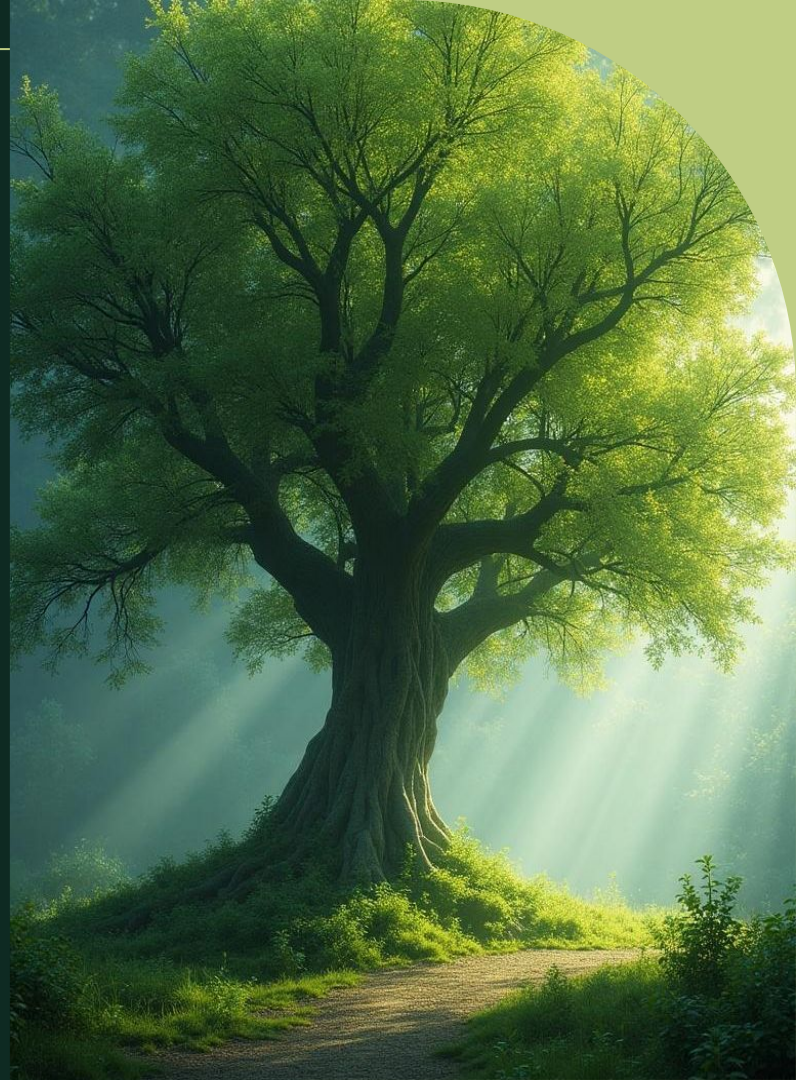


Árvore Patricia

Uma abordagem otimizada para busca e armazenamento de informações.

Integrantes:

Davi Lucas, Gustavo Henrique, Leonardo Henrique, Luiz Felipe, Rafael Soares
e Victor Augusto



Introdução

A árvore Patricia é uma estrutura de dados avançada que permite uma representação eficiente de chaves em sistemas de busca. Neste contexto, abordaremos seu histórico, funcionamento e aplicações práticas.





Histórico

Autor e Publicação

- A árvore Patricia foi criada por Donald R. Morrison e apresentada em 1968 no artigo intitulado "**Practical Algorithm to Retrieve Information Coded in Alphanumeric**". Esta estrutura auxilia a procura de informações em texto grandes como livro, artigos, listas telefônicas, e etc.

Practical

Algorithm

To

Retrieve

Coded

In

Alphanumeric

"Algoritmo Prático para Recuperar Informações Alfanuméricas"

Motivação da Criação



O objetivo principal ao desenvolver a árvore Patricia foi aumentar a eficiência nas buscas de arquivos, especialmente quando se tratava de chaves longas e variáveis, minimizando o espaço de armazenamento ao não manter as chaves completas em cada nó.

02

Funcionamento

Métodos Padrões



A árvore Patricia utiliza métodos de busca, inserção e remoção semelhantes aos das árvores de pesquisa binária. Os métodos são otimizados para comparar caracteres individuais em vez de substrings inteiras, aumentando a eficiência durante as operações.

Métodos Padrões



Busca: Caminha pelos bits/caracteres conforme o índice de decisão armazenado nos nós. Muito eficiente para buscas com prefixo ou igualdade total.

Inserção: Compara os bits ou caracteres da chave com os da árvore. Ao encontrar uma divergência, cria um novo nó no ponto da diferença. Reorganiza os ponteiros para manter a ordem da árvore. Não precisa de balanceamento, pois a estrutura segue os bits da chave como critério.

Remoção: Localiza o nó com a chave desejada. Remove o nó e ajusta o ponteiro do pai. Se um pai ou filho ficar com apenas um filho, pode ser fundido (compactado). Como não é árvore balanceada, não há reestruturação de altura.

Balanceamento: Não precisa ser balanceada! A estrutura é guiada por bits da chave, o que a torna naturalmente eficiente mesmo sem balanceamento.

INSERIR

Inserir a
palavra
"BRASA"

BRASA

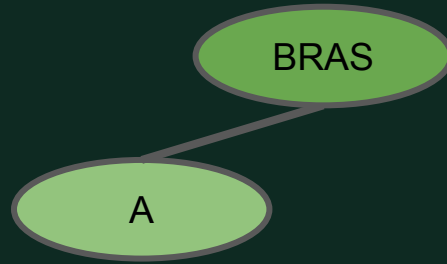
INSERIR

Inserir a
palavra
"BRASIL"

BRASA

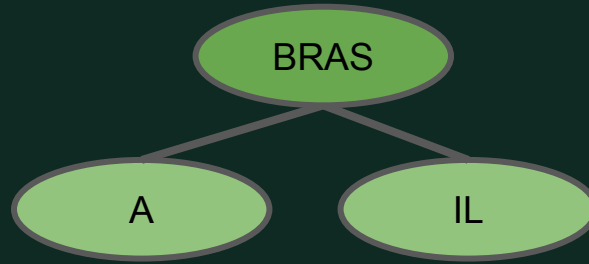
INSERIR

Inserir a
palavra
"BRASIL"



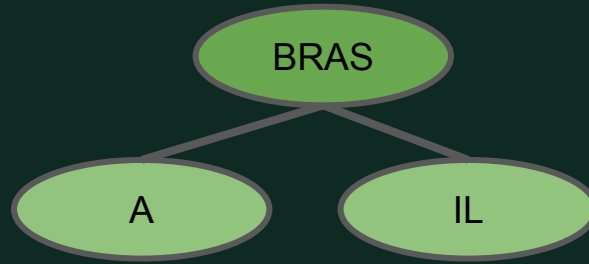
INSERIR

Inserir a
palavra
"BRASIL"



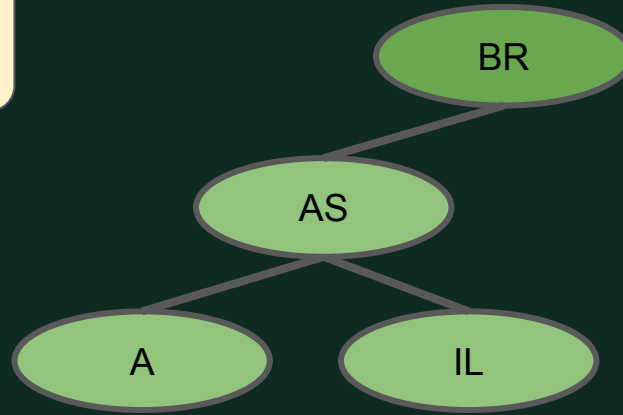
INSERIR

Inserir a
palavra
"BRIGADEIRO"



