

ÁRVORE PATRÍCIA

TRABALHO – UNIDADE 2

Disciplina: ESTRUTURAS DE DADOS – Professor: ALBERTO COSTA NETO

Estudantes: SAULO ALMEIDA DE ARAUJO E THIAGO MECENA SILVA

P – A – T – R – I – C – I – A

r a c t i c a l l g o r i t h m o e t r i e v e n f o r m a t i o n o d e d n l p h a n u m e r i c

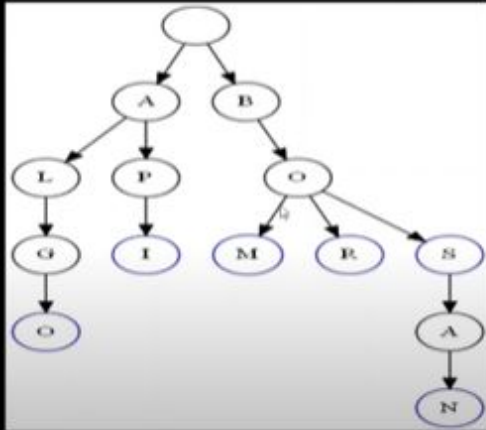
Algoritmo Prático Para Recuperar Informações Codificadas Em Alfanumérico

O QUE É?

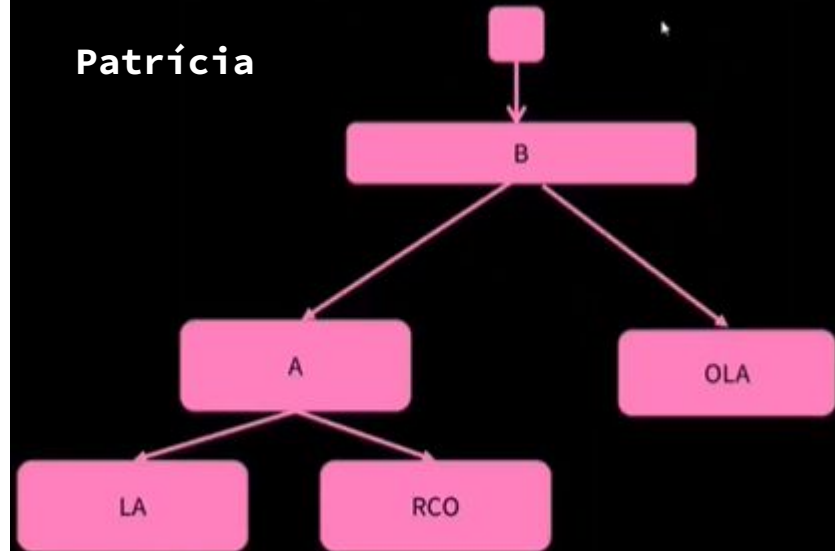
Uma árvore Patrícia (também conhecida como Trie compacta) é uma estrutura de dados utilizada em ciência da computação para armazenar um conjunto dinâmico ou estático de strings. Também é uma árvore de busca digital, que significa que ela é especialmente eficiente para buscar, inserir e excluir strings em tempo linear em relação ao tamanho das strings e ao tamanho do alfabeto usado.

TRIE E PATRÍCIA

ÁRVORE TRIE



Patrícia



IMPORTÂNCIA

A importância da árvore Patricia reside principalmente em sua capacidade de economizar espaço de armazenamento e melhorar a eficiência na busca. já que Uma Trie usa cada uma das partes de uma chave, por vez, para determinar a sub-árvore. Por outro lado, a árvore PATRICIA escolhe um elemento da chave (armazenando a sua posição) para determinar a sub-árvore. Isso remove a necessidade de nós com apenas um descendente.

VANTAGENS

Compactação de Dados: A árvore Patricia é eficiente em termos de espaço, uma vez que compartilha nós com prefixos comuns. Isso permite representar várias palavras ou chaves semelhantes usando uma única estrutura de nó. Isso é particularmente útil quando se lida com dicionários ou bancos de dados que contêm muitos prefixos comuns.

Eficiência de Busca: A busca em uma árvore Patricia é muito eficiente, uma vez que o caminho de busca é determinado pelo próprio conteúdo das chaves. Cada nível da árvore corresponde a um dígito ou caractere na chave, permitindo que a busca seja realizada em tempo linear com base no comprimento da chave, independentemente do número total de chaves armazenadas.

