

Algoritmos e Estruturas de Dados II

Análise de Texto

Alunos

João Nuno Gonçalves Veloso	17729
Joel Filipe Lacerda Martins	17439

Data de Entrega: 26/05/2017

Conteúdo

Introdução.....	3
Estruturas de Dados.....	3
Funcionalidades Implementadas	5
Conclusão	6

Introdução

O trabalho desenvolvido no âmbito da unidade curricular “Algoritmos e Estruturas de Dados II”, tem como objetivo desenvolver um programa que permita analisar um texto com uma determinada estrutura.

Este projeto tem ainda a finalidade de aplicar conteúdos lecionados anteriormente da linguagem C, aprofundando-os, e ainda aplicar novos conteúdos como algoritmos de árvores binárias.

Estruturas de Dados

De modo a guardar informações importante para a execução do trabalho foram utilizadas as seguintes estruturas:

➤ **Data:** esta estrutura vai ser a mais importante do trabalho, uma vez que irá guardar informações mais importantes. A data vai guardar:

char *	<u>word</u> : A palavra
char *	<u>motto</u> : O lema
char *	<u>analyze</u> : A análise
float	<u>prob</u> : A probabilidade
int	<u>totalOccurrences</u> : as vezes que aparece
int	<u>lengthWord</u> : o seu tamanho

- **CumCol3**: esta estrutura vai guardar dados referentes a algumas alíneas tal como a 3.

char *	<u>analyze</u> : a sua análise
float	<u>prob</u> : a sua probabilidade
int	<u>count</u> : a contagem de vezes que aparece
struct <u>CumCol3</u> *	<u>left</u> : ramo da esquerda
struct <u>CumCol3</u> *	<u>right</u> : ramo da direita

- **CumCol4**: outra estrutura que vai guardar dados referentes a algumas alíneas como por exemplo a 4:

char *	<u>analyze</u> : a sua análise
float	<u>prob</u> : a sua probabilidade
int	<u>count</u> : a sua contagem
float	<u>media</u> : a sua media
float	<u>StrDev</u> : aux para desvio padrão
float	<u>totalStrDev</u> : desvio padrão
struct <u>CumCol4</u> *	<u>left</u> : ramo da esquerda
struct <u>CumCol4</u> *	<u>right</u> : ramo da direita

- **BTree**: esta estrutura de dados irá servir para guardar informações de data em árvore:

<u>Data</u>	<u>data</u> : A sua data
struct <u>Btree</u> *	<u>left</u> : os ramos da árvore

Funcionalidades Implementadas

Ao inicializar o programa, este irá apresentar um Menu:

- 1) Ler ficheiro: Esta opção irá permitir ler o ficheiro. Copia para uma árvore organizada pelas palavras e imprime quantas palavras leu e número sem repetições.
- 2) Tabela de Frequências da Categoria Gramatical: Copia o primeiro nodo da árvore para uma árvore e percorre a original e copia para outra e organiza-a através sobre a categoria gramatical, apresentando uma tabela de frequências de forma descendente.
- 3) Tabela de Frequências do Tamanho das Palavras: Percorre a árvore original, com recurso a apontadores, organizamos por classes e por fim apresentamos a tabela.
- 4) Categoria Gramatical Média : Percorre a árvore original, copia para outra árvore organizando a nível gramatical e no fim apresenta a média
- 5) Media / Moda / Mediana: Percorre a árvore original, conta o total de letras em todas as palavras do ficheiro e acrescenta esse valor numa variável. Voltamos a percorrer a mesma árvore para calcular a moda. No fim apresenta a média do tamanho das palavras, a moda e a mediana.

- 6) Quartis das palavras: o utilizador fornece a palavra, o programa passa tudo para minúsculas e procura na árvore se a encontrar fornece o quartil e as vezes que aparece.

Conclusão

Podemos concluir este trabalho foi um sucesso, com a maior parte funcional, com a exceção de algumas coisas que não conseguimos acabar, devido ao elevado número de trabalhos que enfrentamos este semestre, graças a uma pandemia que prejudicou toda gente. Neste trabalho, ficamos a perceber exatamente como funcionam as árvores que é algo que se usa muito frequentemente em linguagem C.