ADA Solutions

Sistema de gestión y análisis de ADN

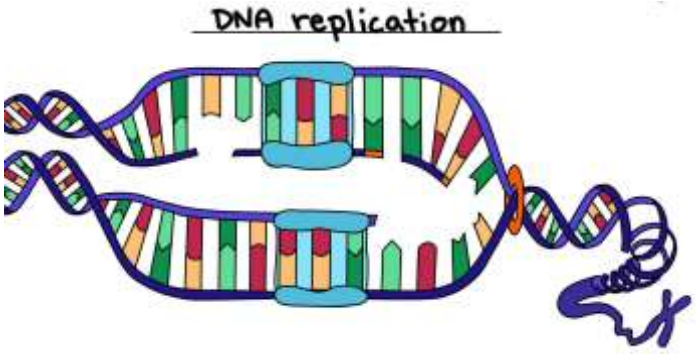
Natalia Monroy Rosas
Santiago Rodríguez Camargo
Víctor Torres Alonso

UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE BOGOTÁ

Problema a resolver

El proyecto se propone a través de las diferentes funcionalidades, aportar información sobre las cadenas de ADN que facilite su análisis para:

- Encontrar el *origen de replicación* de una secuencia.
- Facilitar la *comparación* de cadenas de ADN entre organismos.



Requerimientos funcionales

Comparación de secuencias de ADN :

El programa determina cuántas y cuáles subcadenas de longitud *m*, tienen en común dos secuencias de ADN.

El usuario selecciona las dos cadenas a comparar e introduce la longitud de las subcadenas. Luego el programa retorna la cantidad de secuencias similares en ambas y una lista de estas.

Requerimientos funcionales

Búsqueda de subcadenas más frecuentes :

Se determina cuáles son las subcadenas de longitud m más frecuentes en una secuencia de ADN.

El usuario selecciona la cadena que desea analizar y el tamaño de subcadenas que desea buscar.

El programa busca todas las subcadenas posibles de la longitud dada y determina cuál o cuáles de estas se repiten más veces.

Requerimientos funcionales

Búsqueda de ocurrencias de subcadenas específicas

Se indican los índices donde ocurre una subcadena en una secuencia.

El usuario selecciona la cadena que desea analizar e introduce la subcadena específica que desea buscar.

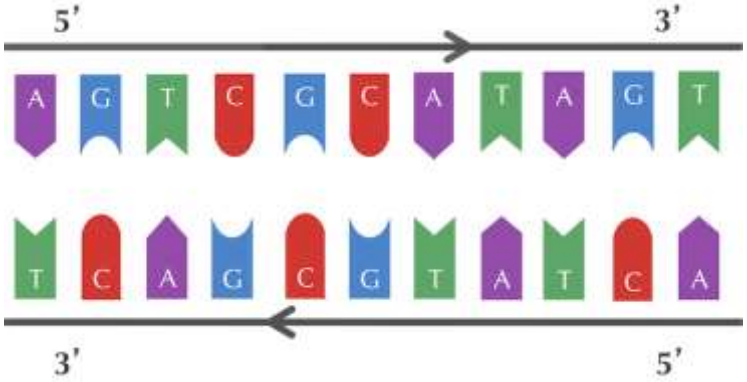
El programa busca cuántas veces aparece la subcadena en la secuencia y reporta las posiciones (índices) en las que se encuentra.

Requerimientos funcionales

Complemento reverso de una secuencia:

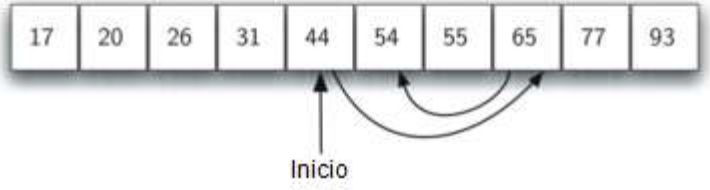
El usuario selecciona la cadena que desea analizar
El programa calcula y retorna su complemento reverso de ADN.