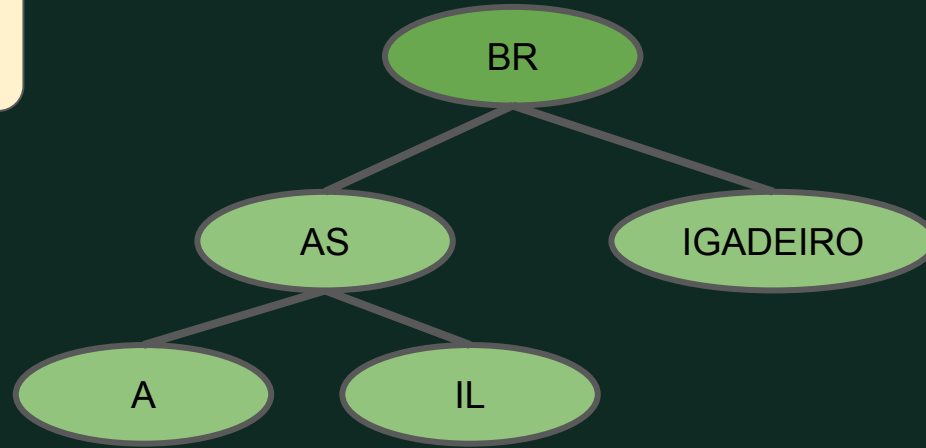
INSERIR

Inserir a
palavra
"BRIGADEIRO"



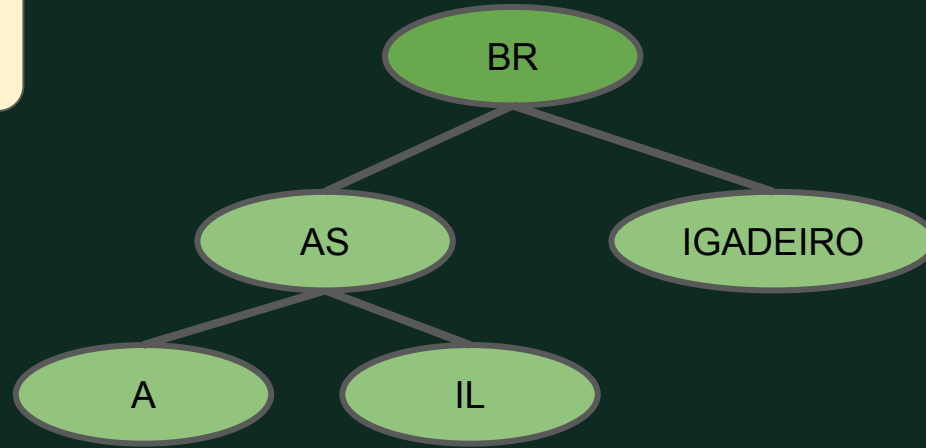
INSERIR

Inserir a
palavra
"BRIGADEIRO"



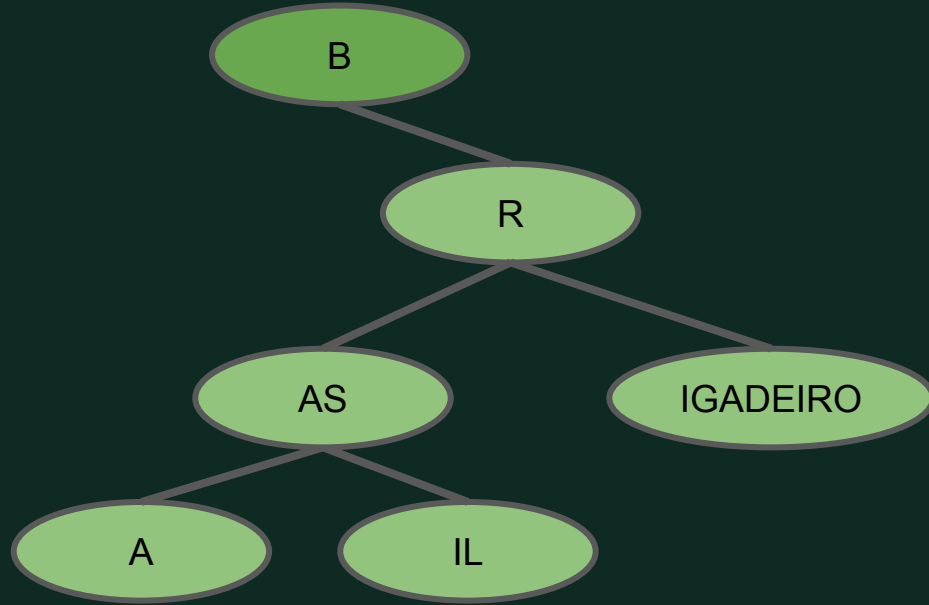
INSERIR

Inserir a
palavra
"BANCO"



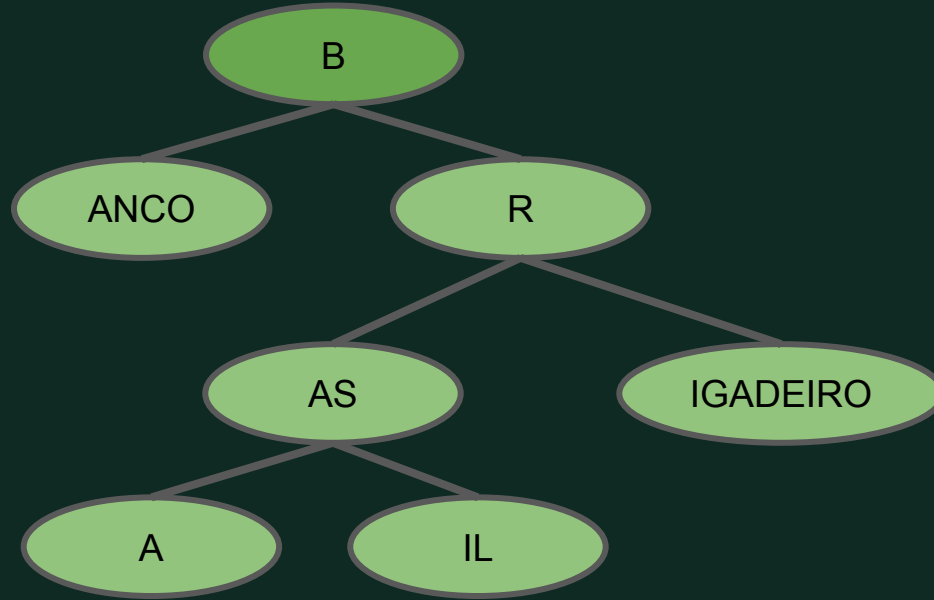
INSERIR

Inserir a
palavra
"BANCO"



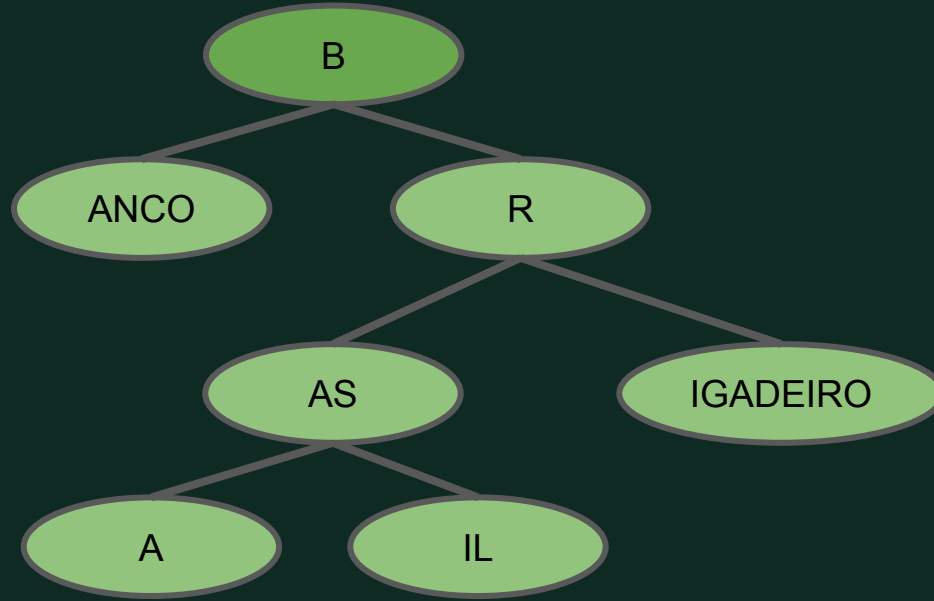
INSERIR

Inserir a
palavra
"BANCO"