DESVANTAGENS

Complexidade de Implementação: A implementação da árvore Patricia pode ser mais complexa do que outras estruturas de dados mais simples, como árvores binárias de busca. Isso pode exigir mais tempo e esforço para desenvolver e depurar, especialmente para programadores menos experientes.

Não Ideal para Dados Não Ordenados: Se os dados a serem armazenados na árvore Patricia não possuírem um padrão claro de ordenação ou similaridade de prefixos, a estrutura pode não ser tão eficiente em economizar espaço ou acelerar as operações de busca.

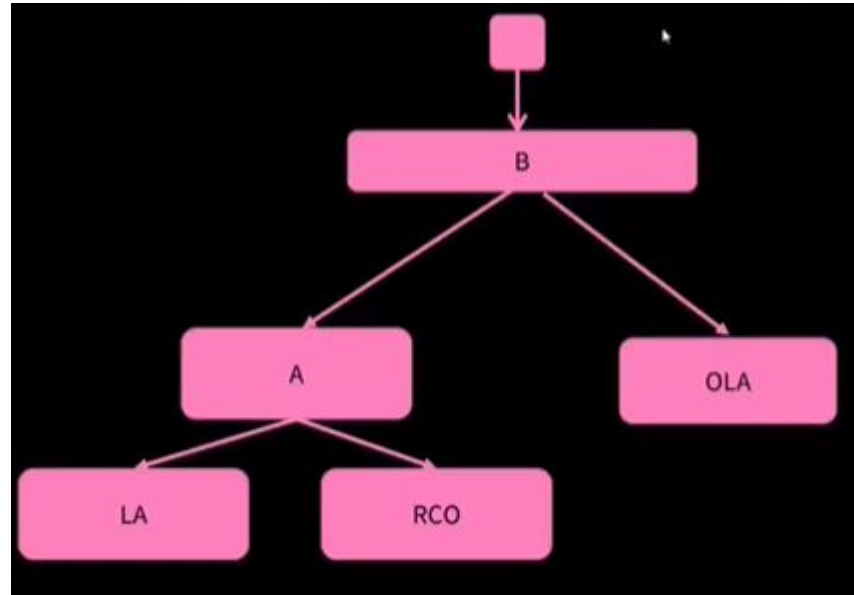
OPERAÇÕES

INSERÇÃO

- 1 - Se a Árvore estiver vazia, crie um Nó (E insira o elemento neste Nó);
- 2 - Caso a Árvore já tenha elementos, faça a comparação caracter a caracter, ou seja, verifique se algum dos nós filhos da Raiz possui algum caractere inicial semelhante ao elemento que deseja ser inserido;
- 3 - Caso encontre algum caractere inicial semelhante ao do elemento, a palavra inserida pertencerá a esse ramo;
- 4 - Continue a comparação caracter a caracter até encontrar uma diferença, “quebre” essa string nesse ponto;
- 5 - Insira o elemento com o primeiro caracter diferente “menor” como filho esquerdo, e o outro elemento como filho a direita.

INSERÇÃO - EXEMPLOS

Inserção das palavras: Bala, Barco, Bola.



