UNIVERSITÀ
DEGLI STUDI
FIRENZE

Dipartimento di Ingegneria Informatica

Laboratorio di Algoritmi e Strutture Dati

Autore:
Francesco Vici

N° Matricola:
7047233

Corso principale:
Algoritmi e Strutture Dati

Docente corso:
Simone Marinai

Contents

1 Introduzione tecnica **2**
1.1 Esperimento assegnato 2
1.2 Specifiche tecniche del dispositivo usato 2
1.3 Come affrontare l'esperimento 2

2 Panoramica Teorica **2**
2.1 Introduzione al problema della Longest Common Subsequence (LCS) 2
2.2 Diverse implementazioni della LCS 3
2.3 Costi delle diverse implementazioni 3

1 Introduzione tecnica

1.1 Esperimento assegnato

L'esperimento assegnato consiste nel confrontare vari modi per calcolare la LCS (Longest Common Subsequence) tra due stringhe:

- versione che utilizza l'algoritmo "forza-bruta"
- versione ricorsiva
- versione ricorsiva con memoization
- versione bottom-up

1.2 Specifiche tecniche del dispositivo usato

Poiché l'esperimento si incentrerà sui tempi di esecuzione degli algoritmi appena elencati, è indispensabile la descrizione delle specifiche hardware del computer usato per effettuare i test. Di seguito le specifiche:

CPU → **nome componente**,

RAM → **nome componente**,

SSD → **nome componente**,

Hard Disk → **nome componente**.

Mentre gli strumenti software utilizzati nella realizzazione del progetto e della relazione sono:

Linguaggio di programmazione → Python 3.13.x,

IDE → Visual Studio Code (latest release),

Editor LaTex → TeXstudio (latest release),

Compilatore LaTex → TeX Live 2023.

1.3 Come affrontare l'esperimento

La relazione dell'esperimento è costituita da 4 parti fondamentali:

- **Breve spiegazione teorica del problema**: oltre al testo dell'esercizio viene fornito un sommario delle principali caratteristiche teoriche del problema in esame, partendo dal materiale fornito durante il corso di Algoritmi e Strutture Dati ed ampliandolo con materiale esterno.
- **Documentazione del codice**: sono riportati i frammenti più importanti del codice Python insieme con uno schema UML di tutte le classi e una breve spiegazione delle struttura del progetto.
- **Descrizione dell'esperimento**: viene ripercorso lo svolgimento dell'esperimento con l'esposizione dei risultati ottenuti.
- **Analisi dei risultati e conclusioni**: i dati ottenuti nel corso dell'esperimento vengono messi in relazione con i risultati teorici attesi ed esposizione di una tesi conclusiva.

2 Panoramica Teorica

2.1 Introduzione al problema della Longest Common Subsequence (LCS)

L'algoritmo LCS, che sta per Longest Common Subsequence, consiste nel trovare la sottosequenza più lunga partendo da due stringhe qualsiasi, X e Y, date in input. Definisco inoltre come sottosequenza x di X una stringa che ha 0 o più caratteri in meno rispetto ad X. Inoltre è importante comprendere che l'ordine dei caratteri delle sequenze di partenza è fondamentale, altrimenti si parlerebbe di sottoinsiemi e non di sottosequenze.

2.2 Diverse implementazioni della LCS

Esistono numerosi metodi d'implementazione dell'algoritmo LCS, quelli utilizzati in questo esperimento saranno i seguenti:

versione con algoritmo 'Brute Force' → utilizza il processo di forza bruta per ottenere tutte le sottosequenze (di qualsiasi dimensione) di una delle due stringhe date in input, poi ne verifica la presenza nell'altra stringa in input.

versione ricorsiva → scompone il problema di partenza in coppie di sottoproblemi più semplici, fino ad arrivare al caso base ovvero quando una delle due stringhe date in input ha dimensione 0.

versione con memoization → per la risoluzione del problema utilizza lo stesso meccanismo della versione ricorsiva, ma durante la risoluzione memorizza in una matrice i le sottosequenze comuni già trovate e le utilizza per evitare di effettuare lo stesso processo più volte. Infatti ad ogni iterazione se per gli input che riceve ha già individuato una soluzione la riporta senza effettuare nuovamente il processo.

versione bottom-up → risolve i problemi a partire dalla dimensione minima e aumentandola ad ogni esecuzione. La risoluzione avviene 'al contrario' rispetto alle altre versioni.

2.3 Costi delle diverse implementazioni