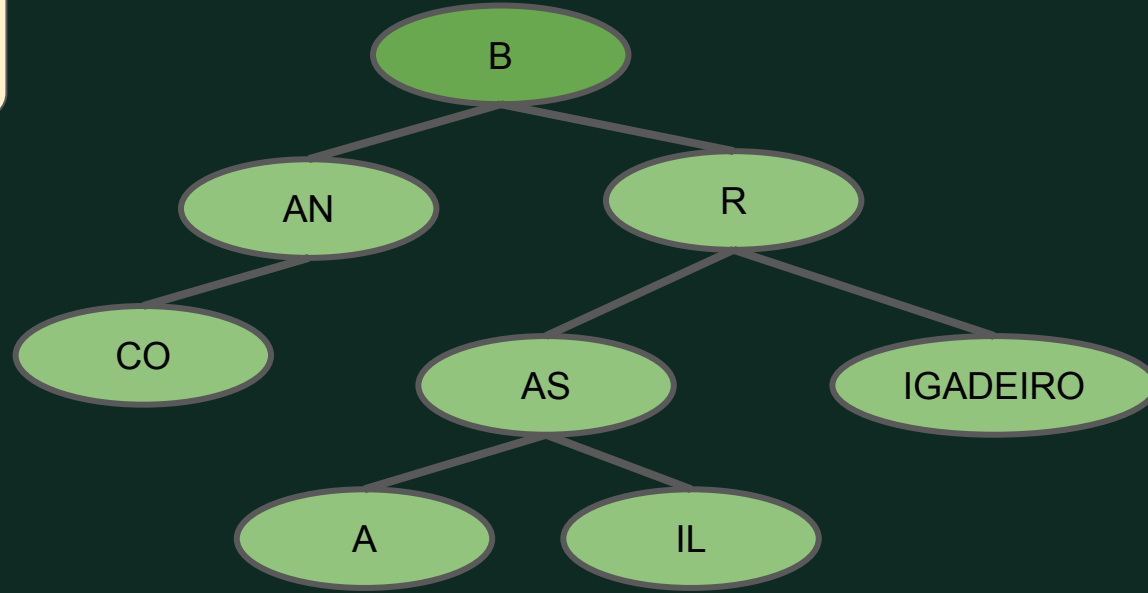
INSERIR

Inserir a
palavra
"BANHISTA"



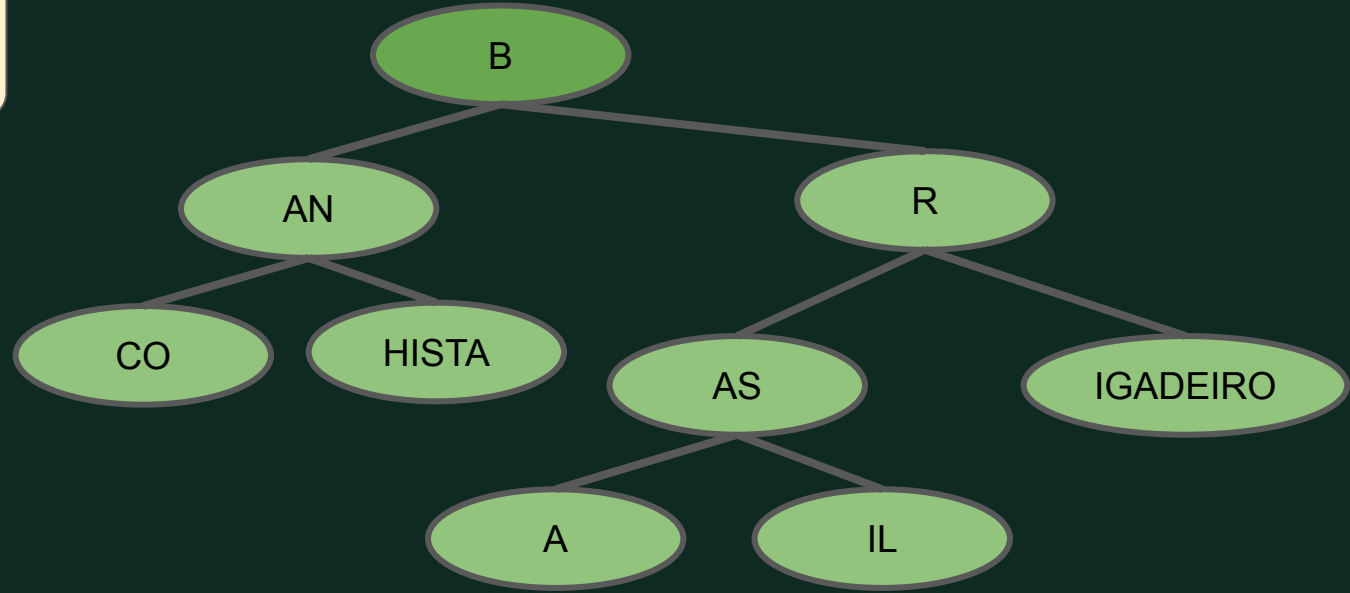
INSERIR

Inserir a
palavra
"BANHISTA"



INSERIR

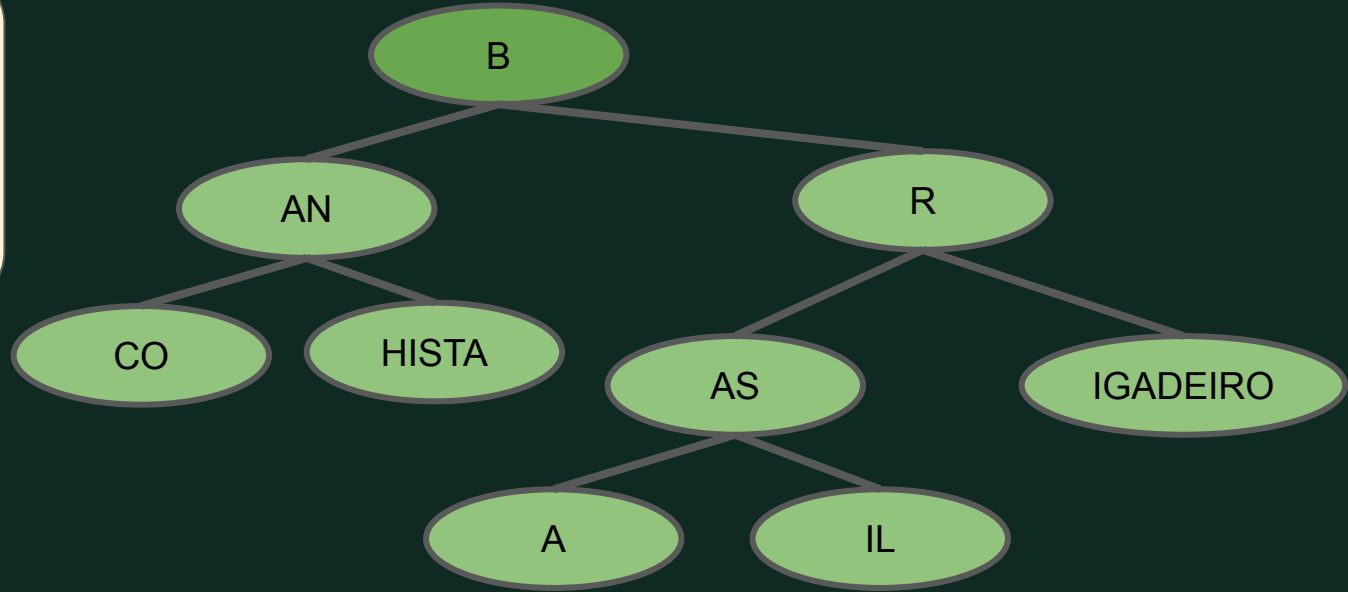
Inserir a
palavra
"BANHISTA"