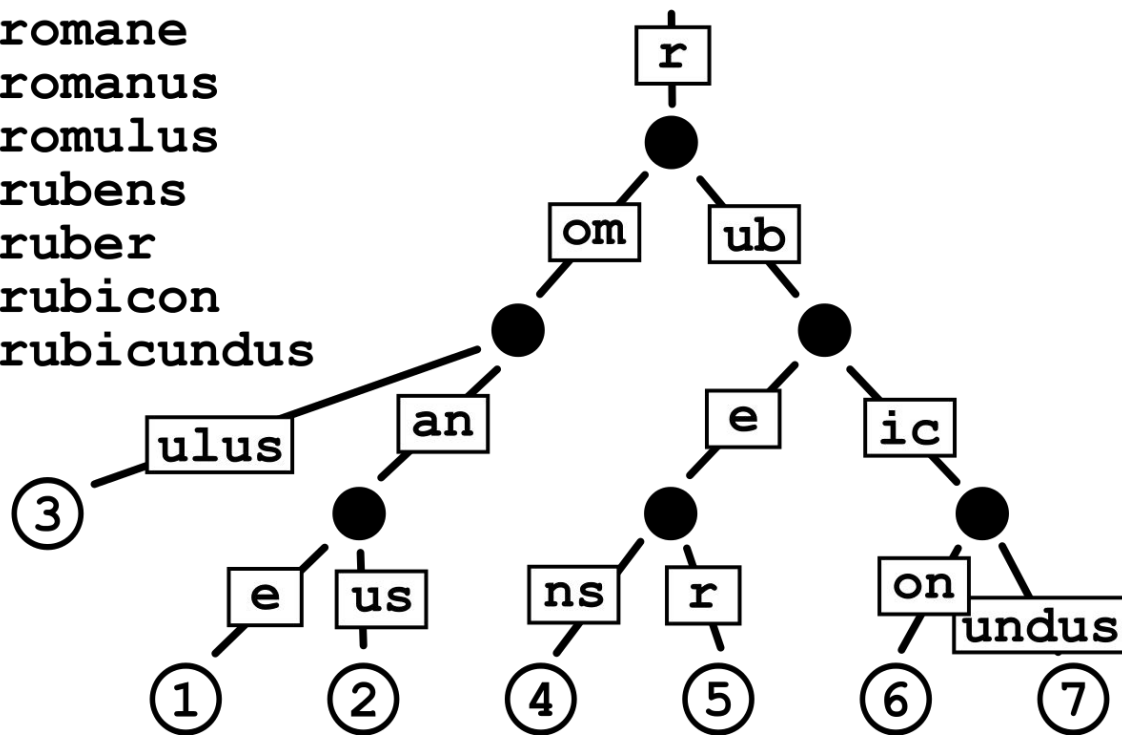
OBS.: Observar o comportamento do Nó Raiz.

BUSCA

- 1 - Verifique se há algum Nó em que o caracter inicial seja igual ao do elemento buscado;
- 2 - Caso haja, entre no ramo e continue a comparação caracter a caracter
- 3 - Se o elemento buscado estiver na Árvore, retorne Verdadeiro
- 4 - Senão, retorne Falso

BUSCA-EXEMPLO

- 1 romane
- 2 romanus
- 3 romulus
- 4 rubens
- 5 ruber
- 6 rubicon
- 7 rubicundus

