Relatório do projeto de Conceção e Análise de Algoritmos

**Trabalho 2**

# Tema 3: Encontra a Palavra

****

**Turma 2 Grupo F:**

Joel Márcio Torres Carneiro 201100775 ei11053@fe.up.pt

Daniel Arménio Silva Mendonça 200906506 ei12167@fe.up.pt

Francisco Teixeira Lopes 201106912 ei11056@fe.up.pt

**Breve descrição da aplicação:**

Este é um jogo para dois jogadores, *A* e *B*. Inicialmente é colocada na mesa uma palavra de *N* letras (*N*<5). Os jogadores tentam construir novas palavras a partir da palavra da mesa, isto é, palavras que contenham a palavra da mesa. Assim, os jogadores jogam alternadamente adicionando uma letra à palavra da mesa (no início ou no fim).  O jogo termina quando um dos jogadores não consegue adicionar qualquer letra à palavra da mesa, e perde o jogo. O jogador vencedor deve mostrar a palavra em que estava a pensar.

Esta aplicação consiste em implementar o jogo com um jogador automático, que jogará contra um humano.

 São aplicados algoritmos de pesquisa de strings adequados, e é necessária a existência de um ficheiro de texto que contém as palavras conhecidas que podem ser usadas no jogo.

Formalização:

D = dicionário de palavras possíveis

L = letra introduzida pelo utilizador

I = palavra inicial utilizada na aplicação

P = palavra construída na aplicação

G = palavra em jogo, já não é a palavra inicial e ainda não é a palavra final

Dados de entrada:

D(dicionário)

I(palavra inicial)

L(letra introduzida)

Dados de saída:

P(palavra final)

Objetivo:

Formar uma nova palavra a partir da que está em jogo para que o oponente não possa jogar.

A melhor jogada possível será uma palavra em que o número de caracteres restantes seja 2n, n ≥0.

Restrições:

A palavra formada pela adição de uma nova letra tem de pertencer ao dicionário.

(G(palavra em jogo) + L(letra introduzida))D(dicionário)

A palavra inicial tem de ter menos do que cinco letras

I(palavra inicial) < 5 ˄ I(palavra inicial) D(dicionário)

**Descrição da solução encontrada:**

O algoritmo de Knuth-Morris-Pratt procura a ocorrência de uma "palavra" W dentro de um "texto" S empregando a simples técnica de que quando aparece uma diferença, a palavra tem em si a informação necessária para determinar onde começar a próxima comparação.

Este algoritmo foi inventado por Knuth e Pratt e independentemente por James Morris em 1977, embora os três tenham-no publicado conjuntamente.

A parte de busca do algoritmo de Knuth-Morris-Pratt tem complexidade O (n ) , onde n é o comprimento de S(texto ou string).

Na nossa aplicação o uso deste algoritmo torna muito rápida e eficiente a pesquisa no dicionario. Desta forma é mais rapido verificar se a palavra que o jogador criou com a adição de uma letra existe ou não. Para tal a nossa implementação do algoritmo vai eliminar do nosso dicionario as palavras que não contêm a sub-string que foi criada durante a utilização da aplicação.

Exemplo:

Palavra na aplicação : “algorit”

Então são eliminadas todas as palavras que não contem “algorit” como sua sub-string. Logo, “concepção”, “análise” serão eliminadas, “algoritmo”, “algoritmos”, “algoritmia” ficarão no ficheiro .txt, por exemplo.

Desta forma, à medida que as palavras vão sendo construidas são necessárias menos comparações para descobrir se a palavra existe, o que torna mais rápido o processamento da aplicação.

A aplicação tem como referência de palavras um ficheiro .txt ao qual foi dado o nome de dicionario. É neste ficheiro que o algoritmo de Knuth-Morris-Pratt é utilizado de forma a optimizar a nossa pesquisa de palavras.

Durante a execução da aplicação é possivel adicionar no inicio ou no fim da palavra inicial uma letra de modo a formar uma nova palavra. Caso a letra introduzida não forme um palavra presente no “dicionario.txt” a aplicação pede para repensar ou então desistir e dessa forma perder o jogo.

A aplicação goza de uma navegação por entre menus de forma a tornar fácil a interação com o utilizador. O utilizador deve então introduzir o caracter que corresponde a cada opção indicada no ecra para navegar dentro da aplicação.

**Diagrama de Classes UML**

**Lista de Casos de Utilização Identificados para a Aplicação**

1. Criar palavra inicial
2. Criar nova palavra
3. Criar pontuação de jogo
4. Atribuir pontuação
5. Verificar se existe a palavra

**Principais dificuldades encontradas na resolução do trabalho**

O desenvolvimento desta aplicação trouxe algumas adversidades na escrita do código e na sua estruturação. Contudo, com alguma reflecção foram ultrapassadas, como por exemplo a implementação do algoritmo de Knuth-Morris-Pratt adaptado às nossas necessidades, e a realização de algumas verificações necessárias para a execução da aplicação.