Relatório: Encontrar Caminho entre Pares de Palavras

1 Introdução

1.1 Descrição do problema

O problema que pretendemos resolver envolve encontrar o caminho mais curto entre duas palavras num dicionário, sujeito à restrição de que cada par consecutivo de palavras no caminho só pode diferir por um único caractere. Para abordar este problema, implementámos um algoritmo que gera uma representação de lista de adjacência do grafo de palavras e emprega a Pesquisa em Largura (BFS) para identificar o caminho mais curto entre as duas palavras. Além disso, utilizamos a Pesquisa Binária para verificar a presença das palavras de entrada no dicionário.

2 Estrutura do Código

O código está organizado em várias funções principais: `adj`, `adiciona_aresta`, `bfs` e `main`.

2.1 Função adj

A função `adj` é responsável por verificar se duas palavras são adjacentes. Duas palavras são adjacentes se, alterando apenas um caractere em uma delas, as palavras se tornam iguais.

2.2 Função adiciona_aresta

A função `adiciona_aresta` é responsável por adicionar uma aresta entre os vértices 'u' e 'v' no grafo de palavras.

2.3 Função bfs

A função `bfs` é responsável por executar o algoritmo de Pesquisa em Largura (BFS) no grafo de palavras e retornar o menor número de arestas entre o vértice de origem e o vértice de destino, ou -1 se não houver um caminho entre os dois vértices.

2.4 Função main

A função `main` é responsável por ler os pares de palavras do ficheiro de entrada, ler as palavras do dicionário, construir os grafos para cada comprimento de palavra e processar os pares de palavras utilizando as funções `adj`, `adiciona_aresta` e `bfs`.

3 Complexidade Temporal

3.1 Complexidade temporal da função adj

A função `adj` tem uma complexidade temporal de

$$\mathcal{O}(l), \tag{1}$$

onde l é o comprimento das palavras. A função compara cada caractere das duas palavras e conta o número de caracteres diferentes.

3.2 Complexidade temporal da função adiciona_aresta

A função `adiciona_aresta` tem uma complexidade temporal de

$$\mathcal{O}(1), \tag{2}$$

pois apenas adiciona uma aresta entre os vértices 'u' e 'v' no grafo de palavras.

3.3 Complexidade temporal da função bfs

A função `bfs` tem uma complexidade temporal

$$\mathcal{O}(n+m), \tag{3}$$

onde n é o número de vértices (palavras) e m é o número de arestas no grafo de palavras. A função BFS visita cada vértice e aresta do grafo uma vez.

3.4 Complexidade temporal da função main

A função `main` tem uma complexidade temporal que depende das várias operações realizadas. As partes mais relevantes são a construção do grafo e o processamento dos pares de palavras.

A construção do grafo envolve iterar sobre todas as palavras do dicionário com o mesmo comprimento e verificar se são adjacentes. A complexidade temporal dessa etapa é

$$\mathcal{O}(n^2 l), \tag{4}$$

onde n é o número de palavras e l é o comprimento das palavras.

O processamento dos pares de palavras envolve a execução do algoritmo BFS, que tem uma complexidade temporal de $\mathcal{O}(n+m)$, e a pesquisa binária, que tem uma complexidade temporal de $\mathcal{O}(\log n)$. Como o BFS é executado para cada par de palavras, a complexidade total de tempo é

$$\mathcal{O}(p(n+m) + p\log n), \tag{5}$$

onde p é o número de pares de palavras.

Portanto, a complexidade temporal total da função `main` é

$$\mathcal{O}(n^2 l + p(n+m) + p\log n). \tag{6}$$

4 Conclusão

Neste relatório, apresentámos a solução para o problema de encontrar o caminho mais curto entre dois pares de palavras num dicionário, sujeito à restrição de que cada par consecutivo de palavras no caminho só pode diferir por um único caractere. A solução emprega um algoritmo de Pesquisa em Largura (BFS) para identificar o caminho mais curto e utiliza a Pesquisa Binária para verificar a presença das palavras de entrada no dicionário. Além disso, analisámos a complexidade temporal de cada função e a função `main` em geral.

O código apresentado é eficiente para resolver o problema proposto e pode ser otimizado ainda mais se necessário, dependendo dos requisitos específicos e das limitações de tempo e espaço.