



**Alunos: André Gustavo, Ronaldo Crispim, Pâmella e Thales**

# **ÁRVORE SPLAY**

**Estrutura de Dados II**



# DEFINIÇÃO DA ÁRVORE SPLAY OU ESPALHADA

Uma árvore splay é uma árvore de busca binária com uma propriedade extra, onde que os elementos recentemente acessados são arranjados de forma que eles possam rapidamente serem acessados novamente.

Como árvores de busca binárias de auto-equilíbrio, uma árvore splay executa operações básicas, como inserção, pesquisa e remoção. Com a notação Big-O do pior caso dessa estrutura de dados sendo o  $O(\log n)$ .

\*O acesso pode ser qualquer uma das operações de uma árvore binária: inserção, pesquisa e remoção.



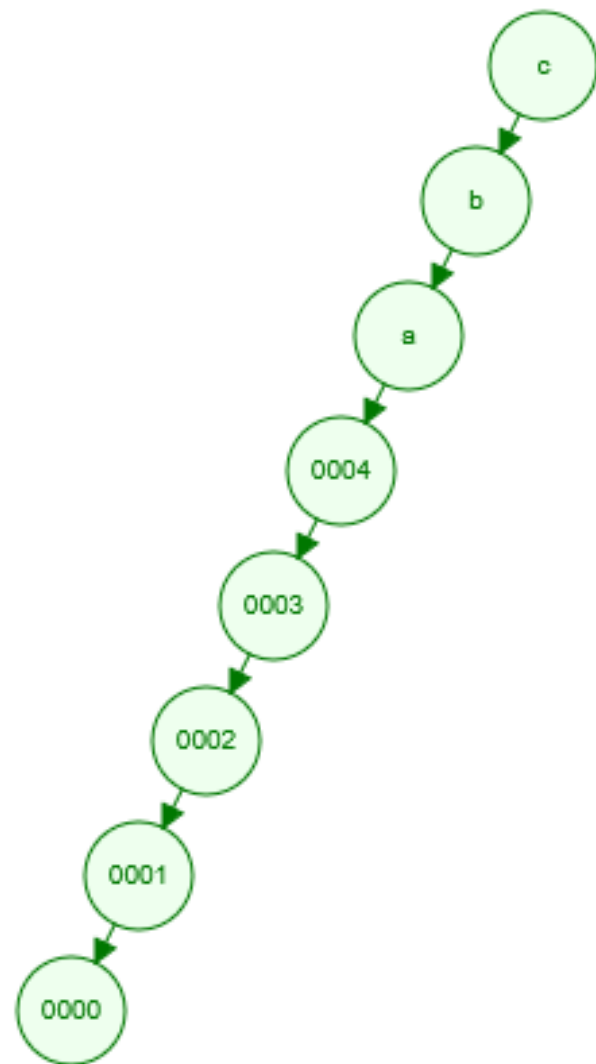
# As operações da árvore splay

Operação	Condição	Movimento
Zig esquerda	Alvo é o filho direito da raiz	Rotação à esquerda
Zig direita	Alvo é o filho esquerdo da raiz	Rotação à direita
Zig-Zig esquerda	Alvo e o pai são filhos direitos do avô	Duas rotações à esquerda
Zig-Zig Direita	Alvo e o pai são filhos esquerdos do avô	Duas rotações à direita
Zig-Zag esq-dir	Pai é filho direito e alvo é filho esquerdo	Rotação direita do pai, esquerda do avô
Zig-Zag dir-esq	Pai é filho esquerdo e alvo é filho direito	Rotação esquerda do pai, direita do avô

