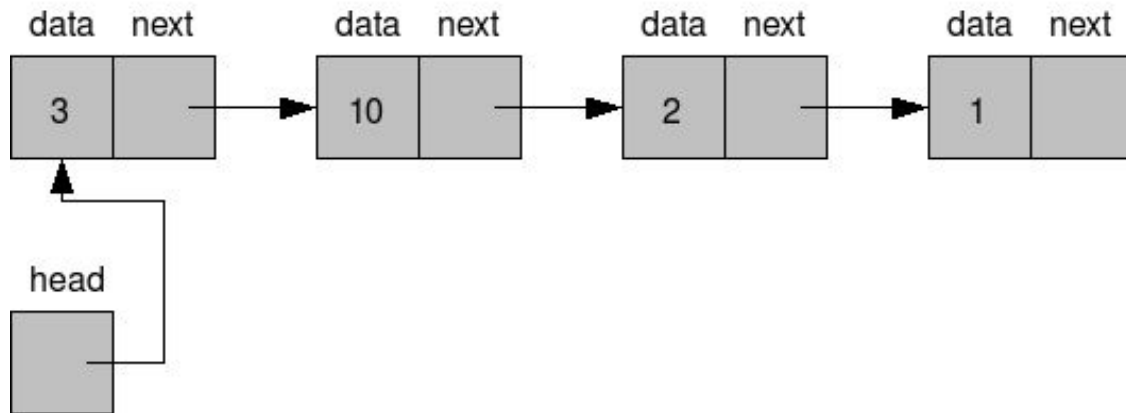
REMOÇÃO

- A primeira etapa é igual em árvores binárias não balanceadas.
- Assim como na inserção, as alturas e fator de balanceamento são alterados, porém em caso de precisar de rebalanceamento ele deve seguir até chegar na raiz.
- Se o fator de balanceamento ficar maior que 2 ou menor que -2, as operações de rotação são realizadas.

APLICAÇÃO DA ÁRVORE AVL

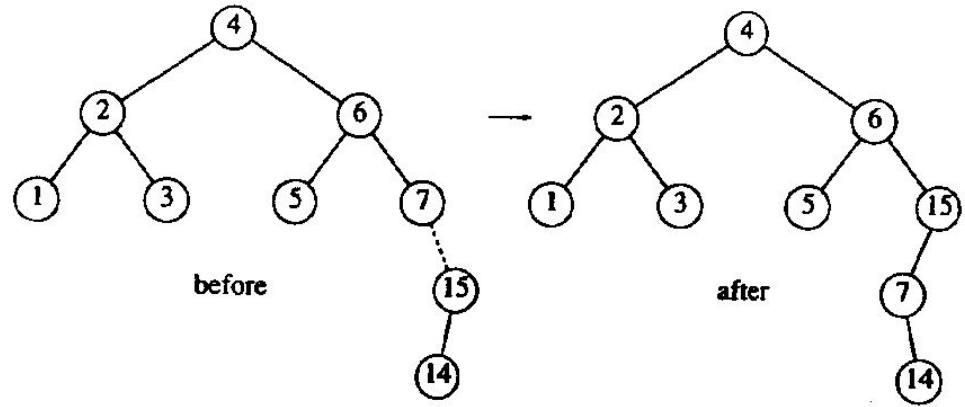
- Ordenação de Nomes e Matrículas de alunos da UFG, dentro de uma amostragem de dados com cerca de 11000 alunos.
- A ordenação, a princípio, é feita pela matrícula (chave principal de cada nó) em ordem crescente.
- O principal objetivo é mostrar que uma árvore avl (corretamente **BALANCEADA**) pode ser extremamente eficiente em certas operações, quando comparada com a estrutura de lista ligada (**estrutura de dado vista em AED1**)

LISTA LIGADA (SINGLE LINKED LIST)