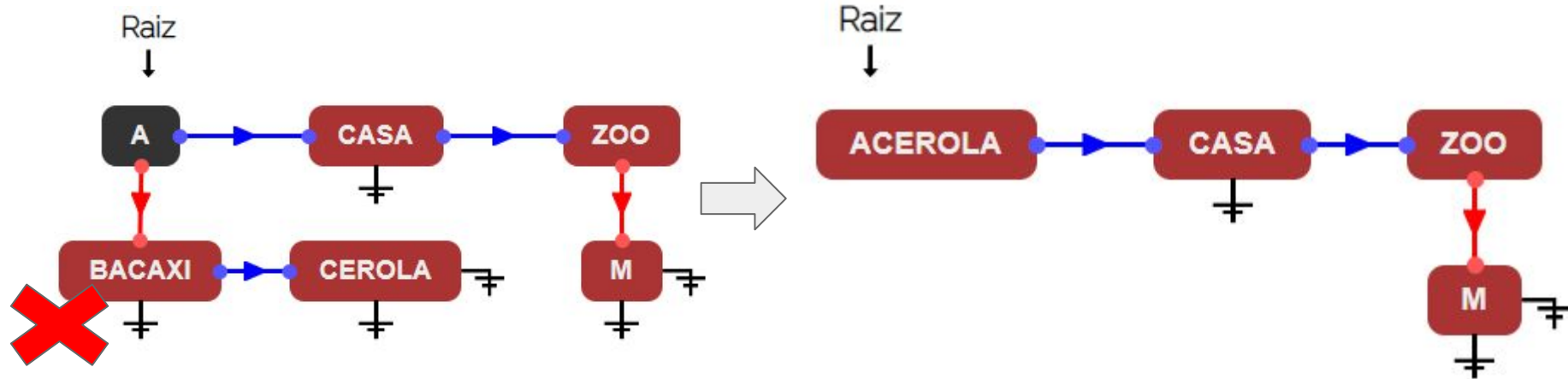


REMOÇÃO

- 1 - Faça a consulta ao elemento que deseja ser removido na Árvore;
- 2 - Se ele é um elemento da Árvore, remova-o;
 - 2.1 - Caso seja Nó Folha e seu Nó Pai só tenha um Filho, junte o Nó Filho ao Nó Pai;
 - 2.2 - Caso seja Nó Folha e seja a Raiz, remova esse Nó;
 - 2.3 - Caso tenha Filho, “remova” como elemento e faça o seu Filho ser o Nó Pai ou caso tenha mais de um Filho, faça o arranjo dessa sub-árvore;
- 3 - Se ele não é um elemento da Árvore, sinalize a ausência do elemento

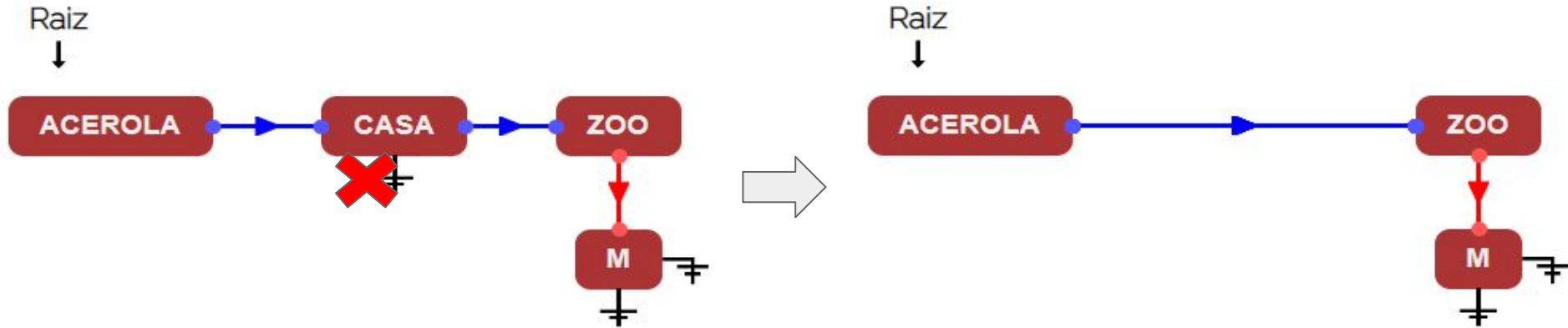
REMOÇÃO - EXEMPLOS

Remoção da palavra: ABACAXI



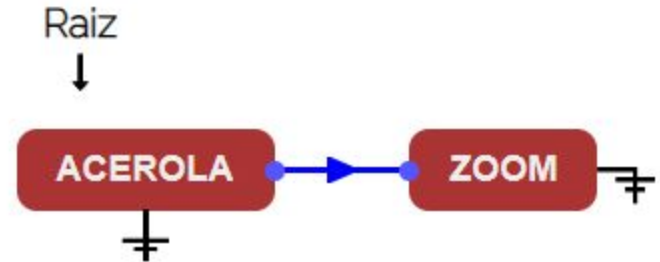
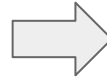
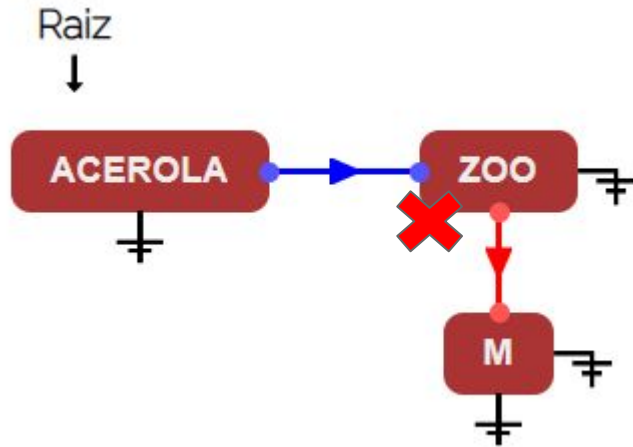
REMOÇÃO - EXEMPLOS

Remoção da palavra: CASA



REMOÇÃO - EXEMPLOS

Remoção da palavra: ZOO



COMPLEXIDADE

A Árvore Patrícia

- Inserção e Remoção: $O(tN)$;
- Busca: $O(t)$;
- Armazenamento: $O(PN + T)$.