Ejemplo:



Uso de estructuras de datos en la solución del problema a resolver

- Diferentes versiones de la implementación principal:
  - Almacenamiento de subcadenas: Listas enlazadas, Listas con arreglos.
  - Organización lexicográfica de subcadenas: QuickSort, sort con insert de listas con arreglos, sort con pilas
  - Búsquedas y comparaciones: Lineal, Binaria.
- Inversión de cadenas, almacenamiento e impresión de resultados: Pilas, Colas

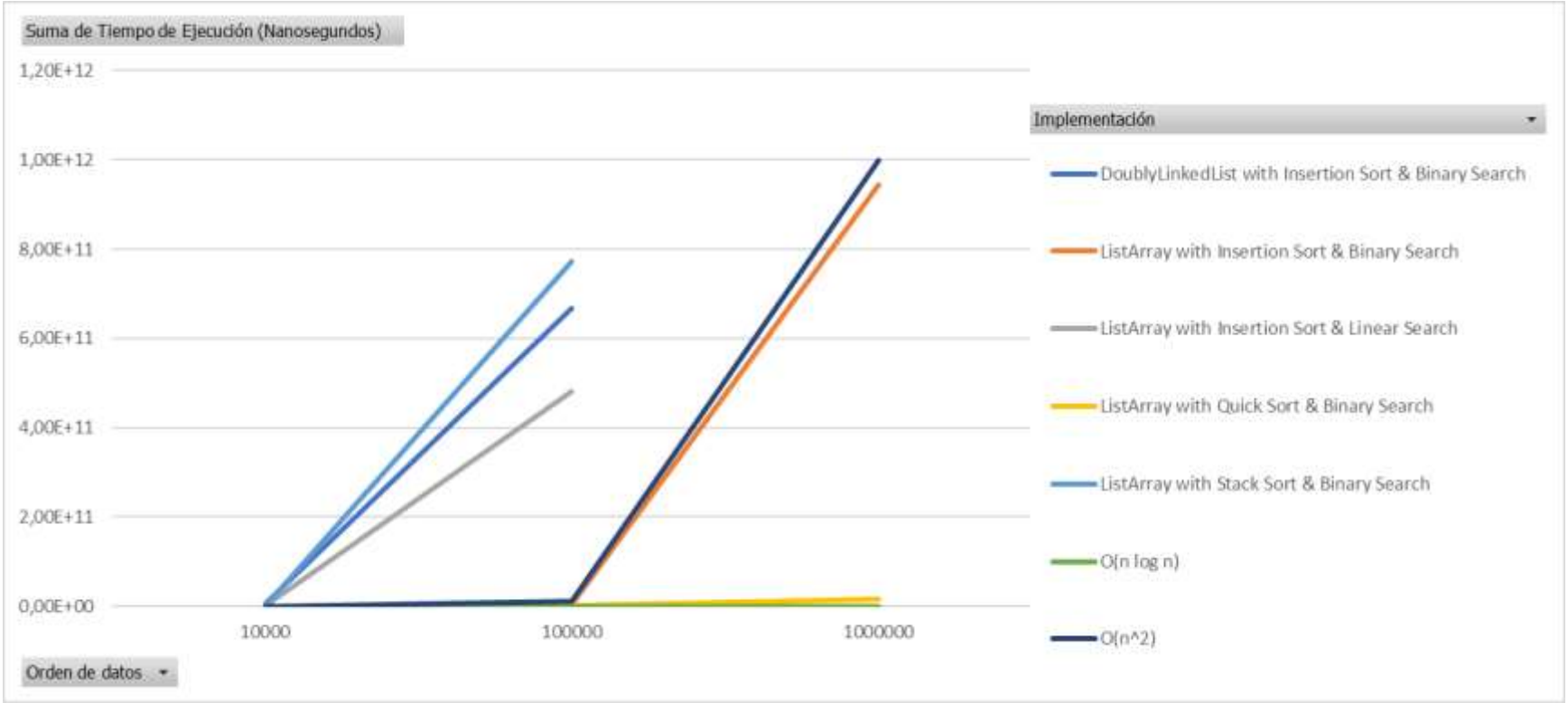


UNIVERSIDAD NACIONAL DE COLOMBIA SEDE BOGOTÁ