INSERIR

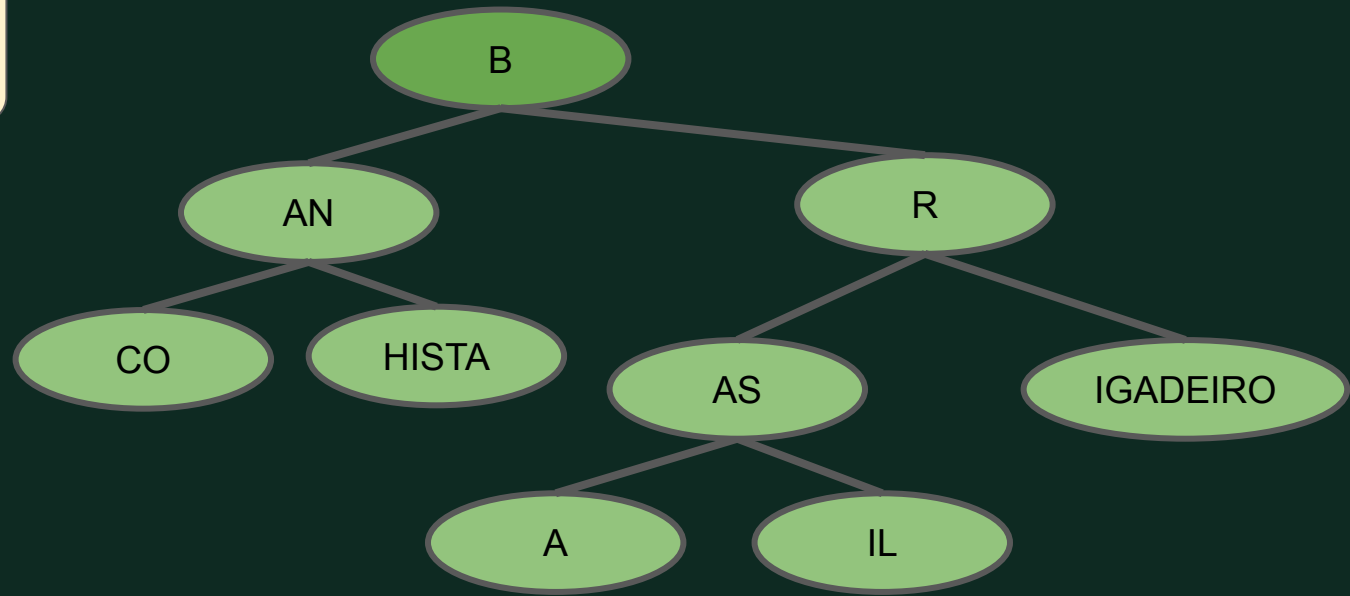
Complexidade:
 $O(t \cdot N)$

t é o tamanho
da palavra
 N é o tamanho
do alfabeto



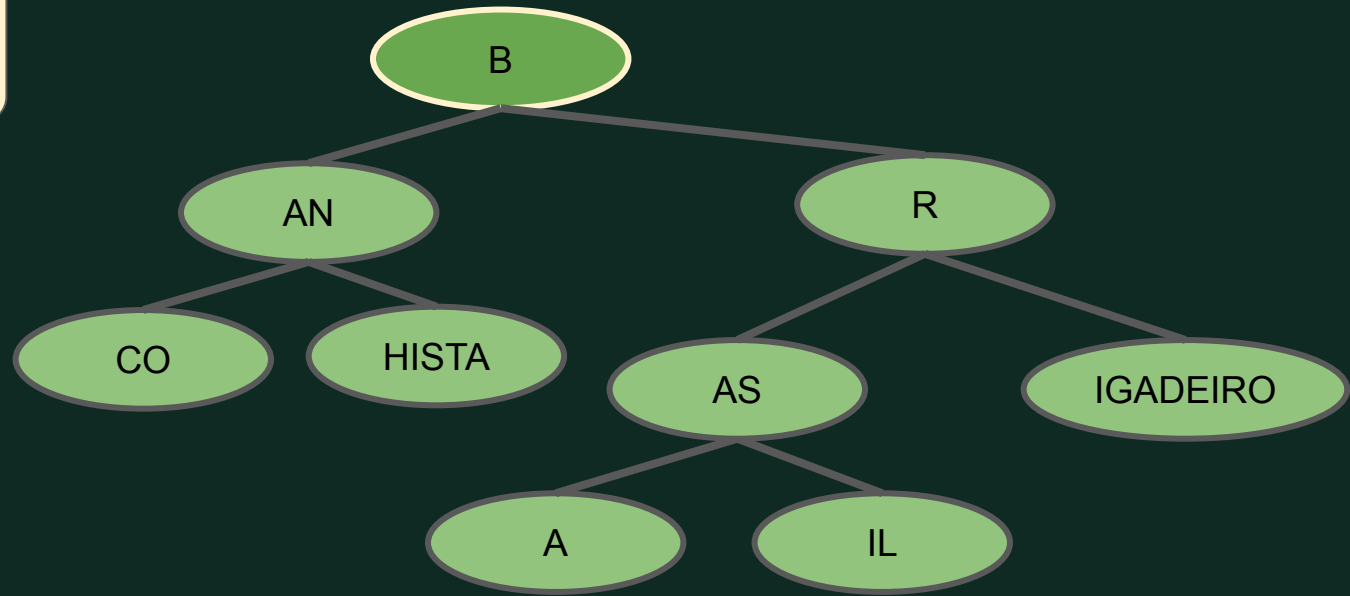
BUSCAR

BUSCAR PALAVRA
"BRASA"



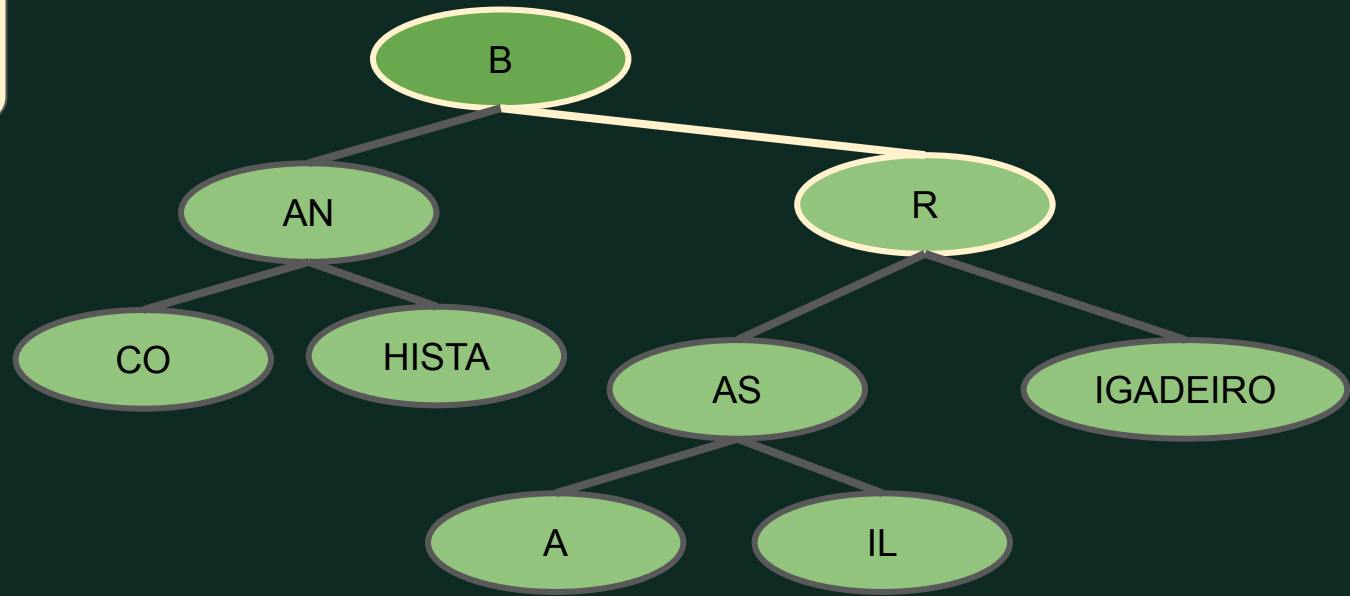
BUSCAR

BUSCAR PALAVRA
"BRASA"



