

Relazione programmazione dinamica – EX_2

Studente: *Simone Milanesio*

Matricola: 946397



Si implementi un'applicazione che usa la funzione `edit_distance_dyn` per determinare, per ogni parola `w` in `correctme.txt`, la lista di parole in `dictionary.txt` con edit distance minima da `w`

In questa relazione, analizzo due implementazioni dell'algoritmo di `edit_distance`: una ricorsiva e una dinamica.

Per quanto riguarda `edit_distance_dyn` ho implementato una versione con **memoization** per preservare la ricorsione.

La misura del tempo di esecuzione è stata effettuata su questo set di dati:

- Dizionario di circa 66000 parole
- File di testo: *“Quando avevo cinque anni, mia madre mi ripeteva sempre che la felicità e la chiave della vita. Quando andai a scuola mi domandarono come volessi essere da grande. Io scrissi: felice. Mi dissero che non avevo capito il compito, e io dissi loro che non avevano capito la vita.”*

Risultati

Nel caso studio condotto, prendendo in considerazione la prima parola del testo [*quando*] ho ottenuto i seguenti risultati:

- Tempo di esecuzione per l'algoritmo ricorsivo: **113 secondi** di media
- Tempo di esecuzione per l'algoritmo dinamico (memoization): **0.8 secondi** circa

Prendendo in considerazione la **totalità del testo**, il tempo di esecuzione per l'algoritmo dinamico si attesta sui **31 secondi** di media.

Il confronto dei tempi di esecuzione dei due algoritmi mostra una differenza significativa nelle prestazioni.

Il risultato è coerente con le aspettative, poiché l'algoritmo ricorsivo effettua molte chiamate ricorsive ridondanti e non sfrutta la memorizzazione dei risultati intermedi.

In conclusione `edit_distance_dyn` ha una complessità $O(n)$, ossia lineare.