

Computación, algoritmos y estructuras de datos

Trabajo Práctico Final

Autor:

Esp. Ing. Fabricio Denardi

Docente:

Dr. Camilo Argoty

Cohorte:

MIA 01-2025

Índice

1. Definición	3
2. Ejercicios	3
2.1. Ejercicio 1	3
2.1.1. Ejemplos	3
2.2. Ejercicio 2	3
2.3. Ejercicio 3	3
2.3.1. A modo de resumen: resolución	4

1. Definición

El trabajo práctico final de la materia Computación, algoritmos y estructuras de datos de la Maestría en Inteligencia artificial (MIA) de la Universidad de Buenos Aires (UBA) consiste en la resolución de una serie de ejercicios relacionados con la temática vista.

Repositorio del proyecto: https://github.com/denardifabricio/MIA_01c_CAED

2. Ejercicios

2.1. Ejercicio 1

Implementar una máquina de Turing que puede sumar dos números binarios. Se desarrolló una máquina con 3 cintas.

A su vez, se investigó y encontró información para realizar una versión con una única cinta, por lo que se implementó y comparó con la desarrollada por el alumno.

La resolución está en [Ejercicio 1](#).

2.1.1. Ejemplos

Para el caso $101 + 110 = 1011$:

1. Versión de 1 cinta:

https://drive.google.com/file/d/16Cg6Snkdgqv9JBCpi40_rP3FWkof1WLe/view?usp=drive_link

2. Versión de 3 cintas:

https://drive.google.com/file/d/1Tw2BcpaJVxhiYiwpn9w0p0koh4mIuGJ1/view?usp=drive_link

2.2. Ejercicio 2

Conteo de inversiones: Sea A un array de los números $1, 2, \dots, n$ en cualquier orden. Una inversión es una pareja (i, j) de índices del array, de forma que $i < j$ pero $A[i] > A[j]$. Encuentre un algoritmo tipo divide y vencerás que cuente el número de inversiones en un array A . Determine su complejidad en tiempo y en memoria. Implemente dicho algoritmo en su lenguaje de programación favorito con 3 arrays de ejemplo de longitud 10.

La resolución está en [Ejercicio 2](#).

2.3. Ejercicio 3

En este ejercicio se aplicarán conceptos fundamentales de algoritmos y estructuras de datos en el contexto de la bioinformática, específicamente en la alineación global de secuencias de nucleótidos. El objetivo es comprender cómo los algoritmos permiten comparar secuencias biológicas para identificar similitudes, mutaciones y relaciones evolutivas.

La resolución está en [Ejercicio 3](#).

2.3.1. A modo de resumen: resolución

- Un informe en PDF con la parte teórica y una explicación breve del código implementado.

Si bien se solicitó 1 o 2 páginas, dado que lo hice con L^AT_EX (suelen ser más páginas comparadas con un word) y quise hacerlo más completo, el informe posee más páginas.

El informe teórico puede verse [aquí](#).

- El código fuente debidamente comentado.

El código fuente puede verse en [needleman_wunsch.py](#)

- Una captura o impresión de los alineamientos generados por el programa.

Se deja un video con la ejecución del algoritmo [aquí](#).