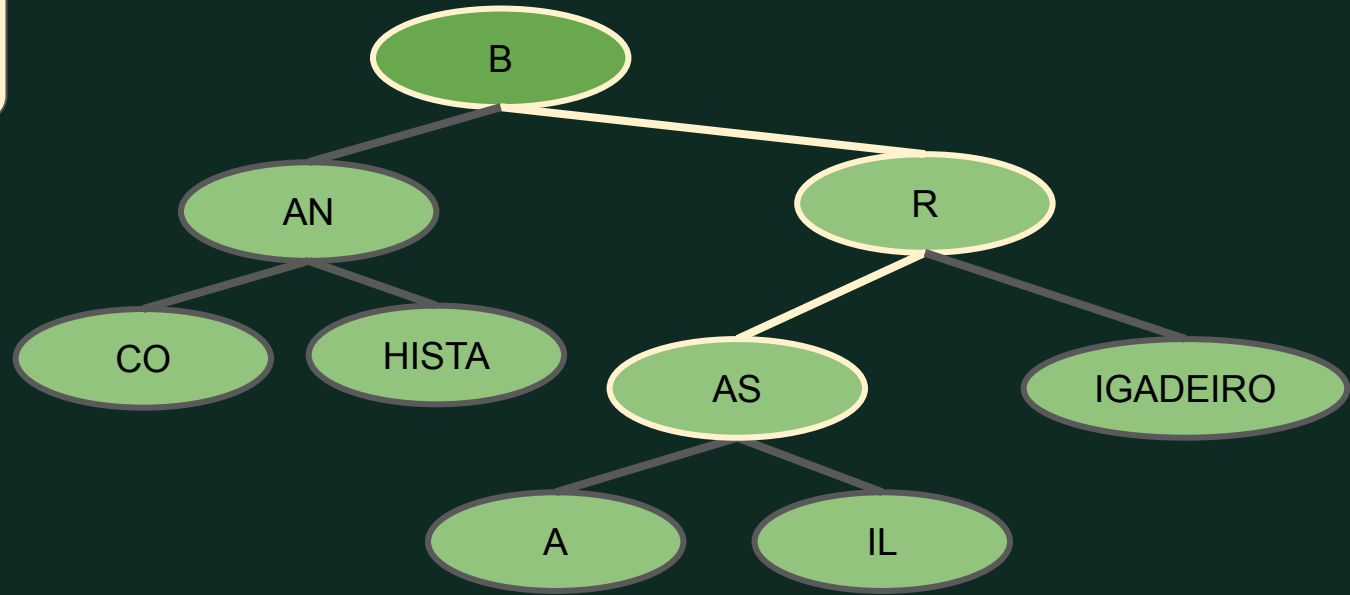
BUSCAR

BUSCAR PALAVRA
"BRASA"



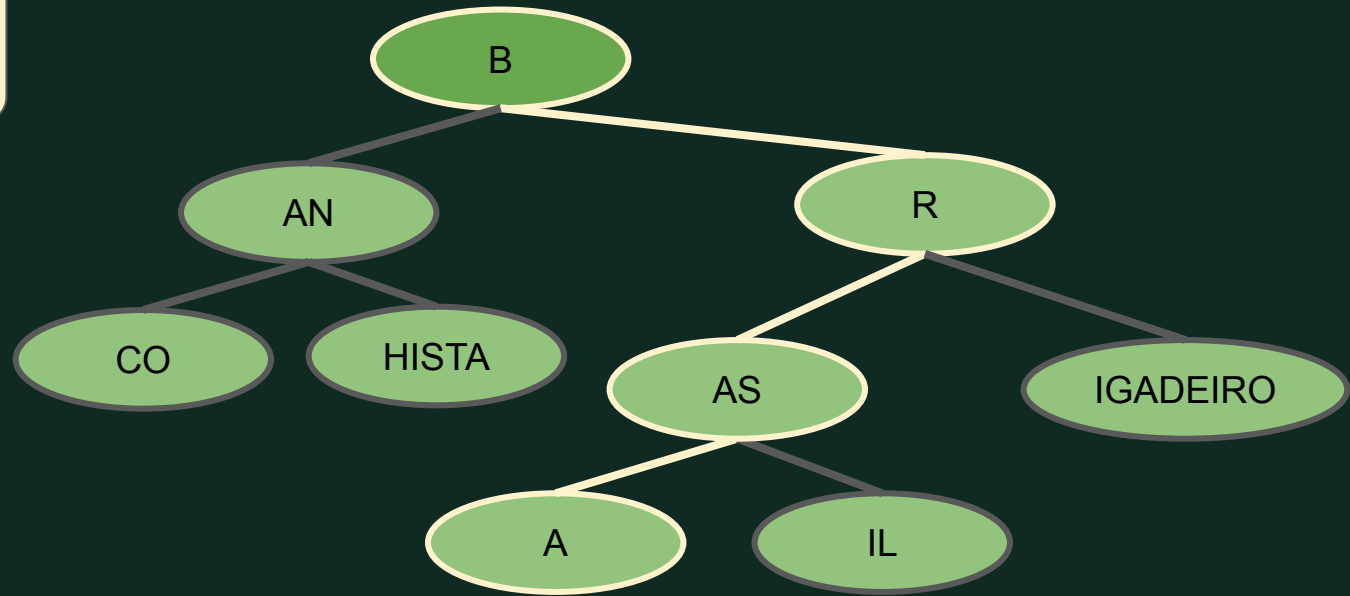
BUSCAR

BUSCAR PALAVRA
"BRASA"



BUSCAR

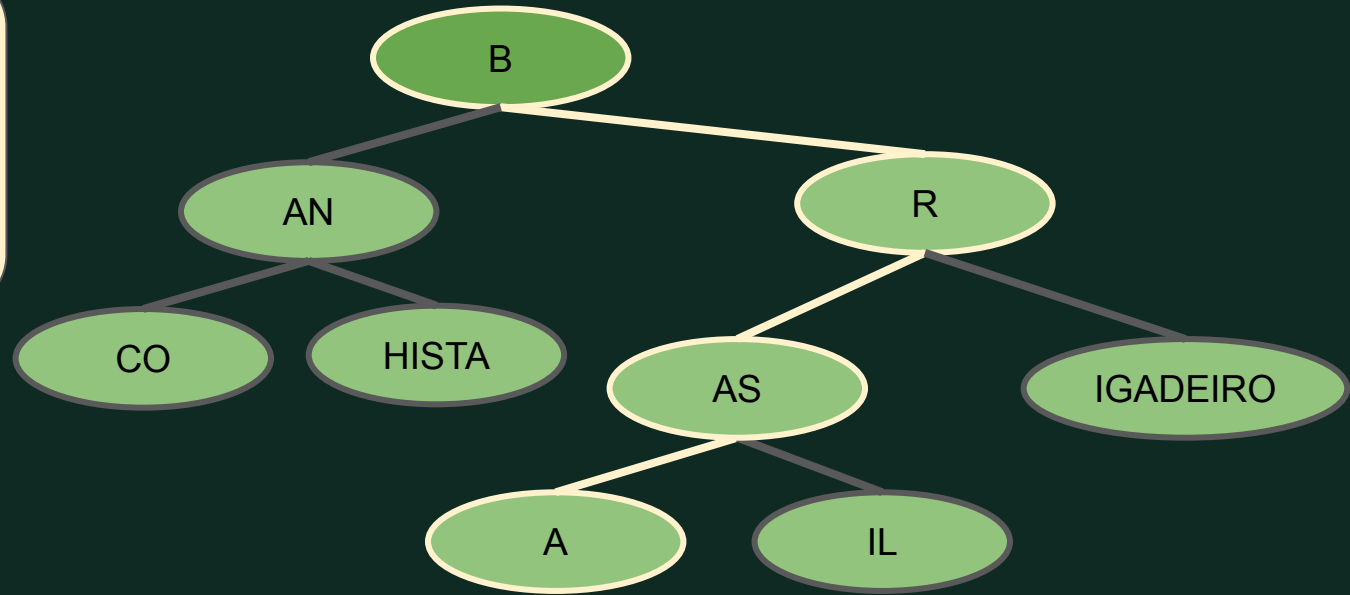
BUSCAR PALAVRA
"BRASA"