Pruebas y análisis comparativo del uso de las estructuras de datos

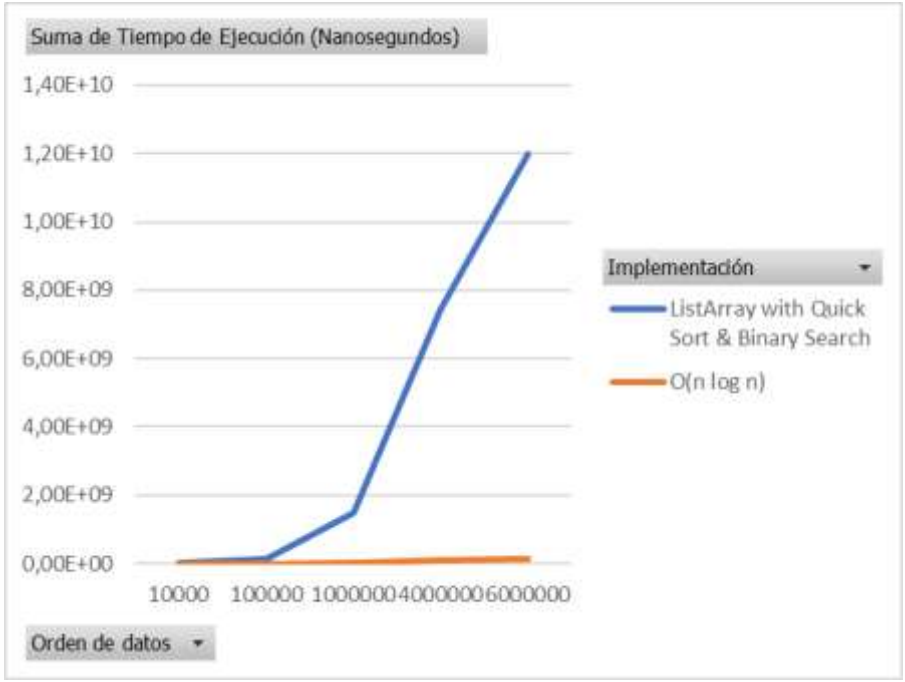
COMPARACIÓN DE SECUENCIAS				
Implementación	Tiempo de ejecución			Big (O)
	10 mil	100 mil	1 millón	
DoublyLinkedList with Insertion Sort & Binary Search	6,98E+09	6,67E+11		O(n^2)
ListArray with Insertion Sort & Binary Search	7,42E+07	4,84E+09	9,44E+11	O(n^2)
ListArray with Insertion Sort & Linear Search	1,73E+09	4,80E+11		O(n^2)
ListArray with Quick Sort & Binary Search	2,11E+07	2,19E+08	1,64E+10	O(n log n)
ListArray with Stack Sort & Binary Search	2,06E+09	7,73E+11		O(n^2)