Legenda:

t: Tamanho da palavra

N: Tamanho do alfabeto

T: Tamanho de todas as palavras na árvore

P: Número de palavras

APLICAÇÕES

APLICAÇÕES

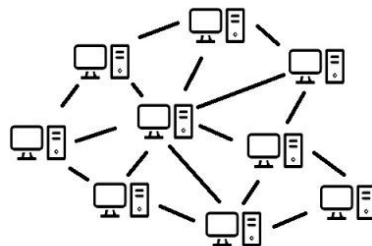
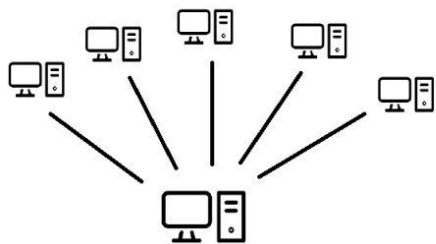
Pilares desse uso:

- Uso eficaz da Memória;
- Compressão de Nó com mesmos prefixos;
- Facilitação de Buscas;
- Eficiência de operações como pesquisa e recuperação de informações, entre outras funções.



APLICAÇÕES

➤ Buscas em Redes P2P (Peer-to-peer)



➤ Banco de Dados



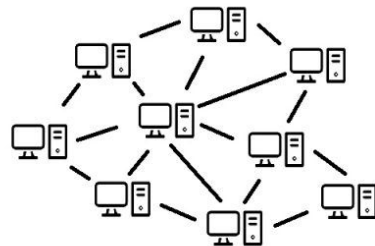
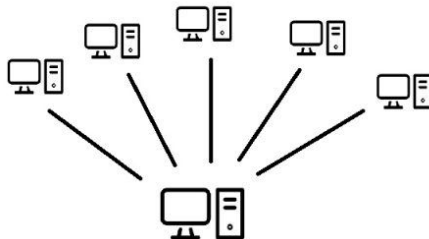
➤ Corretores Ortográficos



BUSCAS EM REDES P2P (PEER-TO-PEER)

Pesquisa eficiente:

Ao utilizar árvores Patricia para indexar conteúdo, os sistemas P2P podem oferecer pesquisas mais eficientes, uma vez que os pares podem navegar rapidamente pelos índices e encontrar os recursos desejados com base em palavras-chave ou metadados.



BANCO DE DADOS

Armazenamento de endereço de IP:

Em sistemas de armazenamento de registros de endereços IP, as árvores Patricia são frequentemente usadas para representar e pesquisar endereços IP de maneira eficiente. Cada nó na árvore representa um bit ou grupo de bits do endereço IP, tornando a pesquisa rápida e eficaz.



CORRETORES ORTOGRÁFICOS

Sugestão de correções:

Quando uma palavra é digitada incorretamente, o corretor ortográfico pode usar uma árvore Patricia para encontrar palavras similares no dicionário. A árvore pode ser usada para gerar sugestões de correção com base em palavras com prefixos ou partes semelhantes.



Dúvidas?



IMPLEMENTAÇÃO

REFERÊNCIAS BIBLIOGRÁFICAS

Introdução às Árvores Patricia. Disponível em: <https://www.youtube.com/watch?v=AgqZ0_Jzsy4&t=2s&ab_channel=ronaldoframos>. Acesso em: 28 ago. 2023.

Árvore Patrícia | Cadilag. Disponível em: <https://portaldoprofessor.fct.unesp.br/projetos/cadilag/apps/structs/rv_patricia.php>. Acesso em: 28 ago. 2023.

AED. Disponível em: <<https://jadefr.github.io/estruturas-de-dados/>>. Acesso em: 28 ago. 2023.

Pesquisa Digital. [s.l: s.n.]. Disponível em: <<https://homepages.dcc.ufmg.br/~clodoveu/files/AEDS2/AEDS2.19%20Pesquisa%20Digital.pdf>>. Acesso em: 5 set. 2023.