BUSCAR

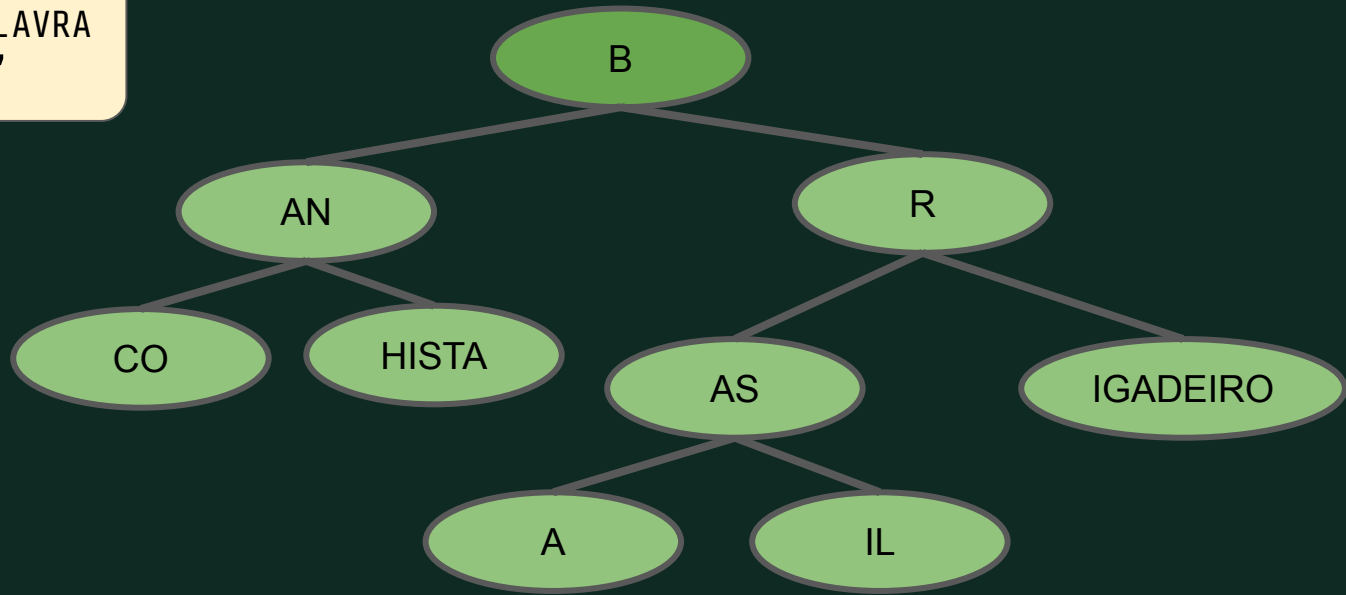
Complexidade:
 $O(t)$

t é o tamanho
da palavra



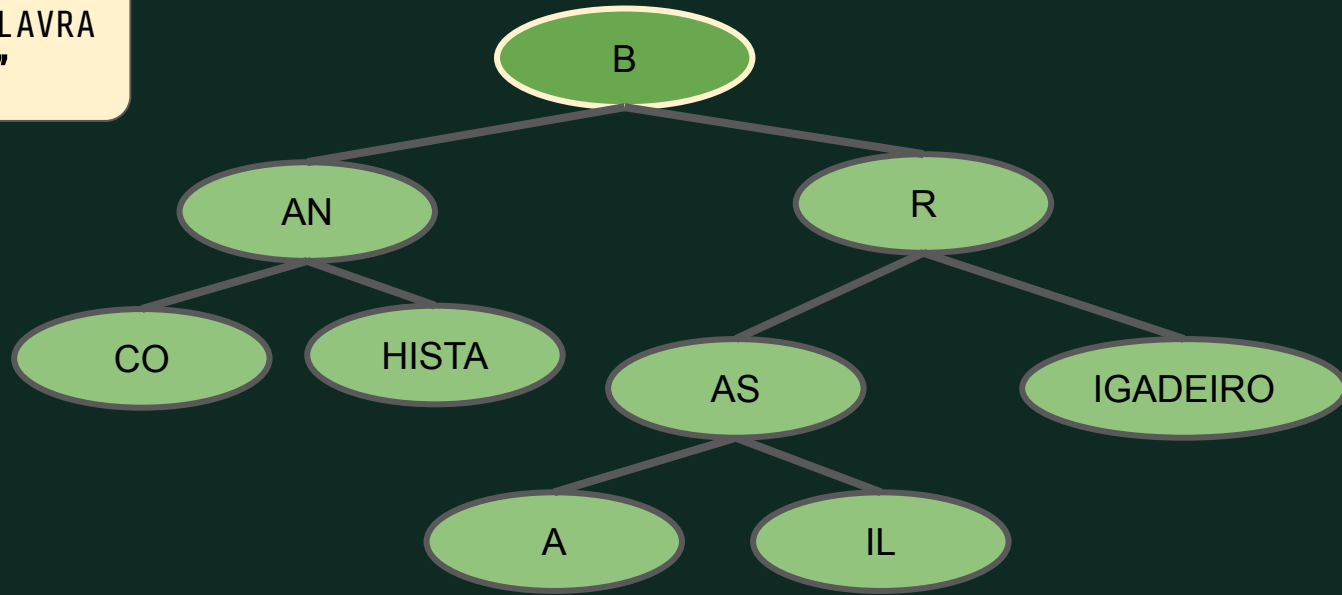
REMOVER

REMOVER PALAVRA
"BRASA"



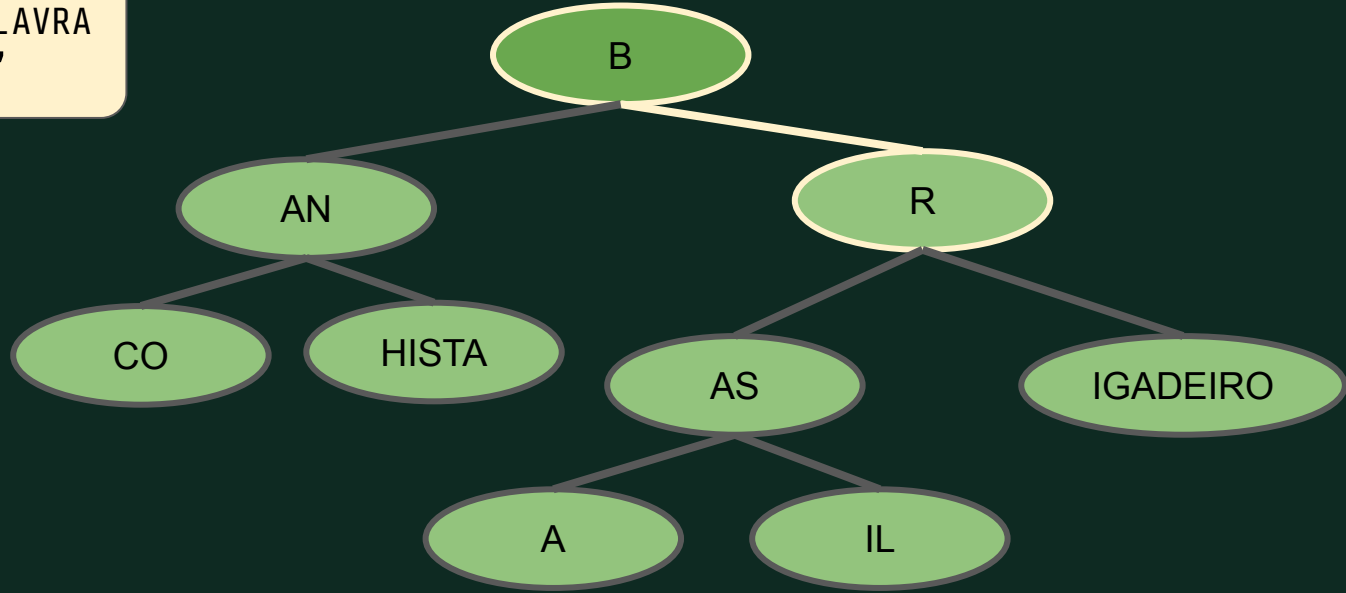
REMOVER

REMOVER PALAVRA
"BRASA"



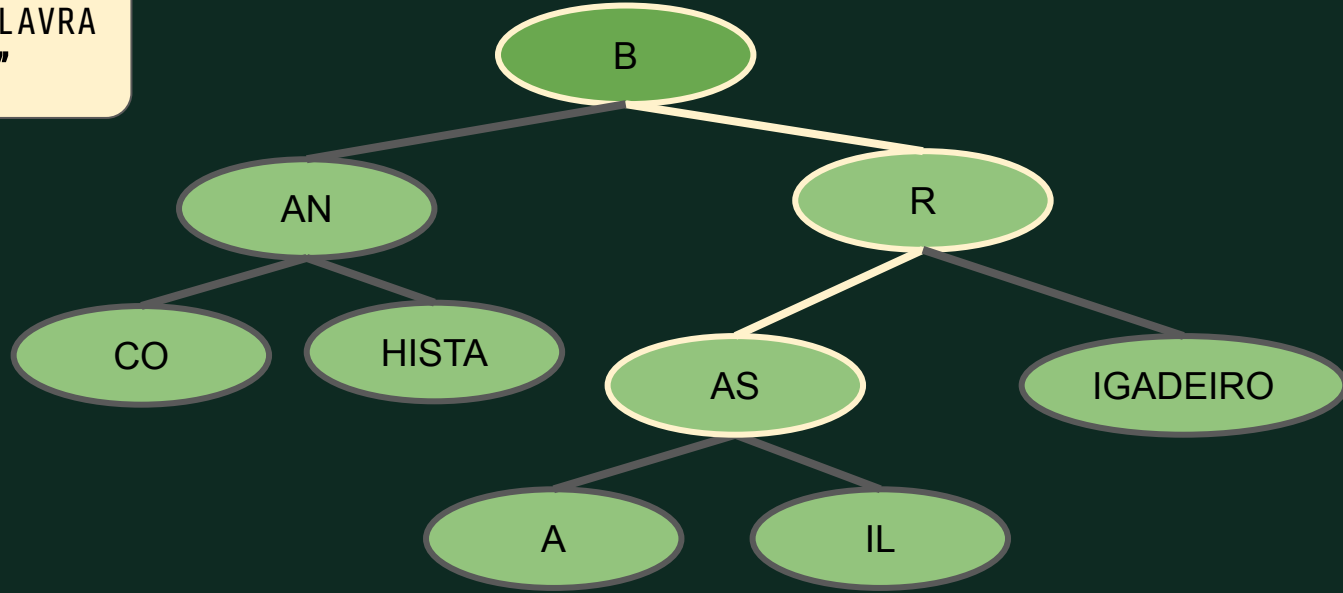
REMOVER

REMOVER PALAVRA
"BRASA"