Comparación de secuencias – Tiempos de Ejecución



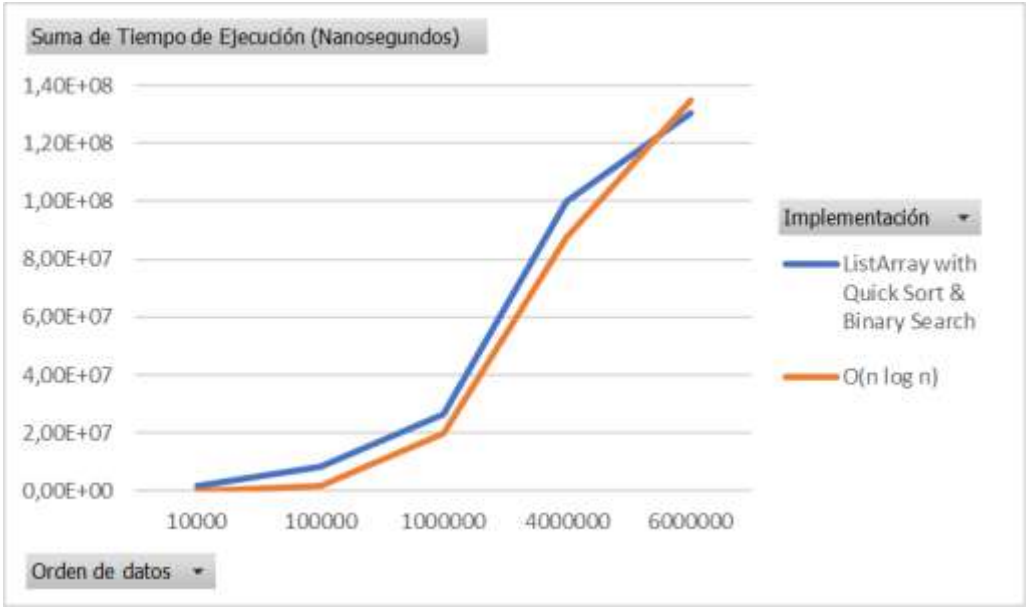
Substring más frecuente

SUBSTRING MÁS FRECUENTE		
Orden de datos	Tiempo de ejecución	Big (O)
10mil	1,93E+07	
100mil	1,39E+08	> O(n log n)
1millón	1,49E+09	< O(n^2)
4millones	7,43E+09	
6millones	1,20E+10	



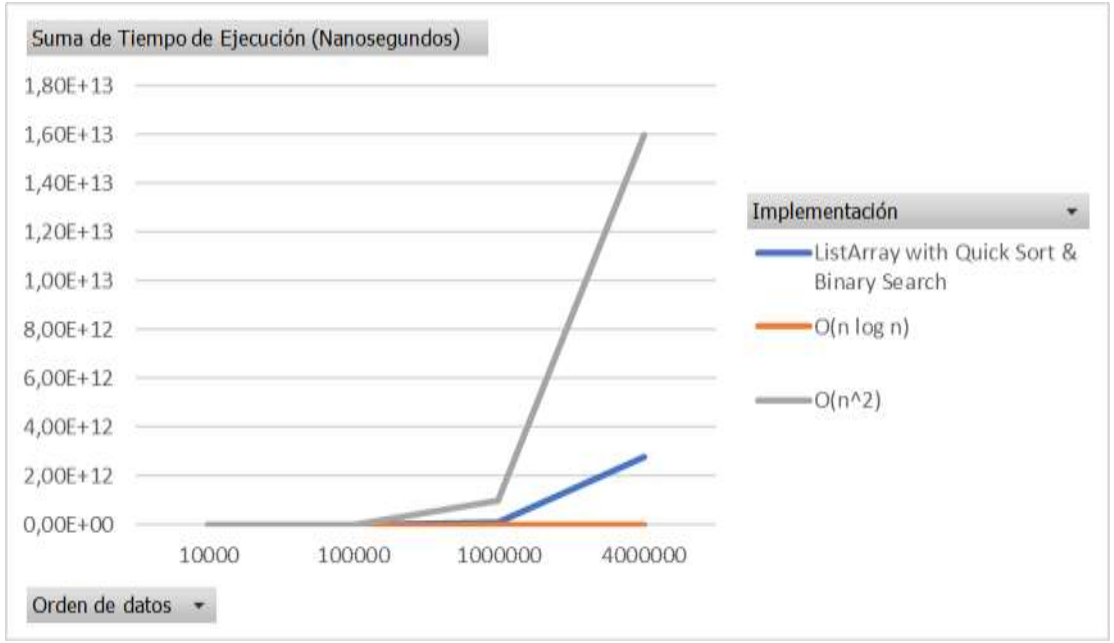
Ocurrencia subcadenas

OCCURRENCIA SUBSTRING EN SECUENCIA		
Orden de datos	Tiempo de ejecución	Big (O)
10mil	1,67E+06	
100mil	8,13E+06	
1millón	2,66E+07	O(n log n)
4millones	9,99E+07	
6millones	1,31E+08	



Complemento Reverso

COMPLEMENTO REVERSO DE SECUENCIA		
Orden de datos	Tiempo de ejecución	Big (O)
10mil	3,31E+07	
100mil	1,28E+09	> O(n log n)
1millón	1,25E+11	< O(n^2)
4millones	2,79E+12	



UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE BOGOTÁ

Gracias