

Desafio: Autocompletar Palavras

Objetivo:

- Criar um sistema de autocompletar palavras usando uma árvore AVL.

Descrição:

- Carregue um corpus (conjunto de textos) para a sua aplicação. Este corpus pode ser um conjunto de livros, artigos, tweets, textos diversos, etc.
- Extraia todas as palavras únicas do corpus e insira-as em uma árvore AVL.
- Implemente uma função que receba um prefixo (parte inicial de uma palavra) e retorne uma lista com as possíveis palavras completas encontradas na árvore.
- O trabalho poderá ser realizado em grupo de até 2 pessoas.
- Compartilhe a solução no repositório do Github criado para a disciplina. Se feito em grupo, apenas um repositório é necessário. Apenas indique no arquivo readme os nomes de todos os membros. No arquivo readme compartilhe o link de um vídeo de até 5min explicando a solução e os resultados (sugestão: utilizar o Loom para gravar o vídeo).
- Na submissão da tarefa indicar na descrição o link do repositório onde está o trabalho. No arquivo em anexo inserir todo os códigos desenvolvidos em um arquivo .zip.

Etapas:

Pré-processamento do Corpus:

- Converta todo o texto para minúsculas.
- Remova pontuação e caracteres especiais.
- Divida o texto em palavras.
- Remova palavras de parada (ex: "e", "ou", "mas", "a", "o" etc.) se desejar.

Construção da Árvore AVL:

- Insira todas as palavras únicas do corpus na árvore AVL.

Autocompletar:

- Implemente uma função que percorre a árvore AVL e retorna palavras que começam com um determinado prefixo. A função deve parar de buscar na árvore assim que encontrar um nó que não corresponda ao prefixo.

Análise de Desempenho:

- Compare o desempenho (tempo de inserção, altura da árvore, tempo de busca) da árvore AVL com uma estrutura de dados mais simples, como uma lista ou uma árvore binária de busca sem balanceamento.
- Analise o impacto do tamanho do corpus no desempenho da árvore AVL.

Avaliação:

- Qualidade do pré-processamento do corpus.
- Eficiência da função de autocompletar.
- Análise detalhada do desempenho.
- Qualidade do código e clareza da implementação.
- O trabalho irá valer 3,00 (três pontos) na Unidade 01.

- O código deverá usar as classes em python disponibilizadas em sala de aula.
- É recomendado o uso do ChatGPT ou outra ferramenta de LLM para auxiliar na construção da solução.

Exemplos de Entrada & Saída

Exemplo 1

Entrada

- Corpus: "O sol brilha durante o dia enquanto a lua brilha durante a noite."
- Prefixo: "du"

Saída

- Lista de Palavras: ["durante"]
-

Exemplo 2

Entrada

- Corpus: "O gato caça o rato, enquanto o cachorro caça o gato."
- Prefixo: "ca"

Saída

- Lista de Palavras: ["caça", "cachorro"]
-

Exemplo 3

Entrada

- Corpus: "As frutas são saudáveis. Adoro comer maçãs, bananas e uvas."
- Prefixo: "ma"

Saída

- Lista de Palavras: ["maçãs"]
-

Exemplo 4

Entrada

- Corpus: "Programar em Python, Java e C++ é divertido."

- Prefixo: "P"

Saída

- Lista de Palavras: ["Programar", "Python"]
-

Exemplo 5

Entrada

- Corpus: "Gosto de café, chá e suco. Mas evito refrigerante."
- Prefixo: "e"

Saída

- Lista de Palavras: ["e", "evito"]
-

Exemplo 6

Entrada

- Corpus: "Ela estuda matemática, física e química."
- Prefixo: "ma"

Saída

- Lista de Palavras: ["matemática"]
-

Exemplo 7

Entrada

- Corpus: "Os planetas do sistema solar são Mercúrio, Vênus, Terra, Marte, Júpiter, Saturno, Urano e Netuno."
- Prefixo: "M"

Saída

- Lista de Palavras: ["Mercúrio", "Marte"]
-

Exemplo 8

Entrada

- Corpus: "Nadar, correr e pular são atividades físicas."
- Prefixo: "pu"

Saída

- Lista de Palavras: ["pular"]
-

Exemplo 9

Entrada

- Corpus: "As estações do ano são primavera, verão, outono e inverno."
- Prefixo: "z"

Saída

- Lista de Palavras: []
-

Exemplo 10

Entrada

- Corpus: "As cores do arco-íris são vermelho, laranja, amarelo, verde, azul, anil e violeta."
- Prefixo: "a"

Saída

- Lista de Palavras: ["as", "arco-íris", "amarelo", "anil"]