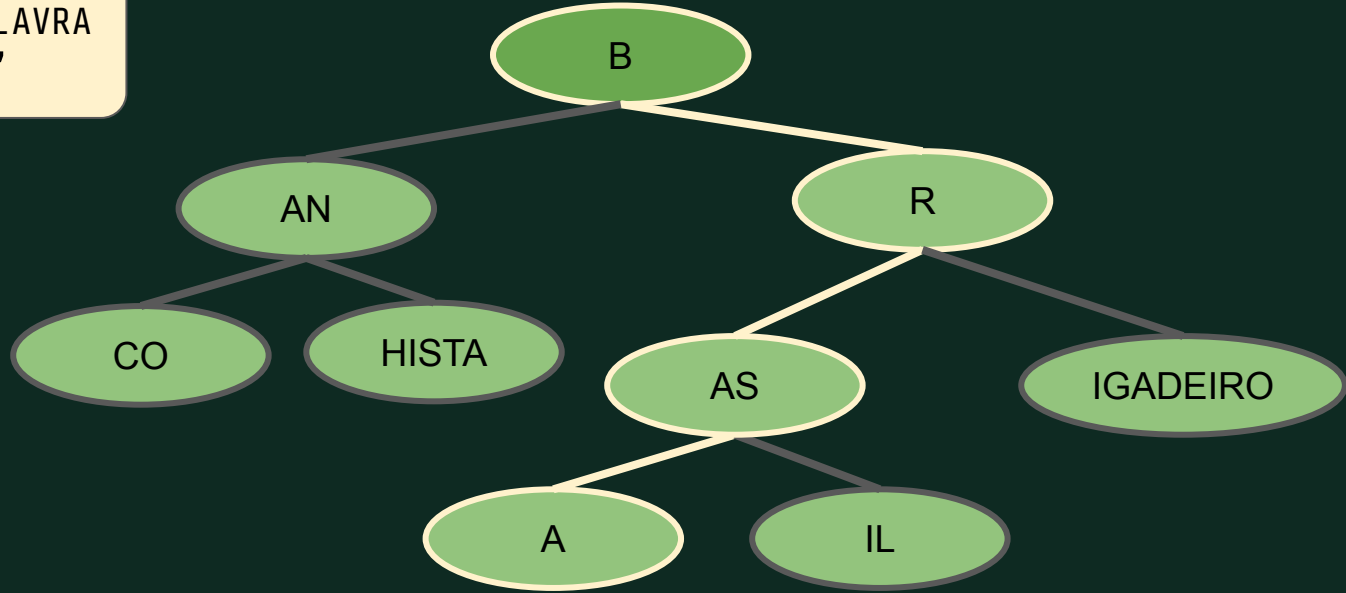
REMOVER

REMOVER PALAVRA
"BRASA"



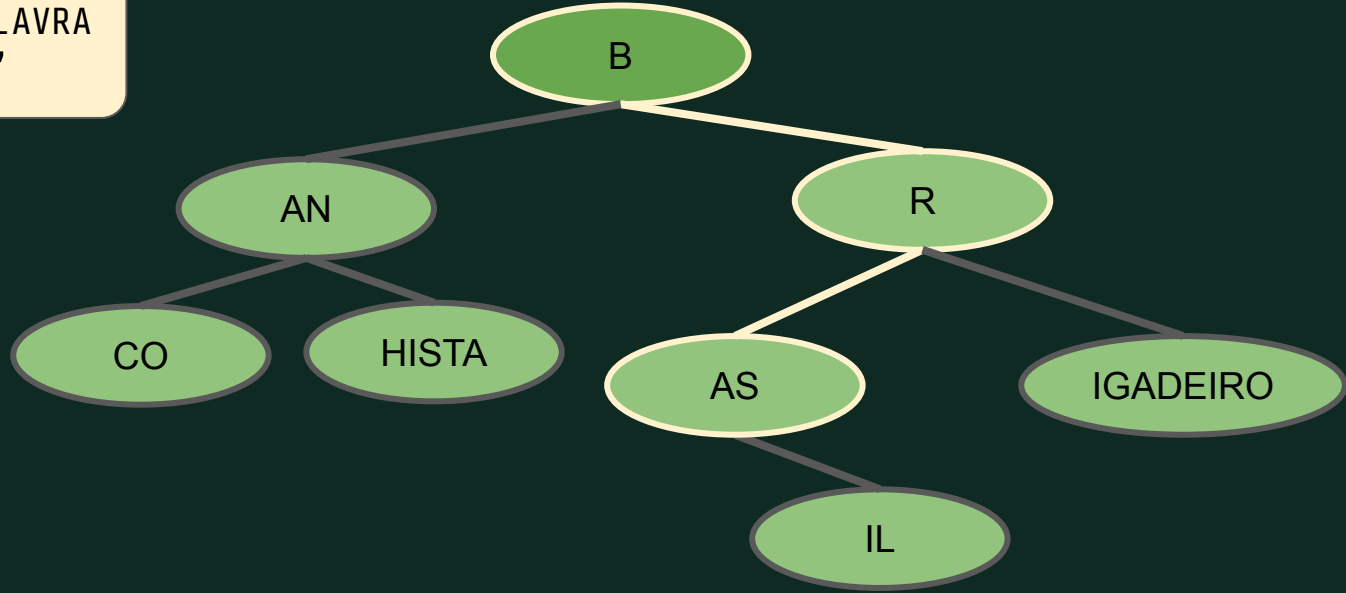
REMOVER

REMOVER PALAVRA
"BRASA"



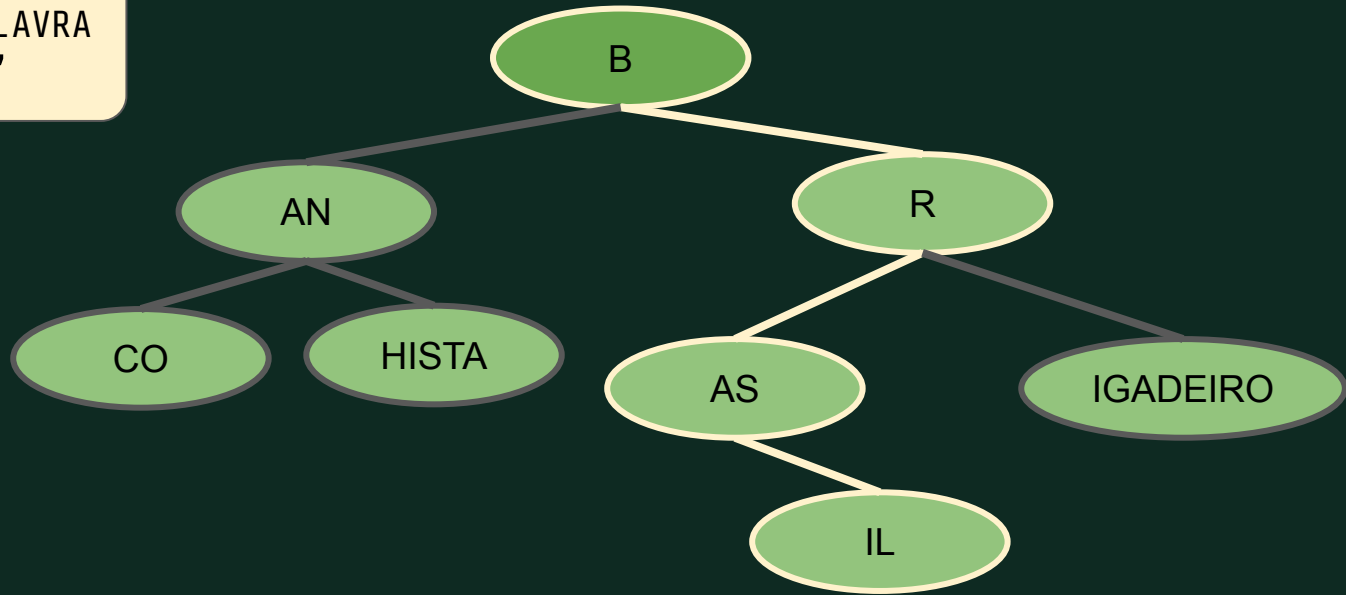
REMOVER

REMOVER PALAVRA
"BRASA"