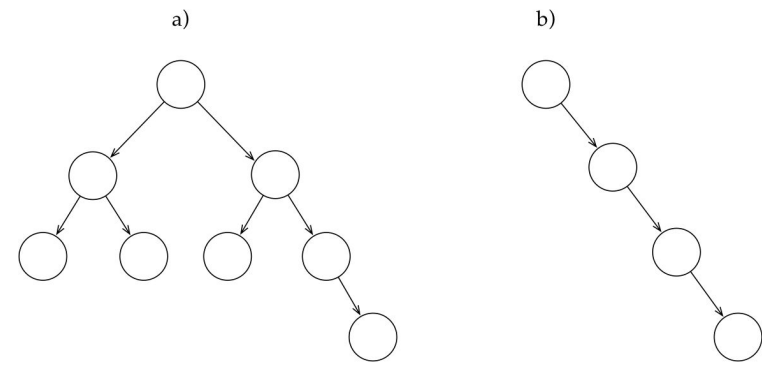
- Complexidade Temporal da função de inserção: $O(n)$
- Complexidade Temporal da função de busca (inclusive a busca recursiva em uma lista ligada): $O(n)$
- Complexidade Temporal da função de remoção: $O(n)$

ÁRVORE AVL



- Complexidade Temporal da função de inserção: $O(n \cdot \log_2 n)$
- Complexidade Temporal da função de busca: $O(\log_2 n)$
- Complexidade Temporal da função de remoção: $O(\log_2 n)$

E QUANDO A ÁRVORE NÃO FICA BALANCEADA?



Podem haver momentos no qual o balanceamento da árvore não pode ser feito. Nesse tipo de caso, se uma certa quantidade de nós é somente inserida em uma lado da árvore, a tendência é que ela se pareça cada vez mais com uma lista ligada. Dessa forma, a complexidade de cada função fica linear $O(n)$.

COMPARAÇÃO DO TEMPO DE EXECUÇÃO

Árvore AVL

Tempo para montar:

17 606 950 nanossegundos

Tempo para buscar nó:

2 144 nanossegundos

Lista Ligada

Tempo para montar:

276 878 992 nanossegundos

Tempo para buscar nó:

57 744 nanossegundos

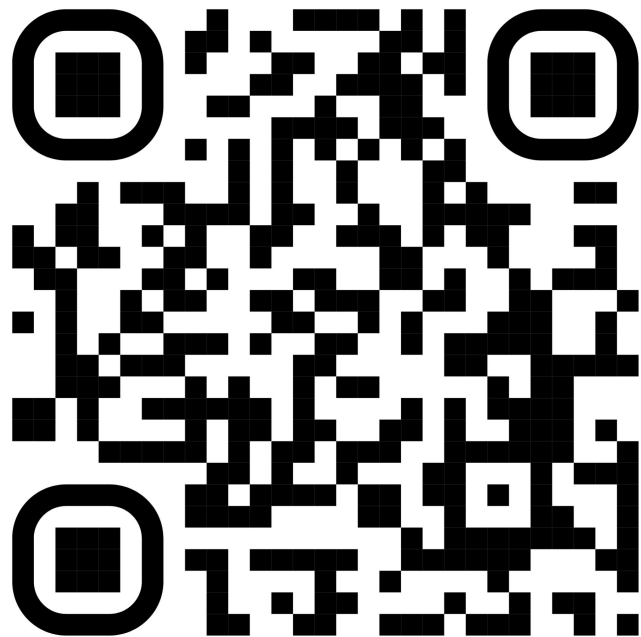
COMPILEI MEU CÓDIGO, MAS...

Não se preocupe se o tempo de execução variar em alguns testes do seu código, isso é totalmente normal.

Toda vez que o programador compila o código e o executa, o tempo de execução sempre será diferente!

O escalonamento de processos da sua CPU é diferente a cada momento, o que aconteceu foi que em um certo momento o processo de execução do seu código pode ter sido priorizado ou não em relação a outro processo.

QR-CODE PARA ACESSAR O REPOSITÓRIO



A close-up portrait of Dilma Rousseff, the former President of Brazil. She has short, curly brown hair and is wearing a dark red patterned top. Her expression is serious and slightly stern. The background is a light blue wall with a faint geometric pattern.

FIM DO SLIDE!

**ACHO BOM TU BATER
PALMAS E NÃO FAZER
PERGUNTAS!**