Relazione laboratorio: Esercizio 2

Per poter realizzare l’algoritmo dell’edit distance è stato chiesto di realizzare due funzioni entrambe ricorsive con la differenza che la seconda avrebbe dovuto attuare l’approccio della programmazione dinamica, per quanto riguarda la prima funzione ***long unsigned int recursive\_edit\_distance(char\* word1, char\* word2)*** sono necessarie solo le parole che andranno ad essere analizzate con questo approccio andiamo ad esaminare la *edit distance* di ogni ma comparandole da sinistra verso destra questo approccio ha il difetto di diramare le chiamate ricorsive creando una relazione di ricorrenza esponenziale nel caso peggiore O(2k) riscontrabile quando due parole non hanno alcun carattere in comune (riscontrato ad esempio con la parola *cinqve* non presente nel dizionario), possiamo notare che le chiamate ricorsive hanno un approccio “verso l’esterno” per questo anche se una certa sottostringa è stata calcolata precedentemente, verrà calcolata di nuovo una volta rincontrata

A questo punto possiamo passare alla versione dinamica che ci permetterà di sfruttare una matrice per poter conservare il risultato sei precedenti confronti e riutilizzarli per poter accedere alle informazioni (se esistenti) in maniera costante, la versione dinamica è ***int dynamic\_edit\_distance(char \* word1, char \* word2, int first\_len, int second\_len, int \*\*rec\_matrix)*** da come possiamo notare dai parametri viene utilizzata una array di puntatori ***int \*\*rec\_matrix*** che sarà responsabile di tenere segno delle varie operazioni, un’altra differenza dalla prima versione è che questa versione scorre le parole da destra verso sinistra potendo tenere traccia dei confronti precedenti la complessità temporale viene ridotta alla sua stessa complessità spaziale ovvero O(m \* n)