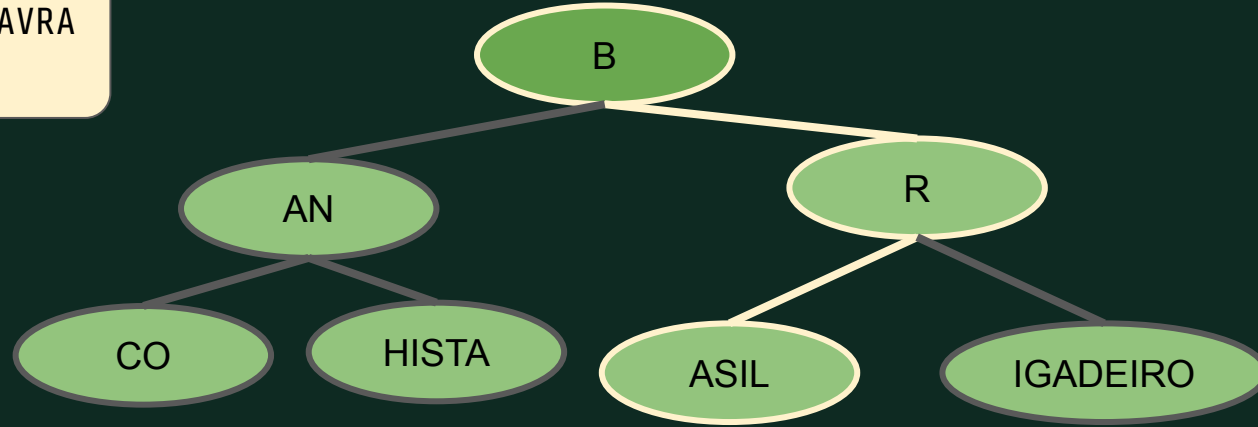
REMOVER

REMOVER PALAVRA
"BRASA"



REMOVER

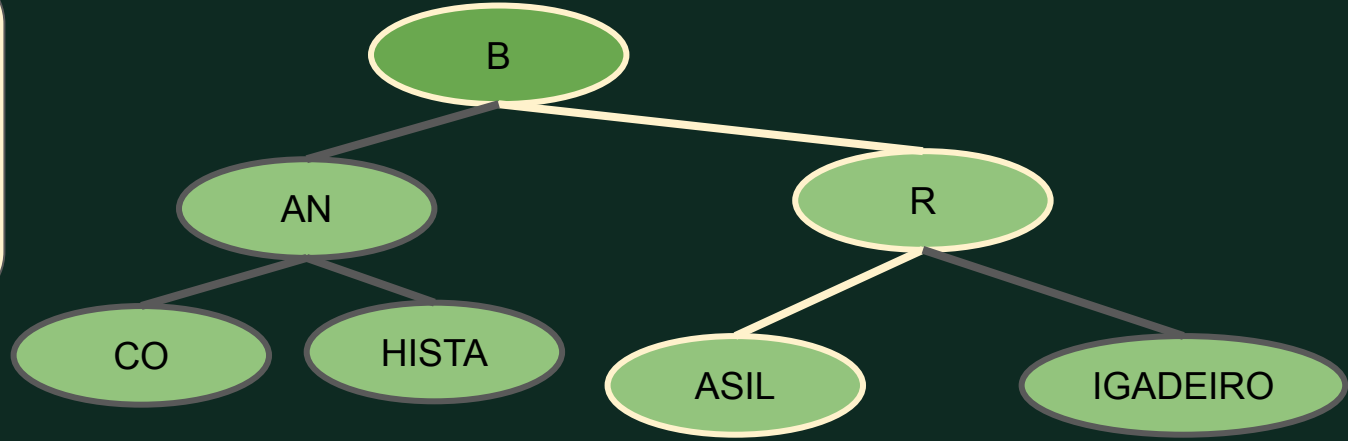
REMOVER PALAVRA
"BRASA"



REMOVER

Complexidade:
 $O(t \ N)$

t é o tamanho
da palavra
 N é o tamanho
do alfabeto



Sua Compactação



É uma variação otimizada da trie. Ela reduz o desperdício de espaço ao combinar sequências de nós com apenas um filho em um único nó.

A indexação é feita comparando as chaves bit a bit ou caractere por caractere, armazenando apenas o ponto onde as chaves passam a diferir.

Características

→

- A cada passo na árvore, seguimos um novo caractere ou bit da chave.
- O caminho da raiz até um nó representa o começo (prefixo) de uma chave.
- Os filhos de um mesmo nó compartilham esse prefixo.
- Se houver um caminho com nós que só têm um filho, ele é juntado em uma única ligação, economizando espaço.
- Diferente das tries comuns, a Árvore Patricia não guarda dados nos nós internos — apenas a posição onde as chaves se diferenciam e ponteiros para os próximos nós.

Complexidade de Operações

A complexidade das operações em uma árvore Patricia é $O(|s| + |E|)$, onde $|s|$ é a string a ser pesquisada ou inserida, e $|E|$ é o alfabeto suportado. A busca, inserção e remoção são realizadas de forma eficiente, permitindo que a árvore gerencie chaves de tamanhos variados.



Complexidade de Operações

t é o tamanho da palavra

N é o tamanho do alfabeto

T é o tamanho de todas as palavras da árvore

P é o número de palavras armazenadas na árvore

	TRIE	PATRICIA
Inserir	$O(tN)$	$O(tN)$
Buscar	$O(t)$	$O(t)$
Remover	$O(tN)$	$O(tN)$
Armazenar	$O(TN)$	$O(PN + T)$

Evolução do Algoritmo

Desde sua apresentação, a árvore Patricia evoluiu em diversas áreas, incluindo sistemas de gerenciamento de banco de dados e redes P2P. Com o tempo, suas aplicações se expandiram, e novas variações surgiram para otimizar busca e armazenamento em diferentes contextos.



Aplicações Práticas



As árvores Patricia são amplamente utilizadas em bancos de dados, sistemas de busca em redes P2P como na construção de roteadores que precisam encontrar o melhor caminho para enviar pacotes de dados, utilizando endereços IP como chaves. Escalabilidade e descentralização de Blockchains. E para aumentar a eficiência no processamento de documentos XML. Essas aplicações demonstram sua versatilidade e capacidade de lidar com estruturas de dados complexas.

Conclusões



A árvore Patricia representa uma solução eficaz para armazenamento e busca de informações, oferecendo melhorias significativas em comparação a estruturas de dados clássicas. Sua aplicação em diversas áreas reafirma sua relevância no campo das ciências da computação.

Obrigado!

Alguma pergunta?



Fontes e Referências:

→ Referências bibliográficas (livros e sites da web).

EDELKAMP, S. "Patricia tree", in Dictionary of Algorithms and Data Structures [online], Vreda Pieterse and Paul E. Black, eds. 1 December 2010.

↑ MORRISON, D. R. 1968. PATRICIA – Practical Algorithm to Retrieve Information Coded in Alphanumeric. Journal of the Association of Computing Machinery. Vol. 15, No. 4, pp. 514–534.

↑ GWEHENBERGER, G. Anwendung einer binären Verweiskettenmethode beim Aufbau von Listen. Elektronische Rechenanlagen 10 (1968), pp. 223–226.

CORMEN, Thomas H. et al. Algoritmos: Teoria e Prática. 3ª Ed. Bookman, 2012.

SEGEWICK, Robert; WAYNE, Kevin. Algorithms. 4th Ed. Addison-Wesley, 2011.

KNUTH, Donald E. The Art of Computer Programming, Vol. 3: Sorting and Searching. Addison-Wesley, 1998.

Radix tree – Wikipedia