

Relazione Ex2

1. Panoramica del Codice

Il codice implementa un sistema per calcolare la distanza di edit tra due stringhe e utilizza tale distanza per correggere errori di testo confrontandoli con un dizionario.

edit_distance.c e edit_distance.h: Contengono le funzioni per calcolare la distanza di edit utilizzando due approcci: ricorsivo e dinamico.

main_ex2.c: Implementa la logica per leggere un file di dizionario e un file di testo da correggere, applicando la correzione basata sulla distanza di edit.

test_ex2.c: Contiene test per le funzioni di calcolo della distanza di edit.

2. Scelte Implementative

Abbiamo implementato due metodi per calcolare la distanza di edit tra due stringhe:

Approccio Ricorsivo (edit_distance):

Utilizza la memoizzazione per ottimizzare il calcolo ricorsivo della distanza di edit. Ogni chiamata ricorsiva memorizza i risultati parziali per evitare ricalcoli.

Questo approccio è meno efficiente rispetto alla versione dinamica in quanto può comportare un alto numero di chiamate ricorsive e ha un alto overhead di memoria e tempo per stringhe lunghe.

Approccio Dinamico (edit_distance_dyn):

Utilizza una tabella bidimensionale per memorizzare i risultati intermedi e calcolare la distanza di edit in modo iterativo.

È più efficiente per stringhe di lunghezza considerevole e riduce il tempo di calcolo rispetto all'approccio ricorsivo.

Funzioni di Supporto:

`remove_punctuation`: Rimuove la punteggiatura dalle parole per facilitare il confronto.

`find_closest_word`: Trova la parola più simile nel dizionario sulla base della distanza di edit, utilizzando una soglia per limitare le correzioni ai termini vicini.

Correzione e Scrittura del Testo:

Legge il testo da correggere e per ogni parola verifica se è presente nel dizionario.

Se la parola non è nel dizionario, la funzione cerca la parola più simile e la sostituisce, altrimenti mantiene la parola originale.

3. Conclusioni

L'approccio dinamico per calcolare la distanza di edit è generalmente preferibile rispetto a quello ricorsivo, soprattutto per lunghezze maggiori delle stringhe, per via della sua efficienza computazionale e della minore complessità di tempo.

La funzione `correct_text` fornisce un buon esempio di utilizzo della distanza di edit per correzione automatica del testo.