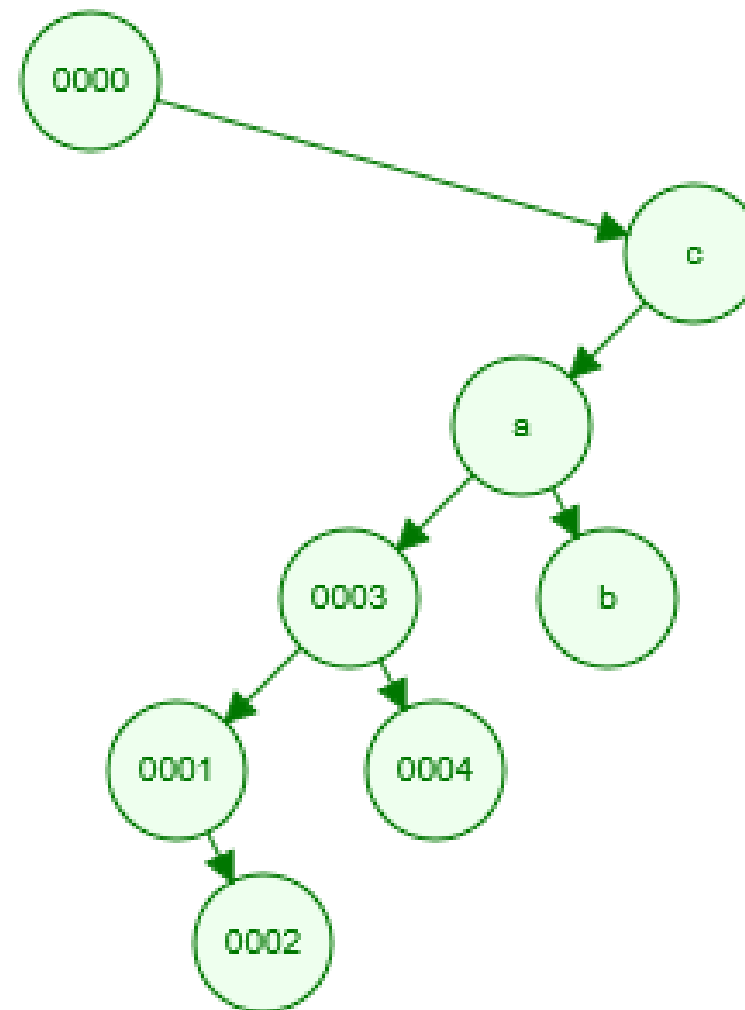


# Funcionamento

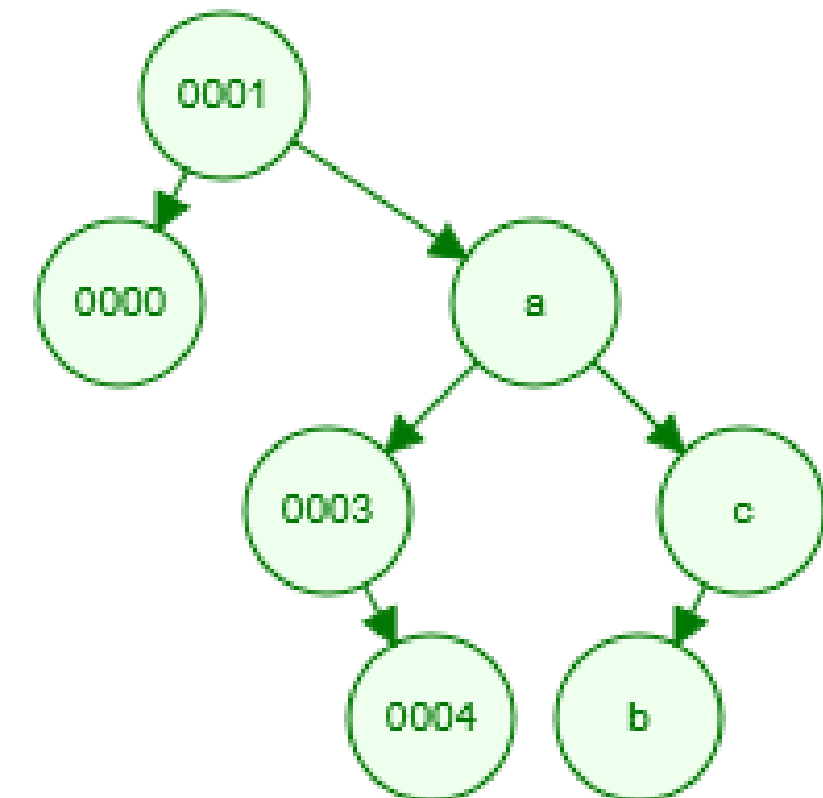
Criação da árvore, somente inserção dos elementos a ela.



Busca pelo elemento 0, usando as operações de zig, zig-zig e zig-zag.



Exclusão de um determinado elemento, aqui usaremos o 2.





# Vantagens

- Desempenho comparável: O desempenho do caso médio é tão eficiente quanto o de outras árvores.
- Pequeno consumo de memória: As árvores Splay não precisam armazenar nenhum dado contábil.

Lembrando que o bom desempenho de uma árvore splay depende do fato de ela ser auto otimizada, ou seja, de que os nós acessados com frequência se moverão para mais perto da raiz, onde poderão ser acessados mais rapidamente.





# Desvantagens

- A desvantagem mais significativa das árvores splay é que a altura de uma árvore splay pode ser linear. Por exemplo, este será o caso após acessar todos os  $n$  elementos em ordem não decrescente. Como a altura de uma árvore corresponde ao pior caso de tempo de acesso, isso significa que o custo real de uma única operação pode ser alto.
- A representação de árvores splay pode mudar mesmo quando acessadas de forma "somente leitura" (ou seja, por operações de busca). Isso complica o uso dessas árvores splay em um ambiente multithread. Especificamente, gerenciamento extra é necessário se várias threads puderem executar operações de busca simultaneamente. Isso também as torna inadequadas para uso geral em programação puramente funcional, embora mesmo assim possam ser usadas de forma limitada para implementar filas de prioridade.
- Por fim, quando o padrão de acesso é aleatório, a sobrecarga de dispersão adicional adiciona um fator constante significativo ao custo em comparação a alternativas menos dinâmicas.



**Obrigado!**