RELAZIONE DEL LABORATORIO DI ALGORITMI

Esercizio n°2: distanza di Levenshtein

		k	i	t	t	e	n
	0	1	2	3	4	5	6
s	1	1	2	3	4	5	6
i	2	2	1	2	3	4	5
t	3	3	2	1	2	3	4
t	4	4	3	2	1	2	3
i	5	5	4	3	2	2	3
n	6	6	5	4	3	3	2
g	7	7	6	5	4	4	3

		S	a	t	u	r	d	a	y
	0	1	2	3	4	5	6	7	8
S	1	0	1	2	3	4	5	6	7
u	2	1	1	2	2	3	4	5	6
n	3	2	2	2	3	3	4	5	6
d	4	3	3	3	3	4	3	4	5
a	5	4	3	4	4	4	4	3	4
y	6	5	4	4	5	5	5	4	3

Costamagna Alberto e Gianotti Damiano

10/06/2018

Matr: 833771 Matr: 835271

INTRODUZIONE

Abbiamo implementato una libreria con i due algoritmi: edit distance e edit distance dinamico. Inoltre abbiamo anche realizzato i gli usuali Unit-test per accertarsi della correttezza.

MATERIALI

1. package editdistance
2. package editdistanceusagejava
3. hamcrest-core.jar & junit.jar

PROCEDURA

COMPILAZIONE

----PER COMPILARE LE CLASSI PER LA STRUTTURA DATI Distance NEL PACKAGE editdistance---
1) posizionarsi in .../Distance/src
2) javac -d ../classes editdistance/EditDistance.java

---PER COMPILARE IL PACKAGE editdistanceusagejava---
1) posizionarsi in .../Distance/src
2) javac -d ../classes editdistanceusagejava/EditDistanceUsageJava.java

---PER COMPILARE LE CLASSI PER GLI UNIT TEST NEL PACKAGE editdistance---
1) posizionarsi in .../Distance/src
2) javac -d ../classes -cp '.;../junit-4.12.jar;../hamcrest-core-1.3.jar' editdistance/*.java

ESECUZIONE

---PER ESEGUIRE EditDistanceUsageJava---
1) posizionarsi in .../Distance/classes
2) java editdistanceusagejava/EditDistanceUsageJava "../dictionary.txt" "../correctme.txt"

---PER ESEGUIRE editdistance/EditDistance_TestsRunner---

1) posizionarsi in .../Distance/classes
2) java -cp '.;../junit-4.12.jar;../hamcrest-core-1.3.jar' editdistance/EditDistance_TestsRunner

DATA

Algoritmo utilizzato/i	File utilizzato/i	Tempo medio di esecuzione
edit_distance	"../dictionary.txt" "../correctme.txt"	tendente a ∞
edit_distance_dyn	"../dictionary.txt" "../correctme.txt"	44.741, 45.564

RISULTATI

Dopo eseguito i vari test e implementazioni richieste, non possiamo che constatare come l'utilizzo della programmazione dinamica sia la chiave per ottimizzare un algoritmo.

Sia n e m la lunghezza rispettivamente di due stringhe qualsiasi.

In particolare :

1. la complessità esponenziale della versione ricorsiva risulta poco efficiente in quanto la sua complessità media $\Theta(mn)$ dipenda dal prodotto delle lunghezze delle stringhe.
2. Invece se utilizziamo una matrice per contenere le distanze di Levenshtein tra tutti i prefissi della prima stringa e tutti i prefissi della seconda, allora possiamo calcolare e salvare i valori nella matrice e quindi trovare la distanza tra le due stringhe complete come l'ultimo valore calcolato. Con un approccio ricorsivo e bottom-up il risultato sarà collocato nella prima cella in alto a sinistra della matrice.